



John Baillieul
Tariq Samad
Editors

Encyclopedia of Systems and Control

Second Edition

Encyclopedia of Systems and Control

John Baillieul • Tariq Samad
Editors

Encyclopedia of Systems and Control

Second Edition

With 729 Figures and 37 Tables

 Springer

Editors

John Baillieu
College of Engineering
Boston University
Boston, MA, USA

Tariq Samad
Technological Leadership Institute
University of Minnesota
Minneapolis, MN, USA

ISBN 978-3-030-44183-8 ISBN 978-3-030-44184-5 (eBook)
ISBN 978-3-030-44185-2 (print and electronic bundle)
<https://doi.org/10.1007/978-3-030-44184-5>

© Springer-Verlag London 2015

© Springer Nature Switzerland AG 2021

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG. The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface to the Second Edition

In preparing the first edition of the *Encyclopedia of Systems and Control*, we, the editors, were faced with the daunting task of organizing a discipline with deep roots and a wide footprint. To help with the undertaking, we were fortunate that leading experts from around the world agreed to serve as section editors. Our initial goal of about 200 articles was substantially exceeded, and the first edition featured 250 contributions in 30 parts – covering both theoretical and application-oriented topics.

In the several years since the first edition was published, we have been gratified by the recognition of the work and its usage. But systems and control is a dynamic discipline. Notable advances have occurred within the topics covered in the first edition. Moreover, new areas have emerged where synergy with control is being recognized and exploited. Indeed, as we reflect on a project that has now extended over almost a decade, we are struck by the increasingly cross-disciplinary currents that are influencing the directions of research in the field while also being guided by its mathematical underpinnings.

The vitality of systems and control as a discipline is evident in the additions to this second edition. Once again our goal, this time of about 50 new articles, was overachieved. With 366 articles in 39 parts, it would take a leap year of reading an article a day to exhaust the material in the new book! In addition to updates from the first edition, new topics include biomedical devices, building control, CPS/IoT, human-in-the-loop control, machine learning, control of micro and nano systems, quantum control, and vision-based control. As with the first edition, the current edition aims to make the vast array of topics that are now considered part of systems and control accessible to everyone who needs an introduction and also provides extensive pointers to further reading.

We continue to owe an enormous debt to major intellectual leaders in the field who agreed to serve as topical section editors (the list appears on the following pages). They have recruited teams of experts as authors and have managed extensive peer review of everything that appears in these volumes. We once again wish to express our gratitude to the many professionals at Springer who have supported the encyclopedia, notably Oliver Jackson, Andrew Spencer, and Vasowati Shome. We hope readers find this second edition of the encyclopedia a useful and valuable compendium for a discipline that is central to our technologically driven world today.

Boston, USA
Minneapolis, USA
April 2021

John Baillicul
Tariq Samad

Preface to the First Edition

The history of Automatic Control is both ancient and modern. If we adopt the broad view that an automatic control system is any mechanism by which an input action and output action are dynamically coupled, then the origins of this encyclopedia's subject matter may be traced back more than 2,000 years to the era of primitive time-keeping and the clepsydra water clock perfected by Ctesibius of Alexandria. In more recent history, frequently cited examples of feedback control include the automatically refilling reservoirs of flush toilets (perfected in the late nineteenth century) and the celebrated flyball steam-flow governor described in J.C. Maxwell's 1868 Royal Society of London paper – "On Governors."

Although it is useful to keep the technologies of antiquity in mind, the history of systems and control as covered in the pages of this encyclopedia begins in the twentieth century. The history was profoundly influenced by work of Nyqvist, Black, Bode, and others who were developing amplifier theory in response to the need to transmit wireline signals over long distances. This research provided major conceptual advances in feedback and stability that proved to be of interest in the theory of servomechanisms that was being developed at the same time. Driven by the need for fast and accurate control of weapons systems during World War II, automatic control developed quickly as a recognizable discipline.

While the developments of the first half of the twentieth century are an important backdrop for the *Encyclopedia of Systems and Control*, most of the topics directly treat developments from 1948 to the present. The year 1948 was auspicious for systems and control – and indeed for all the information sciences. Norbert Wiener's book *Cybernetics* was published by Wiley, the transistor was invented (and given its name), and Shannon's seminal paper "A Mathematical Theory of Communication" was published in the *Bell System Technical Journal*. In the years that followed, important ideas of Shannon, Wiener, Von Neumann, Turing, and many others changed the way people thought about the basic concepts of control systems. The theoretical advances have propelled industrial and societal impact as well (and vice versa). Today, advanced control is a crucial enabling technology in domains as numerous and diverse as aerospace, automotive, and marine vehicles; the process industries and manufacturing; electric power systems; homes and buildings; robotics; communication networks; economics and finance; and biology and biomedical devices.

It is this incredible broadening of the scope of the field that has motivated the editors to assemble the entries that follow. This encyclopedia aims to help students, researchers, and practitioners learn the basic elements of a vast array of topics that are now considered part of systems and control. The goal is to provide entry-level access to subject matter together with cross-references to related topics and pointers to original research and source material.

Entries in the encyclopedia are organized alphabetically by title, and extensive links to related entries are included to facilitate topical reading – these links are listed in “Cross-References” sections within entries. All crossreferenced entries are indicated by a preceding symbol: ►. In the electronic version of the encyclopedia these entries are hyperlinked for ease of access.

The creation of the *Encyclopedia of Systems and Control* has been a major undertaking that has unfolded over a 3-year period. We owe an enormous debt to major intellectual leaders in the field who agreed to serve as topical section editors. They have ensured the value of the opus by recruiting leading experts in each of the covered topics and carefully reviewing drafts. It has been a pleasure also to work with Oliver Jackson and Andrew Spencer of Springer, who have been unfailingly accommodating and responsive over this time.

As we reflect back over the course of this project, we are reminded of how it began. Gary Balas, one of the world’s experts in robust control and aerospace applications, came to one of us after a meeting with Oliver at the Springer booth at a conference and suggested this encyclopedia – but was adamant that he wasn’t the right person to lead it. The two of us took the initiative (ultimately getting Gary to agree to be the section editor for the aerospace control entries). Gary died last year after a courageous fight with cancer. Our sense of accomplishment is infused with sadness at the loss of a close friend and colleague.

We hope readers find this encyclopedia a useful and valuable compendium and we welcome your feedback.

Boston, USA
Minneapolis, USA
May 2015

John Baillieul
Tariq Samad

List of Topics

Adaptive Control

Section Editor: *Richard Hume Middleton*

Adaptive Control of Linear Time-Invariant Systems
Adaptive Control: Overview
Autotuning
Extremum Seeking Control
History of Adaptive Control
Iterative Learning Control
Model Reference Adaptive Control
Nonlinear Adaptive Control
Robust Adaptive Control
Switching Adaptive Control

Aerospace Applications

Section Editor: *Panagiotis Tsiotras*

Air Traffic Management Modernization:
 Promise and Challenges
Aircraft Flight Control
Inertial Navigation
Pilot-Vehicle System Modeling
Satellite Control
Space Robotics
Spacecraft Attitude Determination
Tactical Missile Autopilots
Trajectory Generation for Aerial Multicopters
Unmanned Aerial Vehicle (UAV)

Automotive and Road Transportation

Section Editor: *Luigi Glielmo*

Adaptive Cruise Control
Automated Truck Driving
Connected and Automated Vehicles

Engine Control

Fuel Cell Vehicle Optimization and Control
Lane Keeping Systems
Motorcycle Dynamics and Control
Powertrain Control for Hybrid-Electric and Electric Vehicles
Transmissions
Vehicle Dynamics Control

Biomedical Devices

Section Editor: *B. Wayne Bequette*

Automated Anesthesia Systems
Automated Insulin Dosing for Type 1 Diabetes
Control of Anemia in Hemodialysis Patients
Control of Left Ventricular Assist Devices

Biosystems and Control

Section Editor: *Elisa Franco*

Control of Drug Delivery for Type 1 Diabetes Mellitus
Deterministic Description of Biochemical Networks
Identification and Control of Cell Populations
Modeling of Pandemics and Intervention Strategies: The COVID-19 Outbreak
Monotone Systems in Biology
Reverse Engineering and Feedback Control of Gene Networks
Robustness Analysis of Biological Models
Scale-Invariance in Biological Sensing
Spatial Description of Biochemical Networks
Stochastic Description of Biochemical Networks

Structural Properties of Biological and Ecological Systems
Synthetic Biology

Building Control

Section Editor: *John T. Wen*

Building Comfort and Environmental Control
Building Control Systems
Building Energy Management System
Building Fault Detection and Diagnostics
Building Lighting Systems
Control of Circadian Rhythms and Related Processes
Emergency Building Control
Facility Control and Optimization Problems in Data Centers
Human-Building Interaction (HBI)
Vibration Control System Design for Buildings

Classical Optimal Control

Section Editor: *Michael Cantoni*

Characteristics in Optimal Control Computation
Finite-Horizon Linear-Quadratic Optimal Control with General Boundary Conditions
H₂ Optimal Control
L1 Optimal Control
Linear Matrix Inequality Techniques in Optimal Control
Linear Quadratic Optimal Control
Numerical Methods for Nonlinear Optimal Control Problems
Optimal Control and Pontryagin's Maximum Principle
Optimal Control and the Dynamic Programming Principle
Optimal Control via Factorization and Model Matching
Optimal Sampled-Data Control

Complex Systems with Uncertainty

Section Editor: *Fabrizio Dabbene*

Computational Complexity in Robustness Analysis and Design
Consensus of Complex Multi-agent Systems

Controlling Collective Behavior in Complex Systems
Dynamical Social Networks
Markov Chains and Ranking Problems in Web Search
Randomized Methods for Control of Uncertain Systems
Stability and Performance of Complex Systems Affected by Parametric Uncertainty
Stock Trading via Feedback Control Methods
Uncertainty and Robustness in Dynamic Vision

Computer-Aided Control Systems Design

Section Editor: *Andreas Varga*

Basic Numerical Methods and Software for Computer Aided Control Systems Design
Computer-Aided Control Systems Design: Introduction and Historical Overview
Descriptor System Techniques and Software Tools
Fault Detection and Diagnosis: Computational Issues and Tools
Interactive Environments and Software Tools for CACSD
Matrix Equations in Control
Model Building for Control System Synthesis
Model Order Reduction: Techniques and Tools
Multi-domain Modeling and Simulation
Optimization-Based Control Design Techniques and Tools
Robust Synthesis and Robustness Analysis Techniques and Tools
Validation and Verification Techniques and Tools

Control of Manufacturing Systems

Section Editor: *Dawn Tilbury*

Control of Additive Manufacturing
Control of Machining Processes
Programmable Logic Controllers
Run-to-Run Control in Semiconductor Manufacturing
Statistical Process Control in Manufacturing
Stream of Variations Analysis

Control of Marine Vessels

Section Editor: *Kristin Y. Pettersen*

Advanced Manipulation for Underwater Sampling
 Control of Networks of Underwater Vehicles
 Control of Ship Roll Motion
 Dynamic Positioning Control Systems for Ships and Underwater Vehicles
 Mathematical Models of Marine Vehicle-Manipulator Systems
 Mathematical Models of Ships and Underwater Vehicles
 Motion Planning for Marine Control Systems
 Underactuated Marine Control Systems

Control of Networked Systems

Section Editor: *Jorge Cortés*

Averaging Algorithms and Consensus
 Controllability of Network Systems
 Distributed Estimation in Networks
 Distributed Optimization
 Dynamic Graphs, Connectivity of Estimation and Control over Networks
 Flocking in Networked Systems
 Graphs for Modeling Networked Interactions
 Multi-vehicle Routing
 Nash Equilibrium Seeking over Networks
 Networked Systems
 Optimal Deployment and Spatial Coverage
 Oscillator Synchronization
 Privacy in Network Systems
 Routing in Transportation Networks
 Vehicular Chains

Control of Process Systems

Section Editor: *Sebastian Engell*

Control and Optimization of Batch Processes
 Control Hierarchy of Large Processing Plants: An Overview
 Control of Biotechnological Processes
 Control Structure Selection
 Controller Performance Monitoring
 Industrial MPC of Continuous Processes
 Model-Based Performance Optimizing Control

Multiscale Multivariate Statistical Process Control
 PID Control
 Real-Time Optimization of Industrial Processes
 Scheduling of Batch Plants
 State Estimation for Batch Processes

CPS / IoT

Section Editor: *Karl H. Johansson*

Controller Synthesis for CPS
 Cyber-Physical Maritime Robotic Systems
 Cyber-Physical Security
 Cyber-Physical-Human Systems
 Industrial Cyber-Physical Systems
 Medical Cyber-Physical Systems: Challenges and Future Directions

Discrete-Event Systems

Section Editor: *Christos G. Cassandras*

Applications of Discrete Event Systems
 Diagnosis of Discrete Event Systems
 Discrete Event Systems and Hybrid Systems, Connections Between
 Modeling, Analysis, and Control with Petri Nets
 Models for Discrete Event Systems: An Overview
 Opacity of Discrete Event Systems
 Perturbation Analysis of Discrete Event Systems
 Perturbation Analysis of Steady-State Performance and Relative Optimization
 Supervisory Control of Discrete-Event Systems

Distributed Parameter Systems

Section Editor: *Miroslav Krstic*

Adaptive Control of PDEs
 Backstepping for PDEs
 Bilinear Control of Schrödinger PDEs
 Boundary Control of 1-D Hyperbolic Systems
 Boundary Control of Korteweg-de Vries and Kuramoto-Sivashinsky PDEs
 Control of Fluid Flows and Fluid-Structure Models
 Control of Linear Systems with Delays
 Control of Nonlinear Systems with Delays

Input-to-State Stability for PDEs

Motion Planning for PDEs

Economic and Financial Systems

Section Editor: *Rene Carmona*

Cash Management

Credit Risk Modeling

Financial Markets Modeling

Inventory Theory

Investment-Consumption Modeling

Option Games: The Interface Between Optimal
Stopping and Game Theory

Electric Energy Systems

Section Editor: *Joe Chow*

Active Power Control of Wind Power Plants
for Grid Integration

Cascading Network Failure in Power Grid
Blackouts

Coordination of Distributed Energy Resources
for Provision of Ancillary Services:
Architectures and Algorithms

Demand Response: Coordination of Flexible
Electric Loads

Electric Energy Transfer and Control via Power
Electronics

Lyapunov Methods in Power System Stability

Model Predictive Control for Power Networks

Power System Voltage Stability

Small Signal Stability in Electric Power Systems

Time-Scale Separation in Power System Swing
Dynamics: Singular Perturbations
and Coherency

Wide-Area Control of Power Systems

Estimation and Filtering

Section Editor: *Yaakov Bar-Shalom*

Bounds on Estimation

Data Association

Estimation for Random Sets

Estimation, Survey on

Extended Kalman Filters

Kalman Filters

Nonlinear Filters

Particle Filters

Frequency-Domain Control

Section Editor: *J. David Powell*

Classical Frequency-Domain Design
Methods

Control System Optimization Methods in the
Frequency Domain

Frequency-Response and Frequency-Domain
Models

Polynomial/Algebraic Design Methods

Quantitative Feedback Theory

Spectral Factorization

Game Theory

Section Editor: *Tamer Başar*

Auctions

Cooperative Solutions to Dynamic Games

Dynamic Noncooperative Games

Evolutionary Games

Game Theory for Security

Game Theory: A General Introduction and
a Historical Overview

Learning in Games

Linear Quadratic Zero-Sum Two-Person
Differential Games

Mean Field Games

Mechanism Design

Network Games

Pursuit-Evasion Games and Zero-Sum

Two-Person Differential Games

Stochastic Games and Learning

Strategic Form Games and Nash

Equilibrium

Geometric Optimal Control

Section Editor: *Anthony Bloch*

Discrete Optimal Control

Optimal Control and Mechanics

Optimal Control with State Space
Constraints

Singular Trajectories in Optimal Control

Sub-Riemannian Optimization

Synthesis Theory in Optimal
Control

Human-in-the-Loop Control

Section Editor: *Tariq Samad*

Adaptive Human Pilot Models for Aircraft Flight Control

Human Decision-Making in Multi-agent Systems

Interaction Patterns as Units of Analysis in Hierarchical Human-in-the-Loop Control

Trust Models for Human-Robot Interaction and Control

Hybrid Systems

Section Editor: *Françoise Lamnabhi-Lagarigue*

Event-Triggered and Self-Triggered Control

Formal Methods for Controlling Dynamical Systems

Hybrid Dynamical Systems, Feedback Control of

Hybrid Model Predictive Control

Hybrid Observers

Modeling Hybrid Systems

Nonlinear Sampled-Data Systems

Output Regulation Problems in Hybrid Systems

Safety Guarantees for Hybrid Systems

Simulation of Hybrid Dynamic Systems

Stability Theory for Hybrid Dynamical Systems

Identification and Modeling

Section Editor: *Lennart Ljung*

Deep Learning in a System Identification Perspective

Experiment Design and Identification for Control

Frequency Domain System Identification

Identification of Network Connected Systems

Modeling of Dynamic Systems from First Principles

Nonlinear System Identification Using Particle Filters

Nonlinear System Identification: An Overview of Common Approaches

Nonparametric Techniques in System Identification

Nonparametric Techniques in System Identification: The Time-Varying and Missing Data Cases

Subspace Techniques in System Identification

System Identification Software

System Identification Techniques:

Convexification, Regularization, Relaxation

System Identification: An Overview

Use of Gaussian Processes in System Identification

Information-Based Control

Section Editor: *Wing Shing Wong*

Cyber-Physical Systems Security:

A Control-Theoretic Approach

Data Rate of Nonlinear Control Systems and Feedback Entropy

Information and Communication Complexity of Networked Control Systems

Information-Based Multi-agent Systems

Information-Theoretic Approaches for

Non-classical Information Patterns

Motion Description Languages and

Symbolic Control

Networked Control Systems: Architecture and Stability Issues

Networked Control Systems: Estimation and Control over Lossy Networks

Noisy Channel Effects on Multi-agent Consensus

Quantized Control and Data Rate Constraints

Intelligent Control

Section Editor: *Thomas Parisini*

Approximate Dynamic Programming (ADP)

Fault Detection and Diagnosis

Fault-Tolerant Control

Learning Theory: The Probably Approximately Correct Framework

Neuro-inspired Control

Stochastic Fault Detection

Linear Systems Theory (Time-Domain)

Section Editor: *Panos J Antsaklis*

Controllability and Observability

Linear State Feedback

Linear Systems: Continuous-Time Impulse Response Descriptions

Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions

Linear Systems: Continuous-Time,
Time-Varying State Variable Descriptions
Linear Systems: Discrete-Time Impulse
Response Descriptions
Linear Systems: Discrete-Time, Time-Invariant
State Variable Descriptions
Linear Systems: Discrete-Time, Time-Varying,
State Variable Descriptions
Observer-Based Control
Observers in Linear Systems Theory
Passivity and Passivity Indices for Switched and
Cyber-Physical Systems
Passivity, Dissipativity, and Passivity Indices
Realizations in Linear Systems Theory
Sampled-Data Systems
Stability: Lyapunov, Linear Systems
Tracking and Regulation in Linear Systems

Machine Learning

Section Editor: *Jonathan P. How*

Model-Free Reinforcement Learning-Based
Control for Continuous-Time Systems
Multiagent Reinforcement Learning
Reinforcement Learning and Adaptive Control
Reinforcement Learning for Approximate
Optimal Control
Reinforcement Learning for Control Using
Value Function Approximation

Micro-Nano Control

Section Editor: *S. O. Reza Moheimani*

Control for Precision Mechatronics
Control of Optical Tweezers
Control Systems for Nanopositioning
Dynamics and Control of Active
Microcantilevers
Mechanical Design and Control for Speed
and Precision
Noise Spectra in MEMS Coriolis Vibratory
Gyros
Non-raster Methods in Scanning Probe
Microscopy
PIDs and Biquads: Practical Control
of Mechatronic Systems
Scanning Probe Microscope Imaging Control

Sensor Drift Rejection in X-Ray Microscopy:
A Robust Optimal Control Approach

Miscellaneous

Section Editors: *Tariq Samad and
John Baillieul*

Control Applications in Audio Reproduction

Model Predictive Control

Section Editor: *James B. Rawlings*

Distributed Model Predictive Control
Economic Model Predictive Control
Explicit Model Predictive Control
Model Predictive Control in Practice
Moving Horizon Estimation
Nominal Model-Predictive Control
Optimization Algorithms for Model Predictive
Control
Robust Model Predictive Control
Stochastic Model Predictive Control
Tracking Model Predictive Control

Nonlinear Control

Section Editor: *Alberto Isidori*

Adaptive Horizon Model Predictive Control
and Al'brekht's Method
Differential Geometric Methods in Nonlinear
Control
Disturbance Observers
Feedback Linearization of Nonlinear Systems
Feedback Stabilization of Nonlinear Systems
Input-to-State Stability
Lie Algebraic Methods in Nonlinear Control
Low-Power High-Gain Observers
Lyapunov's Stability Theory
Nonlinear Zero Dynamics
Observers for Nonlinear Systems
Passivity-Based Control
Port-Hamiltonian Systems: From Modeling
to Control
Regulation and Tracking of Nonlinear Systems
Sliding Mode Control: Finite-Time Observation
and Regulation

Quantum Control

Section Editor: *Ian R. Petersen*

Application of Systems and Control Theory to
Quantum Engineering
Conditioning of Quantum Open Systems
Control of Quantum Systems
Learning Control of Quantum Systems
Pattern Formation in Spin Ensembles
Quantum Networks
Quantum Stochastic Processes and the Modelling
of Quantum Noise
Robustness Issues in Quantum Control
Single-Photon Coherent Feedback Control and
Filtering

Robotics

Section Editor: *Bruno Siciliano*

Cooperative Manipulators
Disaster Response Robot
Flexible Robots
Force Control in Robotics
Multiple Mobile Robots
Parallel Robots
Physical Human-Robot Interaction
Redundant Robots
Rehabilitation Robots
Robot Grasp Control
Robot Learning
Robot Motion Control
Robot Teleoperation
Robot Visual Control
Surgical Robotics
Underactuated Robots
Underwater Robots
Walking Robots
Wheeled Robots

Robust Control

Section Editor: *Kemin Zhou*

Fundamental Limitation of Feedback Control
H-Infinity Control
KYP Lemma and Generalizations/Applications
LMI Approach to Robust Control

Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ Control Designs
Optimization-Based Robust Control
Robust Control in Gap Metric
Robust Control of Infinite-Dimensional Systems
Robust Fault Diagnosis and Control
Robust $\mathcal{H}_2$ Performance in Feedback Control
Sampled-Data H-Infinity Optimization
Small Phase Theorem
Structured Singular Value and Applications:
Analyzing the Effect of Linear
Time-Invariant Uncertainty in Linear Systems

Stochastic Control

Section Editor: *Lei Guo*

Backward Stochastic Differential Equations and
Related Control Problems
Numerical Methods for Continuous-Time
Stochastic Control Problems
Risk-Sensitive Stochastic Control
Stochastic Adaptive Control
Stochastic Approximation with Applications
Stochastic Dynamic Programming
Stochastic Linear-Quadratic Control
Stochastic Maximum Principle

Vision-based Control

Section Editor: *Warren E. Dixon*

Control-Based Segmentation
Image-Based Estimation for Robotics and
Autonomous Systems
Image-Based Formation Control of Mobile
Robots
Image-Based Robot Control
Intermittent Image-Based Estimation
2.5D Vision-Based Estimation
Vision-Based Motion Estimation

Water Management

Section Editor: *Tariq Samad*

Demand-Driven Automatic Control of Irrigation
Channels
Distributed Sensing for Monitoring Water
Distribution Systems

Section Editors

Adaptive Control

Richard Hume Middleton School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW, Australia

Aerospace Applications

Panagiotis Tsiotras Georgia Institute of Technology, Atlanta, GA, USA

Automotive and Road Transportation

Luigi Glielmo Dipartimento di Ingegneria, Università del Sannio in Benevento, Benevento, Italy

Biomedical Devices

B. Wayne Bequette Department of Chemical & Biological Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA

Biosystems and Control

Elisa Franco Mechanical and Aerospace Engineering, University of California, Los Angeles, CA, USA

Building Control

John T. Wen Electrical, Computer, and Systems Engineering (ECSE), Rensselaer Polytechnic Institute, Troy, NY, USA

Classical Optimal Control

Michael Cantoni Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, VIC, Australia

Complex Systems with Uncertainty

Fabrizio Dabbene National Research Council of Italy, CNR-IEIIT, Torino, Italy

Computer-Aided Control Systems Design

Andreas Varga Gilching, Germany

Control of Manufacturing Systems

Dawn Tilbury Mechanical Engineering Department, University of Michigan, Ann Arbor, MI, USA

Control of Marine Vessels

Kristin Y. Pettersen Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Control of Networked Systems

Jorge Cortés Department of Mechanical and Aerospace Engineering, University of California, La Jolla, CA, USA

Control of Process Systems

Sebastian Engell Fakultät Bio- und Chemieingenieurwesen, Technische Universität Dortmund, Dortmund, Germany

CPS / IoT

Karl H. Johansson Electrical Engineering and Computer Science, Royal Institute of Technology, Stockholm, Sweden

Discrete-Event Systems

Christos G. Cassandras Division of Systems Engineering, Center for Information and Systems Engineering, Boston University, Brookline, MA, USA

Distributed Parameter Systems

Miroslav Krstic Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA, USA

Economic and Financial Systems

Rene Carmona Princeton University, Princeton, NJ, USA

Electric Energy Systems

Joe Chow Electrical & Computer Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA

Estimation and Filtering

Yaakov Bar-Shalom ECE Dept., University of Connecticut, Storrs, CT, USA

Frequency-Domain Control

J. David Powell Aero/Astro Department, Stanford University, Stanford, CA, USA

Game Theory

Tamer Başar Coordinated Science Laboratory, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA

Geometric Optimal Control

Anthony Bloch Department of Mathematics, The University of Michigan, Ann Arbor, MI, USA

Human-in-the-Loop Control

Tariq Samad Technological Leadership Institute, University of Minnesota, Minneapolis, MN, USA

Hybrid Systems

Françoise Lamnabhi-Lagarrigue Laboratoire des Signaux et Systèmes, CNRS, Gif-sur-Yvette, France

Identification and Modeling

Lennart Ljung Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping, Sweden

Information-Based Control

Wing Shing Wong Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, China

Intelligent Control

Thomas Parisini South Kensington Campus, Electrical and Electronic Engineering, Imperial College, London, UK

Linear Systems Theory (Time-Domain)

Panos J Antsaklis Electrical Engineering Department, University of Notre Dame, Notre Dame, IN, USA

Machine Learning

Jonathan P. How Department of Aeronautics and Astronautics, Aerospace Controls Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA

Micro-Nano Control

S. O. Reza Moheimani The University of Texas at Dallas, Richardson, TX, USA

Miscellaneous

Tariq Samad Technological Leadership Institute, University of Minnesota, Minneapolis, MN, USA

John Baillieul College of Engineering, Boston University, Boston, MA, USA

Model Predictive Control

James B. Rawlings Department of Chemical Engineering, University of California, Santa Barbara, CA, USA

Nonlinear Control

Alberto Isidori Department of Computer, Control, Management Engineering, University of Rome “La Sapienza”, Rome, Italy

Quantum Control

Ian R. Petersen Research School of Electrical, Energy and Materials Engineering, Australian National University, Canberra, ACT, Australia

Robotics

Bruno Siciliano Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy

Robust Control

Kemin Zhou School of Electrical and Automation Engineering, Shangdong University of Science and Technology, Shangdong, China

Stochastic Control

Lei Guo Academy of Mathematics and Systems Science, Chinese Academy of Sciences (CAS), Beijing, China

Vision-based Control

Warren E. Dixon Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA

Water Management

Tariq Samad Technological Leadership Institute, University of Minnesota, Minneapolis, MN, USA

List of Contributors

Veronica A. Adetola Pacific Northwest National Laboratory, Richland, WA, USA

Ole Morten Aamo Department of Engineering Cybernetics, NTNU, Trondheim, Norway

Waseem Abbas Vanderbilt University, Nashville, TN, USA

Daniel Abramovitch Mass Spec Division, Agilent Technologies, Santa Clara, CA, USA

Teodoro Alamo Departamento de Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla, Spain

Douglas A. Allan Department of Chemical Engineering, University of California, Santa Barbara, CA, USA

Frank Allgöwer Institute for Systems Theory and Automatic Control, University of Stuttgart, Stuttgart, Germany

Tansu Alpcan Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC, Australia

Eitan Altman INRIA, Sophia-Antipolis, France

Sean B. Andersson Mechanical Engineering and Division of Systems Engineering, Boston University, Boston, MA, USA

Henrik Anfinssen Department of Engineering Cybernetics, NTNU, Trondheim, Norway

David Angeli Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Dipartimento di Ingegneria dell'Informazione, University of Florence, Florence, Italy

Finn Ankersen European Space Agency, Noordwijk, The Netherlands

Anuradha M. Annaswamy Active-adaptive Control Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA

J. Mark Ansermino BC Children's Hospital Research Institute, Vancouver, BC, Canada

Gianluca Antonelli University of Cassino and Southern Lazio, Cassino, Italy

Panos J. Antsaklis Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Pierre Apkarian Control Department, DCSD, ONERA – The French Aerospace Lab, Toulouse, France

Filippo Arrichiello Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Cassino, Italy

Alessandro Astolfi Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma Tor Vergata, Rome, Italy

Daniele Astolfi CNRS, LAGEPP UMR 5007, Villeurbanne, France

Karl Åström Department of Automatic Control, Lund University, Lund, Sweden

Nikolaos Athanasopoulos School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, UK

Thomas A. Badgwell ExxonMobil Research and Engineering, Annandale, NJ, USA

John Baillieul College of Engineering, Boston University, Boston, MA, USA

Gary Balas Aerospace Engineering and Mechanics Department, University of Minnesota, Minneapolis, MN, USA

B. Ross Barmish ECE Department, Boston University, Boston, MA, USA

Prabir Barooah MAE-B 324, University of Florida, Gainesville, FL, USA

Yaakov Bar-Shalom University of Connecticut, Storrs, CT, USA

Tamer Başar Coordinated Science Laboratory, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA

Georges Bastin Department of Mathematical Engineering, ICTEAM, UCLouvain, Louvain-La-Neuve, Belgium

Karine Beauchard CNRS, CMLS, Ecole Polytechnique, Palaiseau, France

Burcin Becerik-Gerber Sonny Astani Department of Civil and Environmental Engineering, Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA

Nikolaos Bekiaris-Liberis Department of Electrical and Computer Engineering, Technical University of Crete, Chania, Greece

Christine M. Belcastro NASA Langley Research Center, Hampton, VA, USA

Calin Belta Mechanical Engineering, Boston University, Boston, MA, USA

Alberto Bemporad IMT Institute for Advanced Studies Lucca, Lucca, Italy

Peter Benner Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany

Pierre Bernhard INRIA-Sophia Antipolis-Méditerranée, Sophia Antipolis, France

B. Wayne Bequette Chemical and Biological Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA

Tomasz R. Bielecki Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL, USA

Franco Blanchini University of Udine, Udine, Italy

Mogens Blanke Department of Electrical Engineering, Automation and Control Group, Technical University of Denmark (DTU), Lyngby, Denmark
Centre for Autonomous Marine Operations and Systems (AMOS), Norwegian University of Science and Technology, Trondheim, Norway

Anthony Bloch Department of Mathematics, The University of Michigan, Ann Arbor, MI, USA

Bernard Bonnard Institute of Mathematics, University of Burgundy, Dijon, France

Dominique Bonvin Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland

Pablo Borja Faculty of Science and Engineering – Engineering and Technology Institute Groningen, University of Groningen, Groningen, The Netherlands

F. Borrelli University of California, Berkeley, Berkeley, CA, USA

Ugo Boscaïn CNRS, Sorbonne Université, Université de Paris, INRIA team CAGE, Laboratoire Jacques-Louis Lions, Paris, France

Michael S. Branicky Electrical Engineering and Computer Science Department, University of Kansas, Lawrence, KS, USA

James E. Braun Purdue University, West Lafayette, IN, USA

Roger Brockett Harvard University, Cambridge, MA, USA

Linda Bushnell Department of Electrical and Computer Engineering, University of Washington, Seattle, WA, USA

Fabrizio Caccavale School of Engineering, Università degli Studi della Basilicata, Viale dell'Ateneo Lucano 10, Potenza, Italy

Abel Cadenillas University of Alberta, Edmonton, AB, Canada

Kai Cai Department of Electrical and Information Engineering, Osaka City University, Osaka, Japan

Peter E. Caines Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada

Andrea Caiti DII – Department of Information Engineering and Centro “E. Piaggio”, ISME – Interuniversity Research Centre on Integrated Systems for the Marine Environment, University of Pisa, Pisa, Italy

Octavia Camps Electrical and Computer Engineering Department, Northeastern University, Boston, MA, USA

Mark Cannon Department of Engineering Science, University of Oxford, Oxford, UK

Michael Cantoni Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, VIC, Australia

Ming Cao Faculty of Science and Engineering, University of Groningen, Groningen, The Netherlands

Xi-Ren Cao Department of Automation, Shanghai Jiao Tong University, Shanghai, China

Institute of Advanced Study, Hong Kong University of Science and Technology, Hong Kong, China

Daniele Carnevale Dipartimento di Ing. Civile ed Ing. Informatica, Università di Roma “Tor Vergata”, Roma, Italy

Giuseppe Casalino University of Genoa, Genoa, Italy

Francesco Casella Politecnico di Milano, Milano, Italy

Christos G. Cassandras Division of Systems Engineering, Center for Information and Systems Engineering, Boston University, Brookline, MA, USA

David A. Castañón Boston University, Boston, MA, USA

Enrique Del Castillo Department of Industrial and Manufacturing Engineering, The Pennsylvania State University, University Park, PA, USA

Eduardo Cerpa Departamento de Matemática, Universidad Técnica Federico Santa María, Valparaíso, Chile

Aranya Chakraborty Electrical and Computer Engineering Department, North Carolina State University, Raleigh, NC, USA

François Chaumette Inria, Univ Rennes, CNRS, IRISA, Rennes, France

Chien Chern Cheah School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

Jie Chen City University of Hong Kong, Hong Kong, China

Ben M. Chen Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Shatin/Hong Kong, China

Han-Fu Chen Key Laboratory of Systems and Control, Institute of Systems Science, AMSS, Chinese Academy of Sciences, Beijing, People's Republic of China

Jian Chen State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou, China

Tongwen Chen Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada

Wei Chen Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong, China

Xiang Chen Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada

Yimin Chen Building Technology and Urban Systems Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Giovanni Cherubini IBM Research – Zurich, Rüschlikon, Switzerland

Benoît Chevalier-Roignant Emlyon Business School, Écully, France

Joe H. Chow Department of Electrical and Computer Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA

Hsiao-Dong Chiang School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA

Stefano Chiaverini Dipartimento di Ingegneria Elettrica e dell'Informazione "Maurizio Scarano", Università degli Studi di Cassino e del Lazio Meridionale, Cassino (FR), Italy

Luigi Chisci Dipartimento di Ingegneria dell'Informazione, Università di Firenze, Firenze, Italy

Alessandro Chiuso Department of Information Engineering, University of Padova, Padova, Italy

Girish Chowdhary University of Illinois at Urbana Champaign, Urbana, IL, USA

Monique Chyba University of Hawaii-Manoa, Manoa, HI, USA

Martin Corless School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, USA

Jean-Michel Coron Laboratoire Jacques-Louis Lions, Sorbonne Université, Paris, France

Jorge Cortés Department of Mechanical and Aerospace Engineering, University of California, La Jolla, CA, USA

John L. Crassidis Department of Mechanical and Aerospace Engineering, University at Buffalo, The State University of New York, Buffalo, NY, USA

- S. Crea** The Biorobotics Institute, Pisa, Italy
IRCCS Fondazione Don Carlo Gnocchi, Milan, Italy
Department of Excellence in Robotics and AI, Pisa, Italy
- Raffaello D’Andrea** ETH Zurich, Zurich, Switzerland
- Christopher D’Souza** NASA Johnson Space Center, Houston, TX, USA
- Fabrizio Dabbene** CNR-IEIT, Politecnico di Torino, Torino, Italy
National Research Council of Italy, CNR-IEIT, Torino, Italy
- A. P. Dani** Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA
- Mark L. Darby** CMiD Solutions, Houston, TX, USA
- Eyal Dassau** Harvard John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA
- Frederick E. Daum** Raytheon Company, Woburn, MA, USA
- Alessandro De Luca** Sapienza Università di Roma, Roma, Italy
- César de Prada** Departamento de Ingeniería de Sistemas y Automática, University of Valladolid, Valladolid, Spain
- Luigi del Re** Johannes Kepler Universität, Linz, Austria
- Domitilla Del Vecchio** Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA
- Diego di Bernardo** Telethon Institute of Genetics and Medicine, Pozzuoli, Italy
Department of Chemical, Materials and Industrial Engineering, University of Naples Federico II, Napoli, Italy
- Mario di Bernardo** University of Naples Federico II, Naples, Italy
Department of Electrical Engineering and ICT, University of Naples Federico II, Napoli, Italy
- Moritz Diehl** Department of Microsystems Engineering (IMTEK), University of Freiburg, Freiburg, Germany
ESAT-STADIUS/OPTEC, KU Leuven, Leuven-Heverlee, Belgium
- Steven X. Ding** University of Duisburg-Essen, Duisburg, Germany
- Warren E. Dixon** Department of Mechanical and AeroMechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA
- Ian Dobson** Iowa State University, Ames, IA, USA
- Alejandro D. Domínguez-García** University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, USA
- Daoyi Dong** School of Engineering and Information Technology, University of New South Wales, Canberra, ACT, Australia

Sebastian Dormido Departamento de Informatica y Automatica, UNED, Madrid, Spain

Francis J. Doyle III Harvard John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA

Peter M. Dower Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC, Australia

Guy A. Dumont Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada

BC Children's Hospital Research Institute, Vancouver, BC, Canada

Tyrone Duncan Department of Mathematics, University of Kansas, Lawrence, KS, USA

Aleksandr Efremov Moscow Aviation Institute, Moscow, Russia

Magnus Egerstedt Georgia Institute of Technology, Atlanta, GA, USA

Naomi Ehrich Leonard Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, USA

Evangelos Eleftheriou IBM Research – Zurich, Rüschlikon, Switzerland

Abbas Emami-Naeini Electrical Engineering Department, Stanford University, Stanford, CA, USA

Sebastian Engell Fakultät Bio- und Chemieingenieurwesen, Technische Universität Dortmund, Dortmund, Germany

Dale Enns Honeywell International Inc., Minneapolis, MN, USA

Kaan Erkorkmaz Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON, Canada

Heike Faßbender Institut Computational Mathematics, Technische Universität Braunschweig, Braunschweig, Germany

Fabio Fagnani Dipartimento di Scienze Matematiche 'G.L. Lagrange', Politecnico di Torino, Torino, Italy

Paolo Falcone Department of Electrical Engineering, Mechatronics Group, Chalmers University of Technology, Göteborg, Sweden

Engineering Department "Enzo Ferrari", University of Modena and Reggio Emilia, Modena, Italy

Maurizio Falcone Dipartimento di Matematica, SAPIENZA – Università di Roma, Rome, Italy

Alfonso Farina Selex ES, Roma, Italy

Kaveh Fathian Massachusetts Institute of Technology, Cambridge, MA, USA

Zhi Feng School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

Augusto Ferrante Dipartimento di Ingegneria dell'Informazione, Università di Padova, Padova, Italy

Fanny Ficuciello Department of Electrical Engineering and Information Technology, Università degli Studi di Napoli Federico II, Napoli, Italy

Miriam A. Figueroa-Santos University of Michigan, Ann Arbor, MI, USA

Thor I. Fossen Department of Engineering Cyberentics, Centre for Autonomous Marine Operations and Systems, Norwegian University of Science and Technology, Trondheim, Norway

Bruce A. Francis Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada

Elisa Franco Mechanical and Aerospace Engineering, University of California, Los Angeles, CA, USA

Emilio Frazzoli Massachusetts Institute of Technology, Cambridge, MA, USA

Georg Frey Saarland University, Saarbrücken, Germany

Minyue Fu School of Electrical Engineering and Computing, University of Newcastle, Callaghan, NSW, Australia

Sergio Galeani Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma "Tor Vergata", Roma, Italy

Nicholas Gans University of Texas at Arlington, Arlington, TX, USA

Mario Garcia-Sanz Case Western Reserve University, Cleveland, OH, USA

Konstantinos Gatsis Department of Engineering Science, University of Oxford, Oxford, UK

Joseph E. Gaudio Active-adaptive Control Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA

Janos Gertler George Mason University, Fairfax, VA, USA

Giulia Giordano Department of Industrial Engineering, University of Trento, Trento, TN, Italy

Alessandro Giua DIEE, University of Cagliari, Cagliari, Italy
LSIS, Aix-en-Provence, France

S. Torkel Glad Department of Electrical Engineering, Linköping University, Linköping, Sweden

Keith Glover Department of Engineering, University of Cambridge, Cambridge, UK

Ambarish Goswami Intuitive Surgical, Sunnyvale, CA, USA

John Gough Department of Physics, Aberystwyth University, Aberystwyth, Wales, UK

Pulkit Grover Carnegie Mellon University, Pittsburgh, PA, USA

Lars Grüne Mathematical Institute, University of Bayreuth, Bayreuth, Germany

Renhe Guan School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

Carlos Guardiola Universitat Politècnica de València, Valencia, Spain

Ziyang Guo Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong, China

Vijay Gupta Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Fredrik Gustafsson Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping, Sweden

Christoforos N. Hadjicostis Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus

Tore Hägglund Lund University, Lund, Sweden

Christine Haissig Honeywell International Inc., Minneapolis, MN, USA

Bruce Hajek University of Illinois, Urbana, IL, USA

Alain Haurie ORDECSYS and University of Geneva, Geneva, Switzerland
GERAD-HEC, Montréal, PQ, Canada

Aaron Havens University of Illinois at Urbana Champaign, Urbana, IL, USA

W.P.M.H. Heemels Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands

Didier Henrion LAAS-CNRS, University of Toulouse, Toulouse, France
Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic

João P. Hespanha Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA, USA

Ronald A. Hess University of California, Davis, CA, USA

Ian A. Hiskens University of Michigan, Ann Arbor, MI, USA

Håkan Hjalmarsson School of Electrical Engineering and Computer Science, Centre for Advanced Bioproduction, KTH Royal Institute of Technology, Stockholm, Sweden

Jonathan P. How Department of Aeronautics and Astronautics, Aerospace Controls Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA

Guoqiang Hu School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

Ying Hu CNRS, IRMAR – UMR6625, Université Rennes, Rennes, France

Luigi Iannelli Università degli Studi del Sannio, Benevento, Italy

Pablo A. Iglesias Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD, USA

Petros A. Ioannou University of Southern California, Los Angeles, CA, USA

Alf J. Isaksson ABB AB, Corporate Research, Västerås, Sweden

Hideaki Ishii Department of Computer Science, Tokyo Institute of Technology, Yokohama, Japan

Alberto Isidori Department of Computer and System Sciences “A. Ruberti”, University of Rome “La Sapienza”, Rome, Italy

Christina M. Ivler Shiley School of Engineering, University of Portland, Portland, OR, USA

Tetsuya Iwasaki Department of Mechanical and Aerospace Engineering, University of California, Los Angeles, CA, USA

Ali Jadbabaie University of Pennsylvania, Philadelphia, PA, USA

Matthew R. James Research School of Electrical, Energy and Materials Engineering, Australian National University, Canberra, Australia

Monique Jeanblanc Laboratoire de Mathématiques et Modélisation, LaMME, IBGBI, Université Paris Saclay, Evry, France

Karl H. Johansson Electrical Engineering and Computer Science, KTH – Royal Institute of Technology, Stockholm, Sweden

ACCESS Linnaeus Center, Royal Institute of Technology, Stockholm, Sweden

Ramesh Johari Stanford University, Stanford, CA, USA

Eric N. Johnson Pennsylvania State University, University Park, PA, USA

Girish Joshi University of Illinois at Urbana Champaign, Urbana, IL, USA

Mihailo R. Jovanović Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA

A. Agung Julius Department of Electrical, Computer, and Systems Engineering, Center for Lighting Enabled Systems and Applications, Rensselaer Polytechnic Institute, Troy, NY, USA

Raphael M. Jungers UCLouvain, ICTEAM Institute, Louvain-la-Neuve, Belgium

Yiannis Kantaros Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA

Iasson Karafyllis Department of Mathematics, National Technical University of Athens, Athens, Greece

Peter Karasev Department of Computer Science, Stony Brook University, Stony Brook, NY, USA

Hans-Michael Kaltenbach ETH Zürich, Basel, Switzerland

Christoph Kawan Courant Institute of Mathematical Sciences, New York University, New York, USA

Matthias Kawski School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ, USA

Hassan K. Khalil Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA

Mustafa Khammash Department of Biosystems Science and Engineering, Swiss Federal Institute of Technology at Zurich (ETHZ), Basel, Switzerland

Dong-Ki Kim Department of Aeronautics and Astronautics, Aerospace Controls Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA

Rudibert King Technische Universität Berlin, Berlin, Germany

Jens Kober Cognitive Robotics Department, Delft University of Technology, Delft, The Netherlands

Ivan Kolesov Department of Computer Science, Stony Brook University, Stony Brook, NY, USA

Peter Kotanko Renal Research Institute, New York, NY, USA
Icahn School of Medicine at Mount Sinai, New York, NY, USA

Basil Kouvaritakis Department of Engineering Science, University of Oxford, Oxford, UK

Arthur J. Krener Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA, USA

Miroslav Krstic Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA, USA

Vladimír Kučera Faculty of Electrical Engineering, Czech Technical University of Prague, Prague, Czech Republic

A. A. Kurzhanskiy University of California, Berkeley, Berkeley, CA, USA

Stéphane Lafortune Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA

Mark Lantz IBM Research – Zurich, Rüschlikon, Switzerland

J. Lataire Department ELEC, Vrije Universiteit Brussel, Brussels, Belgium

Kam K. Leang Department of Mechanical Engineering, University of Utah, Salt Lake City, UT, USA

Insup Lee University of Pennsylvania, Philadelphia, PA, USA

Jerome Le Ny Department of Electrical Engineering and GERAD, Polytechnique Montreal, Montreal, QC, Canada

Arie Levant Department of Applied Mathematics, Tel-Aviv University, Ramat-Aviv, Israel

Frank L. Lewis University of Texas at Arlington, Automation and Robotics Research Institute, Fort Worth, TX, USA

Jr-Shin Li Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, USA

Daniel Limon Departamento de Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla, Spain

Kang-Zhi Liu Department of Electrical and Electronic Engineering, Chiba University, Chiba, Japan

Lennart Ljung Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping, Sweden

Marco Lovera Politecnico di Milano, Milano, Italy

J. Lygeros Automatic Control Laboratory, Swiss Federal Institute of Technology Zurich (ETHZ), Zurich, Switzerland

Kevin M. Lynch Mechanical Engineering Department, Northwestern University, Evanston, IL, USA

Shangke Lyu School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

Robert T. M'Closkey Mechanical and Aerospace Engineering, Samueli School of Engineering and Applied Science, University of California, Los Angeles, CA, USA

Ronald Mahler Random Sets LLC, Eagan, MN, USA

Lorenzo Marconi C.A.S.Y. – DEI, University of Bologna, Bologna, Italy

Iven Mareels Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, VIC, Australia
IBM Research Australia, Southbank, VIC, Australia

Jonas Mårtensson Electrical Engineering and Computer Science, KTH – Royal Institute of Technology, Stockholm, Sweden

David Martin De Diego Instituto de Ciencias Matemáticas (CSIC-UAM-UC3M-UCM), Madrid, Spain

Nuno C. Martins Department of Electrical and Computer Engineering, Institute for Systems Research, University of Maryland, College Park, MD, USA

Sonia Martínez Department of Mechanical and Aerospace Engineering, University of California, La Jolla, San Diego, CA, USA

Sheikh Mashrafi Argonne National Laboratory, Lemont, IL, USA

Johanna L. Mathieu Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA

Wolfgang Mauntz Fakultät Bio- und Chemieingenieurwesen, Technische Universität Dortmund, Dortmund, Germany

Anastasia Mavrommati Advanced Research and Technology Office, The MathWorks, Inc., Natick, MA, USA

Volker Mehrmann Institut für Mathematik MA 4-5, Technische Universität Berlin, Berlin, Germany

Claudio Melchiorri Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione, Alma Mater Studiorum Università di Bologna, Bologna, Italy

Mehran Mesbahi University of Washington, Seattle, WA, USA

Alexandre R. Mesquita Department of Electronics Engineering, Federal University of Minas Gerais, Belo Horizonte, Brazil

Bérénice Mettler International Computer Science Institute (ICSI), Berkeley, CA, USA

Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN, USA

Thomas Meurer Automatic Control Chair, Faculty of Engineering, Kiel University, Kiel, Germany

Wim Michiels Numerical Analysis and Applied Mathematics Section, KU Leuven, Leuven (Heverlee), Belgium

Richard Hume Middleton School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW, Australia

Sandipan Mishra Department of Mechanical, Aerospace, and Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA

S. O. Reza Moheimani The University of Texas at Dallas, Richardson, TX, USA

Julian Morris School of Chemical Engineering and Advanced Materials, Centre for Process Analytics and Control Technology, Newcastle University, Newcastle Upon Tyne, UK

Pieter J. Mosterman Advanced Research and Technology Office, The MathWorks, Inc., Natick, MA, USA

James Moyne Mechanical Engineering Department, University of Michigan, Ann Arbor, MI, USA

Curtis P. Mracek Raytheon Missile Systems, Waltham, MA, USA

- Mark W. Mueller** University of California, Berkeley, CA, USA
- Hideo Nagai** Department of Mathematics, Kansai University, Osaka, Japan
- William S. Nagel** Department of Mechanical Engineering, University of Utah, Salt Lake City, UT, USA
- Girish N. Nair** Department of Electrical and Electronic Engineering, University of Melbourne, Melbourne, VIC, Australia
- Ciro Natale** Dipartimento di Ingegneria, Università degli Studi della Campania Luigi Vanvitelli, Aversa, Italy
- Angelia Nedić** Industrial and Enterprise Systems Engineering, University of Illinois, Urbana, IL, USA
- Arye Nehorai** Preston M. Green Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, USA
- Dragan Nesic** Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC, Australia
- Yuqing Ni** Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong, China
- Brett Ninness** School of Electrical and Computer Engineering, University of Newcastle, Newcastle, Australia
- Hidekazu Nishimura** Graduate School of System Design and Management, Keio University, Yokoyama, Japan
- Dominikus Noll** Institut de Mathématiques, Université de Toulouse, Toulouse, France
- Lorenzo Ntogramatzidis** Department of Mathematics and Statistics, Curtin University, Perth, WA, Australia
- Hendra I. Nurdin** School of Electrical Engineering and Telecommunications, University of New South Wales (UNSW), Sydney, NSW, Australia
- Tom Oomen** Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
- Giuseppe Oriolo** Sapienza Università di Roma, Roma, Italy
- Romeo Ortega** Laboratoire des Signaux et Systèmes, Centrale Supélec, CNRS, Plateau de Moulon, Gif-sur-Yvette, France
- Richard W. Osborne** University of Connecticut, Storrs, CT, USA
- Martin Otter** Institute of System Dynamics and Control, German Aerospace Center (DLR), Wessling, Germany
- David H. Owens** Zhengzhou University, Zhengzhou, P.R. China
Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK

Hitay Özbay Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey

Asuman Ozdaglar Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA

Andrew Packard Mechanical Engineering Department, University of California, Berkeley, CA, USA

Fernando Paganini Universidad ORT Uruguay, Montevideo, Uruguay

Gabriele Pannocchia University of Pisa, Pisa, Italy

Angeliki Pantazi IBM Research – Zurich, Rüschlikon, Switzerland

Lucy Y. Pao University of Colorado, Boulder, CO, USA

George J. Pappas Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA

Shinkyu Park Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, USA

Frank C. Park Robotics Laboratory, Seoul National University, Seoul, Korea

Bozenna Pasik-Duncan Department of Mathematics, University of Kansas, Lawrence, KS, USA

Fabio Pasqualetti Department of Mechanical Engineering, University of California at Riverside, Riverside, CA, USA

Ron J. Patton School of Engineering, University of Hull, Hull, UK

Lacra Pavel Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada

Marco Pavone Stanford University, Stanford, CA, USA

Shige Peng Shandong University, Jinan, Shandong Province, China

Tristan Perez Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, QLD, Australia

Ian R. Petersen Research School of Electrical, Energy and Materials Engineering, Australian National University, Canberra, ACT, Australia

Kristin Y. Pettersen Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Benedetto Piccoli Mathematical Sciences and Center for Computational and Integrative Biology, Rutgers University, Camden, NJ, USA

Rik Pintelon Department ELEC, Vrije Universiteit Brussel, Brussels, Belgium

Boris Polyak Institute of Control Science, Moscow, Russia

Romain Postoyan Université de Lorraine, CRAN, France

CNRS, CRAN, France

J. David Powell Aero/Astro Department, Stanford University, Stanford, CA, USA

Maria Pozzi Department of Information Engineering and Mathematics, University of Siena, Siena, Italy

Advanced Robotics Department, Istituto Italiano di Tecnologia, Genova, Italy

Laurent Praly MINES ParisTech, PSL, Fontainebleau, France

Domenico Prattichizzo Department of Information Engineering and Mathematics, University of Siena, Siena, Italy

Advanced Robotics Department, Istituto Italiano di Tecnologia, Genova, Italy

Curt Preissner Argonne National Laboratory, Lemont, IL, USA

James A. Primbs Department of Finance, California State University Fullerton, Fullerton, CA, USA

Anton Proskurnikov Department of Electronics and Telecommunications, Politecnico di Torino, Turin, TO, Italy

Institute for Problems in Mechanical Engineering, Russian Academy of Sciences, St. Petersburg, Russia

S. Joe Qin University of Southern California, Los Angeles, CA, USA

Li Qiu Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong, China

Rajesh Rajamani Department of Mechanical Engineering, University of Minnesota, Twin Cities, Minneapolis, MN, USA

Akshay Rajhans Advanced Research and Technology Office, The MathWorks, Inc., Natick, MA, USA

Saša V. Raković School of Automation, Beijing Institute of Technology, Beijing, China

Chiara Ravazzi Institute of Electronics, Computer and Telecommunication Engineering (IEIIT), National Research Council of Italy (CNR), Turin, TO, Italy

James B. Rawlings Department of Chemical Engineering, University of California, Santa Barbara, CA, USA

Jean-Pierre Raymond Institut de Mathématiques, Université Paul Sabatier Toulouse III and CNRS, Toulouse Cedex, France

Adam Regnier Kinetic Buildings, Philadelphia, PA, USA

Juan Ren Mechanical Engineering Department, Iowa State University, Ames, IA, USA

Wei Ren Department of Electrical Engineering, University of California, Riverside, CA, USA

Spyros Reveliotis School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Giorgio Rizzoni Department of Mechanical and Aerospace Engineering, Center for Automotive Research, The Ohio State University, Columbus, OH, USA

L. C. G. Rogers University of Cambridge, Cambridge, UK

Sabrina Rogg Fresenius Medical Care Deutschland GmbH, Bad Homburg, Germany

Marcello Romano Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, CA, USA

Pierre Rouchon Centre Automatique et Systèmes, Mines ParisTech, Paris Cedex 06, France

Michael G. Ruppert The University of Newcastle, Callaghan, NSW, Australia

Murti V. Salapaka Department of Electrical and Computer Engineering, University of Minnesota Twin-Cities, Minneapolis, MN, USA

Srinivasa Salapaka University of Illinois at Urbana-Champaign, Urbana, IL, USA

Henrik Sandberg Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden

Ricardo G. Sanfelice Department of Electrical and Computer Engineering, University of California, Santa Cruz, CA, USA

Simo Särkkä Aalto University, Helsinki, Finland

Ketan Savla Sonny Astani Department of Civil and Environmental Engineering, University of Southern California, Los Angeles, CA, USA

Heinz Schättler Washington University, St. Louis, MO, USA

Thomas B. Schön Department of Information Technology, Uppsala University, Uppsala, Sweden

Johan Schoukens Department ELEC, Vrije Universiteit Brussel, Brussels, Belgium

Abu Sebastian IBM Research – Zurich, Rüschlikon, Switzerland

Michael Sebek Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague 6, Czech Republic

Peter Seiler Aerospace Engineering and Mechanics Department, University of Minnesota, Minneapolis, MN, USA

Lina Sela Department of Civil, Architectural and Environmental Engineering, The University of Texas at Austin, Austin, TX, USA

Suresh P. Sethi Jindal School of Management, The University of Texas at Dallas, Richardson, TX, USA

Sirish L. Shah Department of Chemical and Materials Engineering, University of Alberta Edmonton, Edmonton, AB, Canada

Jeff S. Shamma School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Pavel Shcherbakov Institute of Control Science, Moscow, Russia

Jianjun Shi Georgia Institute of Technology, Atlanta, GA, USA

Ling Shi Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong, China

Hyungbo Shim Department of Electrical and Computer Engineering, Seoul National University, Seoul, Republic of Korea

Sigurd Skogestad Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Manuel Silva Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza, Spain

Vasile Sima National Institute for Research and Development in Informatics, Bucharest, Romania

Marwan A. Simaan Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, USA

Robert E. Skelton Texas A&M University, College Station, TX, USA

Oleg Sokolsky University of Pennsylvania, Philadelphia, PA, USA

Eduardo D. Sontag Departments of Bioengineering and Electrical and Computer Engineering, Northeastern University, Boston, MA, USA

Asgeir J. Sørensen Department of Marine Technology, Centre for Autonomous Marine Operations and Systems (AMOS), Norwegian University of Science and Technology, NTNU, Trondheim, Norway

Mark W. Spong Department of Systems Engineering, Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, TX, USA

R. Srikant Department of Electrical and Computer Engineering and the Coordinated Science Lab, University of Illinois at Urbana-Champaign, Champaign, IL, USA

Anna G. Stefanopoulou University of Michigan, Ann Arbor, MI, USA

Jörg Stelling ETH Zürich, Basel, Switzerland

Jing Sun University of Michigan, Ann Arbor, MI, USA

Krzysztof Szajowski Faculty of Pure and Applied Mathematics, Wrocław University of Science and Technology, Wrocław, Poland

Mario Sznaiier Electrical and Computer Engineering Department, Northeastern University, Boston, MA, USA

Paulo Tabuada Electrical and Computer Engineering Department, UCLA, Los Angeles, CA, USA

Department of Electrical Engineering, University of California, Los Angeles, CA, USA

Satoshi Tadokoro Tohoku University, Sendai, Japan

Gongguo Tang Department of Electrical Engineering and Computer Science, Colorado School of Mines, Golden, CO, USA

Shanjian Tang Fudan University, Shanghai, China

Allen Tannenbaum Department of Computer Science, Stony Brook University, Stony Brook, NY, USA

João Tasso de Figueiredo Borges de Sousa Laboratório de Sistemas e Tecnologia Subaquática (LSTS), Faculdade de Engenharia da Universidade do Porto, Porto, Portugal

Andrew R. Teel Electrical and Computer Engineering Department, University of California, Santa Barbara, CA, USA

Roberto Tempo CNR-IEIIT, Politecnico di Torino, Torino, Italy

Onur Toker Electrical and Computer Engineering, Florida Polytechnic University, Lakeland, FL, USA

H.L. Trentelman Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, Groningen, AV, The Netherlands

Jorge Otávio Trierweiler Group of Intensification, Modelling, Simulation, Control and Optimization of Processes (GIMSCOP), Department of Chemical Engineering, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, RS, Brazil

Lenos Trigeorgis Bank of Cyprus Chair Professor of Finance, University of Cyprus, Nicosia, Cyprus

E. Trigili The Biorobotics Institute, Pisa, Italy
Department of Excellence in Robotics and AI, Pisa, Italy

H. Eric Tseng Ford Motor Company, Dearborn, MI, USA

Valerio Turri Electrical Engineering and Computer Science, KTH – Royal Institute of Technology, Stockholm, Sweden

Roberto G. Valenti Advanced Research and Technology Office, The MathWorks, Inc., Natick, MA, USA

Kyriakos G. Vamvoudakis The Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Arjan van der Schaft Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, Jan C. Willems Center for Systems and Control, University of Groningen, Groningen, The Netherlands

O. Arda Vanli Department of Industrial and Manufacturing Engineering, High Performance Materials Institute, Florida A&M University, Florida State University, Tallahassee, FL, USA

P. Varaiya University of California, Berkeley, Berkeley, CA, USA

Andreas Varga Gilching, Germany

Paul Van Dooren ICTEAM: Department of Mathematical Engineering, Catholic University of Louvain, Louvain-la-Neuve, Belgium

Rafael Vazquez Department of Aerospace Engineering, Universidad de Sevilla, Sevilla, Spain

Michel Verhaegen Delft Center for Systems and Control, Delft University, Delft, The Netherlands

Mathukumalli Vidyasagar University of Texas at Dallas, Richardson, TX, USA

Luigi Villani Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, Università degli Studi di Napoli Federico II, Napoli, Italy

Richard Vinter Imperial College, London, UK

N. Vitiello The Biorobotics Institute, Pisa, Italy

IRCCS Fondazione Don Carlo Gnocchi, Milan, Italy

Department of Excellence in Robotics and AI, Pisa, Italy

Vijay Vittal Arizona State University, Tempe, AZ, USA

Antonio Visioli Dipartimento di Ingegneria Meccanica e Industriale, University of Brescia, Brescia, Italy

Costas Vournas School of Electrical and Computer Engineering, National Technical University of Athens, Zografou, Greece

Maria Vrakopoulou Electrical and Electronic Engineering, University of Melbourne, Melbourne, VIC, Australia

Samir Wadhwania Department of Aeronautics and Astronautics, Aerospace Controls Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA

Steffen Waldherr Institute for Automation Engineering, Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany

Aiping Wang Department of Electronic and Information Engineering, Bozhou University, Bozhou, Anhui, PR China

Dan Wang Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong, China

Fred Wang University of Tennessee, Knoxville, TN, USA

Hong Wang Energy and Transportation Science, Oak Ridge National Laboratory, Oak Ridge, TN, USA

The University of Manchester, Manchester, UK

Yue Wang Department of Mechanical Engineering, Clemson University, Clemson, SC, USA

Zhikui Wang Cupertino, CA, USA

Yorai Wardi School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

John M. Wassick The Dow Chemical Company, Midland, MI, USA

James Weimer University of Pennsylvania, Philadelphia, PA, USA

Jin Wen Department of Civil, Architectural, and Environmental Engineering, Drexel University, Philadelphia, PA, USA

John T. Wen Electrical, Computer, and Systems Engineering (ECSE), Rensselaer Polytechnic Institute, Troy, NY, USA

Paul J. Werbos National Science Foundation, Arlington, VA, USA

Kelilah L. Wolkowicz Harvard John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA

Wing Shing Wong Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, China

W.M. Wonham Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada

Lihua Xie School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

Naoki Yamamoto Department of Applied Physics and Physico-Informatics, Keio University, Yokohama, Japan

Yutaka Yamamoto Graduate School of Informatics, Kyoto University, Kyoto, Japan

Katsu Yamane Honda Research Institute, Inc., San Jose, CA, USA

Hong Ye The Mathworks, Inc., Natick, MA, USA

Ivan Yegorov Department of Mathematics, North Dakota State University, Fargo, ND, USA

Yildiray Yildiz Department of Mechanical Engineering, Bilkent University, Ankara, Turkey

George Yin Department of Mathematics, Wayne State University, Detroit, MI, USA

Serdar Yüksel Department of Mathematics and Statistics, Queen's University, Kingston, ON, Canada

Hasan Zakeri Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Renato Zanetti University of Texas at Austin, Austin, TX, USA

Michael M. Zavlanos Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, USA

Guofeng Zhang Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, China

Qing Zhang Department of Mathematics, The University of Georgia, Athens, GA, USA

Qinghua Zhang Inria, Campus de Beaulieu, Rennes Cedex, France

Di Zhao Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong SAR, China

Kemin Zhou School of Electrical and Automation Engineering, Shangdong University of Science and Technology, Shangdong, China

Liangjia Zhu Department of Computer Science, Stony Brook University, Stony Brook, NY, USA

Qingze Zou Mechanical and Aerospace Engineering Department, Rutgers, The State University of New Jersey, New Brunswick, NJ, USA

Active Power Control of Wind Power Plants for Grid Integration

Lucy Y. Pao
University of Colorado, Boulder, CO, USA

Abstract

Increasing penetrations of intermittent renewable energy sources, such as wind, on the utility grid have led to concerns over the reliability of the grid. One approach for improving grid reliability with increasing wind penetrations is to actively control the real power output of wind turbines and wind power plants. Providing a full range of responses requires derating wind power plants so that there is headroom to both increase and decrease power to provide grid balancing services and stabilizing responses. Results thus far indicate that wind turbines may be able to provide primary frequency control and frequency regulation services more rapidly than conventional power plants.

Keywords

Active power control · Automatic generation control · Frequency regulation · Grid balancing · Grid integration · Primary frequency control · Wind energy

Introduction

Wind penetration levels across the world have increased dramatically, with installed capacity growing at a mean annual rate of 17% over the last decade (Broehl and Asmus 2018). Some nations in Western Europe, particularly Denmark, Ireland, Portugal, Spain, the United Kingdom, and Germany, have seen wind provide more than 18% of their annual electrical energy needs (Wiser and Bolinger 2018). To maintain grid frequency at its nominal value, the electrical generation must equal the electrical load on the grid. This balancing has historically been left up to conventional utilities with synchronous generators, which can vary their active power output by simply varying their fuel input. Grid frequency control is performed across a number of regimes and time scales, with both manual and automatic control commands. Further details can be found in Diaz-Gonzalez et al. (2014) and Ela et al. (2011).

Wind turbines and wind power plants are recognized as having the potential to meet demanding grid stabilizing requirements set by transmission system operators (Ela et al. 2011; Aho et al. 2013a, b; Diaz-Gonzalez et al. 2014; Ela et al. 2014; Fleming et al. 2016). Recent grid code requirements have spurred the development of wind turbine active power control (APC) systems (Diaz-Gonzalez et al. 2014), in some cases mandating wind turbines to participate in

grid frequency regulation and provide stabilizing responses to changes in grid frequency. The ability of wind turbines to provide APC services also allows them to follow forecast-based power production schedules.

For a wind turbine to fully participate in grid frequency control, it must be derated (to P_{derated}) with respect to the maximum power (P_{max}) that can be generated given the available wind, allowing for both increases and decreases in power, if necessary. Wind turbines can derate their power output by pitching their blades to shed aerodynamic power or reducing their generator torque in order to operate at higher-than-optimal rotor speeds. Wind turbines can then respond at different time scales to provide more or less power through pitch control (which can provide a power response within seconds) and generator torque control (which can provide a power response within milliseconds) (Aho et al. 2016).

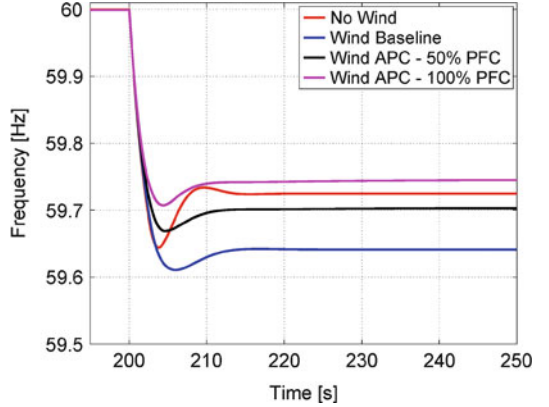
Wind Turbine Inertial and Primary Frequency Control

Inertial and primary frequency control is generally considered to be the first 5–30 s after a frequency event occurs. In this regime, the governors of capable generators actuate, allowing for a temporary increase or decrease in the utilities' power outputs. The primary frequency control (PFC) response provided by conventional synchronous generators can be characterized by a droop curve, which relates fluctuations in grid frequency to a change in power from the utility. For example, a 3% droop curve means that a 3% change in grid frequency yields a 100% change in commanded power.

Although modern wind turbines do not inherently provide inertial or primary frequency control responses because their power electronics impart a buffer between their generators and the grid, such responses can be produced through careful design of the wind turbine control systems. While the physical properties of a conventional synchronous generator yield a static droop characteristic, a wind turbine can be controlled to provide a primary frequency response via either a static or time-varying droop curve.

A time-varying droop curve can be designed to be more aggressive when the magnitude of the rate of change of frequency of the grid is larger.

Figure 1 shows a simulation of a grid response under different scenarios when 5% of the generating capacity suddenly goes offline. When the wind power plant (10% of the generation on the grid) is operating with its normal baseline control system that does not provide APC services, the frequency response is worse than the no-wind scenario, due to the reduced amount of conventional generation in the wind-baseline scenario that can provide power control services. However, compared to both the no-wind and wind-baseline cases, using PFC with a droop curve results in the frequency decline being arrested at a minimum (nadir) frequency f_{nadir} that is closer to the nominal $f_{\text{nom}} = 60$ Hz frequency level; further, the steady-state frequency f_{ss} after the PFC response is also closer to f_{nom} . It is important to prevent the difference $f_{\text{nom}} - f_{\text{nadir}}$ from exceeding a threshold that can



Active Power Control of Wind Power Plants for Grid Integration, Fig. 1 Simulation results showing the capability of wind power plants to provide APC services on a small-scale grid model. The total grid size is 3 GW, and a frequency event is induced due to the sudden active power imbalance when 5% of generation is taken offline at time = 200 s. Each wind power plant is derated to 90% of its rated capacity. The system response with all conventional generation (no wind) is compared to the cases when there are wind power plants on the grid at 10% penetration (i) with a baseline control system (wind baseline) where wind does not provide APC services and (ii) with an APC system (wind APC) that uses a 3% droop curve where either 50% or 100% of the wind power plants provide PFC

lead to underfrequency load shedding (UFLS) or rolling blackouts. The particular threshold varies across utility grids, but the largest such threshold in North America is 1.0 Hz.

Stability issues arising from the altered control algorithms must be analyzed (Buckspan et al. 2013; Wilches-Bernal et al. 2016). The trade-offs between aggressive primary frequency control and resulting structural loads also need to be evaluated carefully. Initial research shows that potential grid support can be achieved while not causing increases in average structural loading (Fleming et al. 2016). Further research is needed to more carefully assess how changes in structural loading affect fatigue damage and operations and maintenance costs.

Wind Turbine Automatic Generation Control

Secondary frequency control, also known as automatic generation control (AGC), occurs on a slower time scale than PFC. AGC commands can be generated from highly damped proportional integral (PI) controllers or logic controllers to regulate grid frequency and are used to control the power output of participating power plants. In many geographical regions, frequency regulation services are compensated through a competitive market, where power plants that provide faster and more accurate AGC command tracking are preferred.

An active power control system that combines both primary and secondary/AGC frequency control capabilities has been detailed in Aho et al. (2013a). Figure 2 presents experimental field test results of this active power controller, in response to prerecorded frequency events, showing how responsive wind turbines can be to both manual derating commands and rapidly changing automatic primary frequency control commands generated via a droop curve. Overall, results indicate that wind turbines can respond more rapidly than conventional power plants. However, increasing the power control and regulation performance of a wind turbine should be carefully considered due to a number of complicating factors, including coupling with existing control loops, a desire to

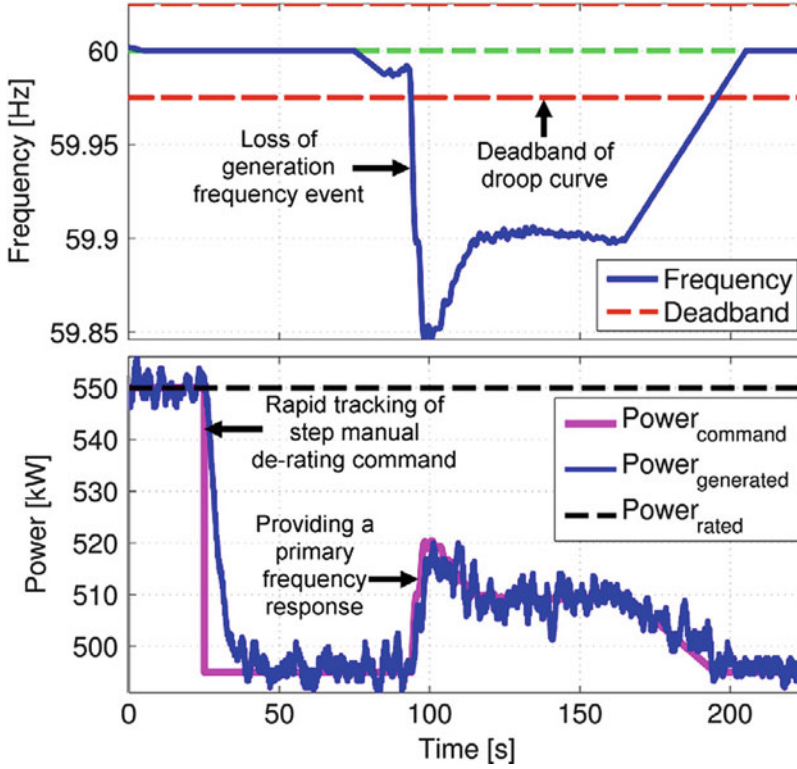
limit actuator usage and structural loading, and wind variability.

Active Power Control of Wind Power Plants

A wind power plant, often referred to as a wind farm, consists of many wind turbines. In wind power plants, wake effects can reduce generation in downstream turbines to less than 60% of the lead turbine (Barthelmie et al. 2009; Porté-Agel et al. 2013). There are many emerging areas of active research, including the modeling of wakes and wake effects and how these models can then be used to coordinate the control of individual turbines so that the overall wind power plant can reliably track the desired power reference command (Knudsen et al. 2015). A wind farm controller can be interconnected with the utility grid, transmission system operator (TSO), and individual turbines as shown in Fig. 3. By properly accounting for the wakes, wind farm controllers can allocate appropriate power reference commands to the individual wind turbines. Individual turbine generator torque and blade pitch controllers, as discussed earlier, can be designed so that each turbine follows the power reference command issued by the wind farm controller. Methods for intelligent, distributed control of entire wind farms to rapidly respond to grid frequency disturbances can significantly reduce frequency deviations and improve recovery speed to such disturbances (Boersma et al. 2019; Shapiro et al. 2017). Note that distributing a wind farm power reference among the operating individual wind turbines, in general, does not have a unique solution; further optimization can lead to structural load alleviation and thus prolongation of the operational lifespan of the individual wind turbines (Vali et al. 2019).

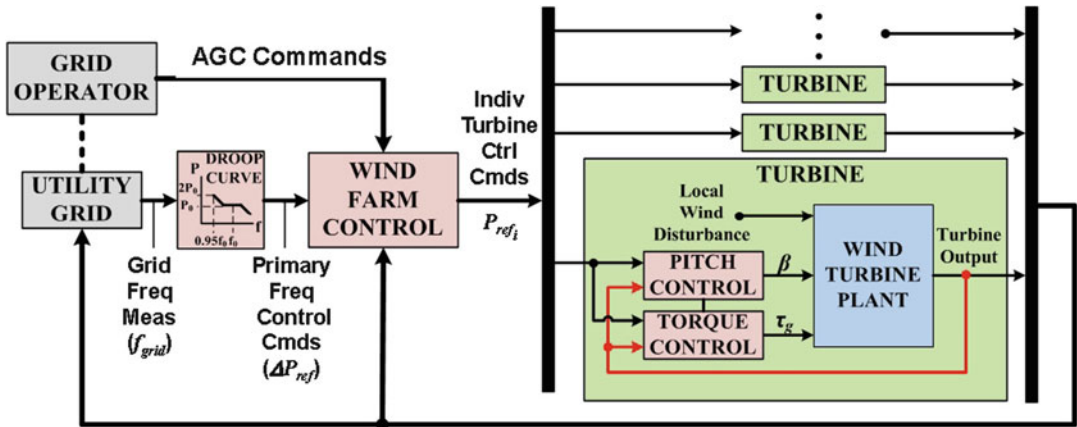
Summary and Future Directions

Ultimately, active power control of wind turbines and wind power plants should be combined with both demand-side management and storage to provide a more comprehensive solution that



Active Power Control of Wind Power Plants for Grid Integration, Fig. 2 The frequency data input and power that is commanded and generated during a field test with a 550 kW research wind turbine at the US National Renewable Energy Laboratory (NREL). The frequency data was recorded on the Electric Reliability Council of Texas (ERCOT) interconnection (data courtesy of Vahan

Gevorgian, NREL). The upper plot shows the grid frequency, which is passed through a 5% droop curve to generate a power command. The high-frequency fluctuations in the generated power would be smoothed when aggregating the power output of an entire wind power plant



Active Power Control of Wind Power Plants for Grid Integration, Fig. 3 Schematic showing the communication and coupling between the wind farm control system, individual wind turbines, utility grid, and the grid operator.

The wind farm controller uses measurements of the utility grid frequency and automatic generation control power command signals from the grid operator to determine a power reference for each turbine in the wind farm

enables balancing electrical generation and electrical load with large penetrations of wind energy on the grid. Demand-side management (Callaway and Hiskens 2011; Palensky and Dietrich 2011) aims to alter the demand in order to mitigate peak electrical loads and hence to maintain sufficient control authority among generating units. As more effective and economical energy storage solutions (Pickard and Abbott 2012; Castillo and Gayme 2014) at the power plant scale are developed, wind (and solar) energy can then be stored when wind (and solar) energy availability is not well matched with electrical demand. Advances in wind forecasting (Pinson 2013; Okumus and Dinler 2016) will also improve wind power forecasts to facilitate more accurate scheduling of larger amounts of wind power on the grid. Finally, considering active power control in conjunction with reactive power and voltage control of wind power plants is also important for the stability and synchronization of the electrical power grid (Jain et al. 2015; Karthikeya and Schutt 2014).

Cross-References

- [Coordination of Distributed Energy Resources for Provision of Ancillary Services: Architectures and Algorithms](#)
- [Wide-Area Control of Power Systems](#)

Recommended Reading

A comprehensive report on active power control that covers topics ranging from control design to power system engineering to economics can be found in Ela et al. (2014) and the references therein.

Bibliography

- Aho J, Buckspan A, Pao L, Fleming P (2013a) An active power control system for wind turbines capable of primary and secondary frequency control for supporting grid reliability. In: Proceedings of the AIAA aerospace sciences meeting, Grapevine, Jan 2013
- Aho J, Buckspan A, Dunne F, Pao LY (2013b) Controlling wind energy for utility grid reliability. *ASME Dyn Syst Control Mag* 1(3):4–12
- Aho J, Fleming P, Pao LY (2016) Active power control of wind turbines for ancillary services: a comparison of pitch and torque control methodologies. In: Proceedings of the American control conference, Boston, July 2016, pp 1407–1412
- Barthelmie RJ, Hansen K, Frandsen ST, Rathmann O, Schepers JG, Schlez W, Phillips J, Rados K, Zervos A, Politis ES, Chaviaropoulos PK (2009) Modelling and measuring flow and wind turbine wakes in large wind farms offshore. *Wind Energy* 12:431–444
- Boersma S, Doekemeijer BM, Siniscalchi-Minna S, van Wingerden JW (2019) A constrained wind farm controller providing secondary frequency regulation: an LES study. *Renew Energy* 134:639–652
- Broehl J, Asmus P (2018) World wind energy market update 2018. Navigant Research, Sep 2018
- Buckspan A, Pao L, Aho J, Fleming P (2013) Stability analysis of a wind turbine active power control system. In: Proceedings of the American control conference, Washington, DC, June 2013, pp 1420–1425
- Callaway DS, Hiskens IA (2011) Achieving controllability of electric loads. *Proc IEEE* 99(1):184–199
- Castillo A, Gayme DF (2014) Grid-scale energy storage applications in renewable energy integration: a survey. *Energy Convers Manag* 87:885–894
- Díaz-Gonzalez F, Hau M, Sumper A, Gomis-Bellmunt O (2014) Participation of wind power plants in system frequency control: review of grid code requirements and control methods. *Renew Sust Energ Rev* 34: 551–564
- Ela E, Milligan M, Kirby B (2011) Operating reserves and variable generation. Technical report, National Renewable Energy Laboratory, NREL/TP-5500-51928
- Ela E, Gevorgian V, Fleming P, Zhang YC, Singh M, Muljadi E, Scholbrock A, Aho J, Buckspan A, Pao L, Singhvi V, Tuohy A, Pourbeik P, Brooks D, Bhatt N (2014) Active power controls from wind power: bridging the gaps. Technical report, National Renewable Energy Laboratory, NREL/TP-5D00-60574, Jan 2014
- Fleming P, Aho J, Buckspan A, Ela E, Zhang Y, Gevorgian V, Scholbrock A, Pao L, Damiani R (2016) Effects of power reserve control on wind turbine structural loading. *Wind Energy* 19(3):453–469
- Jain B, Jain S, Nema RK (2015) Control strategies of grid interfaced wind energy conversion system: an overview. *Renew Sust Energ Rev* 47:983–996
- Karthikeya BR, Schutt RJ (2014) Overview of wind park control strategies. *IEEE Trans Sust Energ* 5(2): 416–422
- Knudsen T, Bak T, Svenstrup M (2015) Survey of wind farm control – power and fatigue optimization. *Wind Energy* 18(8):1333–1351
- Okumus I, Dinler A (2016) Current status of wind energy forecasting and a hybrid method for hourly predictions. *Energy Convers Manag* 123:362–371
- Palensky P, Dietrich D (2011) Demand-side management: demand response, intelligent energy systems, and smart loads. *IEEE Trans Ind Inform* 7(3):381–388
- Pickard WF, Abbott D (eds) (2012) The intermittency challenge: massive energy storage in a sustainable future. *Proc IEEE* 100(2):317–321. Special issue

- Pinson P (2013) Wind energy: forecasting challenges for its operational management. *Stat Sci* 28(4):564–585
- Porté-Agel F, Wu Y-T, Chen C-H (2013) A numerical study of the effects of wind direction on turbine wakes and power losses in a large wind farm. *Energies* 6:5297–5313
- Shapiro CR, Bauweraerts P, Meyers J, Meneveau C, Gayme DF (2017) Model-based receding horizon control of wind farms for secondary frequency regulation. *Wind Energy* 20(7):1261–1275
- Vali M, Petrovic V, Steinfeld G, Pao LY, Kühn M (2019) An active power control approach for wake-induced load alleviation in a fully developed wind farm boundary layer. *Wind Energy Sci* 4(1):139–161
- Wilches-Bernal F, Chow JH, Sanchez-Gasca JJ (2016) A fundamental study of applying wind turbines for power system frequency control. *IEEE Trans Power Syst* 31(2):1496–1505
- Wiser R, Bolinger M (2018) 2017 Wind technologies market report. Lawrence Berkeley National Laboratory Report, Aug 2018

Adaptive Control of Linear Time-Invariant Systems

Petros A. Ioannou

University of Southern California, Los Angeles, CA, USA

Abstract

Adaptive control of linear time-invariant (LTI) systems deals with the control of LTI systems whose parameters are constant but otherwise completely unknown. In some cases, large norm bounds as to where the unknown parameters are located in the parameter space are also assumed to be known. In general, adaptive control deals with LTI plants which cannot be controlled with fixed gain controllers, i.e., nonadaptive control methods, and their parameters even though assumed constant for design and analysis purposes may change over time in an unpredictable manner. Most of the adaptive control approaches for LTI systems use the so-called certainty equivalence principle where a control law motivated from the known parameter case is combined with an adaptive law for estimating on line the unknown parameters. The control law could be associated with different control

objectives and the adaptive law with different parameter estimation techniques. These combinations give rise to a wide class of adaptive control schemes. The two popular control objectives that led to a wide range of adaptive control schemes include model reference adaptive control (MRAC) and adaptive pole placement control (APPC). In MRAC, the control objective is for the plant output to track the output of a reference model, designed to represent the desired properties of the plant, for any reference input signal. APPC is more general and is based on control laws whose objective is to set the poles of the closed loop at desired locations chosen based on performance requirements. Another class of adaptive controllers for LTI systems that involves ideas from MRAC and APPC is based on multiple models, search methods, and switching logic. In this class of schemes, the unknown parameter space is partitioned to smaller subsets. For each subset, a parameter estimator or a stabilizing controller is designed or a combination of the two. The problem then is to identify which subset in the parameter space the unknown plant model belongs to and/or which controller is a stabilizing one and meets the control objective. A switching logic is designed based on different considerations to identify the most appropriate plant model or controller from the list of candidate plant models and/or controllers. In this entry, we briefly describe the above approaches to adaptive control for LTI systems.

Keywords

Adaptive pole placement control · Direct MRAC · Indirect MRAC · LTI systems · Model reference adaptive control · Robust adaptive control

Model Reference Adaptive Control

In model reference control (MRC), the desired plant behavior is described by a reference model which is simply an LTI system with a transfer function $W_m(s)$ and is driven by a reference

input. The controller transfer function $C(s, \theta^*)$, where θ^* is a vector with the coefficients of $C(s)$, is then developed so that the closed-loop plant has a transfer function equal to $W_m(s)$. This transfer function matching guarantees that the plant will match the reference model response for any reference input signal. In this case the plant transfer function $G_p(s, \theta_p^*)$, where θ_p^* is a vector with all the coefficients of $G_p(s)$, together with the controller transfer function $C(s, \theta^*)$ should lead to a closed-loop transfer function from the reference input r to the plant output y_p that is equal to $W_m(s)$, i.e.,

$$\frac{y_p(s)}{r(s)} = W_m(s) = \frac{y_m(s)}{r(s)}, \quad (1)$$

where y_m is the output of the reference model. For this transfer matching to be possible, $G_p(s)$ and $W_m(s)$ have to satisfy certain assumptions. These assumptions enable the calculation of the controller parameter vector θ^* as

$$\theta^* = F(\theta_p^*), \quad (2)$$

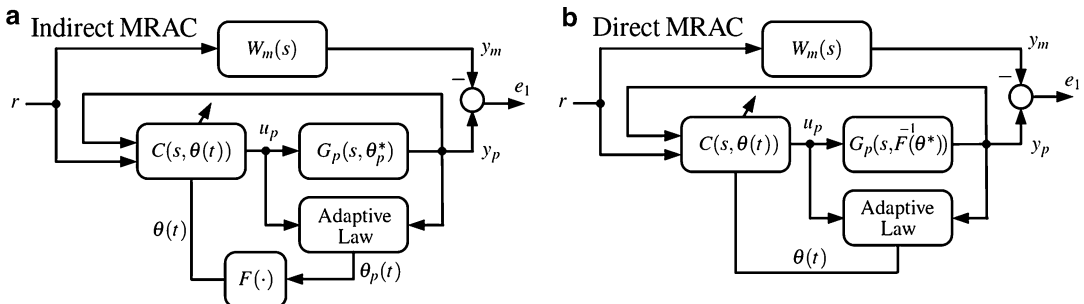
where F is a function of the plant parameters θ_p^* , to satisfy the matching equation (1). The function in (2) has a special form in the case of MRC that allows the design of both direct and indirect MRAC. For more general classes of controller structures, this is not possible in general as the function F is nonlinear. This transfer function matching guarantees that the tracking error $e_1 = y_p - y_m$ converges to zero for any given reference input signal r . If the plant parameter vector θ_p^* is known, then the controller parameters θ^* can be calculated using (2), and

the controller $C(s, \theta^*)$ can be implemented. We are considering the case where θ_p^* is unknown. In this case, the use of the certainty equivalence (CE) approach, (Astrom and Wittenmark 1995; Egardt 1979; Ioannou and Fidan 2006; Ioannou and Kokotovic 1983; Ioannou and Sun 1996; Landau et al. 1998; Morse 1996; Landau 1979; Narendra and Annaswamy 1989; Narendra and Balakrishnan 1997; Sastry and Bodson 1989; Stefanovic and Safonov 2011; Tao 2003) where the unknown parameters are replaced with their estimates, leads to the adaptive control scheme referred to as *indirect MRAC*, shown in Fig. 1a.

The unknown plant parameter vector θ_p^* is estimated at each time t denoted by $\theta_p(t)$, using an online parameter estimator referred to as adaptive law. The plant parameter estimate $\theta_p(t)$ at each time t is then used to calculate the controller parameter vector $\theta(t) = F(\theta_p(t))$ used in the controller $C(s, \theta)$. This class of MRAC is called indirect MRAC, because the controller parameters are not updated directly, but calculated at each time t using the estimated plant parameters. Another way of designing MRAC schemes is to parameterize the plant transfer function in terms of the desired controller parameter vector θ^* . This is possible in the MRC case, because the structure of the MRC law is such that we can use (2) to write

$$\theta_p^* = F^{-1}(\theta^*), \quad (3)$$

where F^{-1} is the inverse of the mapping $F(\cdot)$, and then express $G_p(s, \theta_p^*) = G_p(s, F^{-1}(\theta^*)) = \bar{G}_p(s, \theta^*)$. The adaptive law for estimating θ^* online can now be developed by using



Adaptive Control of Linear Time-Invariant Systems, Fig. 1 Structure of (a) indirect MRAC, (b) direct MRAC

$y_p = \bar{G}_p(s, \theta^*)u_p$ to obtain a parametric model that is appropriate for estimating the controller vector θ^* as the unknown parameter vector. The MRAC can then be developed using the CE approach as shown in Fig. 1b. In this case, the controller parameter $\theta(t)$ is updated directly without any intermediate calculations, and for this reason, the scheme is called *direct MRAC*.

The division of MRAC to indirect and direct is, in general, unique to MRC structures, and it is possible due to the fact that the inverse maps in (2) and (3) exist which is a direct consequence of the control objective and the assumptions the plant and reference model are required to satisfy for the control law to exist. These assumptions are summarized below:

Plant Assumptions: $G_p(s)$ is minimum phase, i.e., has stable zeros, its relative degree, $n^* = \text{number of poles} - \text{number of zeros}$, is known and an upper bound n on its order is also known. In addition, the sign of its high-frequency gain is known even though it can be relaxed with additional complexity.

Reference Model Assumptions: $W_m(s)$ has stable poles and zeros, its relative degree is equal to n^* that of the plant, and its order is equal or less to the one assumed for the plant, i.e., of n .

The above assumptions are also used to meet the control objective in the case of known parameters, and therefore the minimum phase and relative degree assumptions are characteristics of the control objective and do not arise because of adaptive control considerations. The relative degree matching is used to avoid the need to differentiate signals in the control law. The minimum phase assumption comes from the fact that the only way for the control law to force the closed-loop plant transfer function to be equal to that of the reference model is to cancel the zeros of the plant using feedback and replace them with those of the reference model using a feedforward term. Such zero pole cancelations are possible if the zeros are stable, i.e., the plant is minimum phase; otherwise stability cannot be guaranteed for nonzero initial conditions and/or inexact cancelations.

The design of MRAC in Fig. 1 has additional variations depending on how the adaptive law is designed. If the reference model is chosen to be strictly positive real (SPR) which limits its transfer function and that of the plant to have relative degree 1, the derivation of adaptive law and stability analysis is fairly straightforward, and for this reason, this class of MRAC schemes attracted a lot of interest. As the relative degree changes to 2, the design becomes more complex as in order to use the SPR property, the CE control law has to be modified by adding an extra nonlinear term. The stability analysis remains to be simple as a single Lyapunov function can be used to establish stability. As the relative degree increases further, the design complexity increases by requiring the addition of more nonlinear terms in the CE control law (Ioannou and Fidan 2006; Ioannou and Sun 1996). The simplicity of using a single Lyapunov function analysis for stability remains however. This approach covers both direct and indirect MRAC and lead to adaptive laws which contain no normalization signals (Ioannou and Fidan 2006; Ioannou and Sun 1996). A more straightforward design approach is based on the CE principle which separates the control design from the parameter estimation part and leads to a much wider class of MRAC which can be direct or indirect. In this case, the adaptive laws need to be normalized for stability, and the analysis is far more complicated than the approach based on SPR with no normalization. An example of such a direct MRAC scheme for the case of known sign of the high-frequency gain which is assumed to be positive for both plant and reference model is listed below:

Control law:

$$u_p = \theta_1^T(t) \frac{\alpha(s)}{\Lambda(s)} u_p + \theta_2^T \frac{\alpha(s)}{\Lambda(s)} y_p + \theta_3(t) y_p + c_0(t) r = \theta^T(t) \omega, \quad (4)$$

where $\alpha \triangleq \alpha_{n-2}(s) = [s^{n-2}, s^{n-3}, \dots, s, 1]^T$ for $n \geq 2$, and $\alpha(s) \triangleq 0$ for $n = 1$, and $\Lambda(s)$ is a monic polynomial with stable roots and degree $n - 1$ having numerator of $W_m(s)$ as a factor.

Adaptive law:

$$\dot{\theta} = \Gamma \varepsilon \phi, \quad (5)$$

where Γ is a positive definite matrix referred to as the adaptive gain and $\dot{\rho} = \gamma \varepsilon \xi$, $\varepsilon = \frac{e_1 - \rho \xi}{m_s^2}$, $m_s^2 = 1 + \phi^T \phi + u_f^2$, $\xi = \theta^T \phi + u_f$, $\phi = -W_m(s)\omega$, and $u_f = W_m(s)u_p$.

The stability properties of the above direct MRAC scheme which are typical for all classes of MRAC are the following (Ioannou and Fidan 2006; Ioannou and Sun 1996): (i) All signals in the closed-loop plant are bounded, and the tracking error e_1 converges to zero asymptotically and (ii) if the plant transfer function contains no zero pole cancelations and r is sufficiently rich of order $2n$, i.e., it contains at least n distinct frequencies, then the parameter error $|\tilde{\theta}| = |\theta - \theta^*|$ and the tracking error e_1 converge to zero exponentially fast.

Adaptive Pole Placement Control

Let us consider the SISO LTI plant:

$$y_p = G_p(s)u_p, \quad G_p(s) = \frac{Z_p(s)}{R_p(s)}, \quad (6)$$

where $G_p(s)$ is proper and $R_p(s)$ is a monic polynomial. The control objective is to choose the plant input u_p so that the closed-loop poles are assigned to those of a given monic Hurwitz polynomial $A^*(s)$, and y_p is required to follow a certain class of reference signals y_m assumed to satisfy $Q_m(s)y_m = 0$ where $Q_m(s)$ is known as the internal model of y_m and is designed to have all roots in $\text{Re}\{s\} \leq 0$ with no repeated roots on the $j\omega$ -axis. The polynomial $A^*(s)$, referred to as the desired closed-loop characteristic polynomial, is chosen based on the closed-loop performance requirements. To meet the control objective, we make the following assumptions about the plant:

P1. $G_p(s)$ is strictly proper with known degree, and $R_p(s)$ is a monic polynomial whose degree n is known and $Q_m(s)Z_p(s)$ and $R_p(s)$ are coprime.

Assumption P1 allows Z_p and R_p to be non-Hurwitz in contrast to the MRAC case where Z_p is required to be Hurwitz.

The design of the APPC scheme is based on the CE principle. The plant parameters are estimated at each time t and used to calculate the controller parameters that meet the control objective for the estimated plant as follows: Using (6) the plant equation can be expressed in a form convenient for parameter estimation via the model (Goodwin and Sin 1984; Ioannou and Fidan 2006; Ioannou and Sun 1996):

$$z = \theta_p^* \phi,$$

where $z = \frac{s^n}{\Lambda_p(s)} y_p$, $\theta_p^* = [\theta_b^{*T}, \theta_a^{*T}]^T$, $\phi = [\frac{\alpha_{n-1}^T(s)}{\Lambda_p(s)} u_p, -\frac{\alpha_{n-1}^T(s)}{\Lambda_p(s)} y_p]^T$, $\alpha_{n-1} = [s^{n-1}, \dots, s, 1]^T$, $\theta_a^* = [a_{n-1}, \dots, a_0]^T$, $\theta_b^* = [b_{n-1}, \dots, b_0]^T$, and $\Lambda_p(s)$ is a Hurwitz monic design polynomial. As an example of a parameter estimation algorithm, we consider the gradient algorithm

$$\dot{\theta}_p = \Gamma \varepsilon \phi, \quad \varepsilon = \frac{z - \theta_p^T \phi}{m_s^2}, \quad m_s^2 = 1 + \phi^T \phi, \quad (7)$$

where $\Gamma = \Gamma^T > 0$ is the adaptive gain and $\theta_p = [\hat{b}_{n-1}, \dots, \hat{b}_0, \hat{a}_{n-1}, \dots, \hat{a}_0]^T$ are the estimated plant parameters which can be used to form the estimated plant polynomials $\hat{R}_p(s, t) = s^n + \hat{a}_{n-1}(t)s^{n-1} + \dots + \hat{a}_1(t)s + \hat{a}_0(t)$ and $\hat{Z}_p(s, t) = \hat{b}_{n-1}(t)s^{n-1} + \dots + \hat{b}_1(t)s + \hat{b}_0(t)$ of $R_p(s)$ and $Z_p(s)$, respectively, at each time t . The adaptive control law is given as

$$u_p = \left(\Lambda(s) - \hat{L}(s, t) Q_m(s) \right) \frac{1}{\Lambda(s)} u_p - \hat{P}(s, t) \frac{1}{\Lambda(s)} (y_p - y_m), \quad (8)$$

where $\hat{L}(s, t)$ and $\hat{P}(s, t)$ are obtained by solving the polynomial equation $\hat{L}(s, t) \cdot Q_m(s) \cdot \hat{R}_p(s, t) + \hat{P}(s, t) \cdot \hat{Z}_p(s, t) = A^*(s)$ at each time t . The operation $X(s, t) \cdot Y(s, t)$ denotes a multiplication of polynomials where s is simply treated as a variable. The existence and

uniqueness of $\hat{L}(s, t)$ and $\hat{P}(s, t)$ is guaranteed provided $\hat{R}_p(s, t) \cdot Q_m(s)$ and $\hat{Z}_p(s, t)$ are coprime at each frozen time t . The adaptive laws that generate the coefficients of $\hat{R}_p(s, t)$ and $\hat{Z}_p(s, t)$ cannot guarantee this property, which means that at certain points in time, the solution $\hat{L}(s, t)$, $\hat{P}(s, t)$ may not exist. This problem is known as the stabilizability problem in indirect APPC and further modifications are needed in order to handle it (Goodwin and Sin 1984; Ioannou and Fidan 2006; Ioannou and Sun 1996). Assuming that the stabilizability condition holds at each time t , it can be shown (Goodwin and Sin 1984; Ioannou and Fidan 2006; Ioannou and Sun 1996) that all signals are bounded and the tracking error converges to zero with time. Other indirect adaptive pole placement control schemes include adaptive linear quadratic (Ioannou and Fidan 2006; Ioannou and Sun 1996). In principle any nonadaptive control scheme can be made adaptive by replacing the unknown parameters with their estimates in the calculation of the controller parameters. The design of direct APPC schemes is not possible in general as the map between the plant and controller parameters is nonlinear, and the plant parameters cannot be expressed as a convenient function of the controller parameters. This prevents parametrization of the plant transfer function with respect to the controller parameters as done in the case of MRC. In special cases where such parametrization is possible such as in MRAC which can be viewed as a special case of APPC, the design of direct APPC is possible. Chapters on ► [Adaptive Control: Overview](#), ► [Robust Adaptive Control](#), and ► [History of Adaptive Control](#) provide additional information regarding MRAC and APPC.

the adaptive law cannot guarantee such a property, an approach emerged that involves the precalculation of a set of controllers based on the partitioning of the plant parameter space. The problem then becomes one of identifying which one of the controllers is the most appropriate one. The switching to the “best” possible controller could be based on some logic that is driven by some cost index, multiple estimation models, and other techniques (Fekri et al. 2007; Hespanha et al. 2003; Kuipers and Ioannou 2010; Morse 1996; Narendra and Balakrishnan 1997; Stefanovic and Safonov 2011). One of the drawbacks of this approach is that it is difficult if at all possible to find a finite set of stabilizing controllers that cover the whole unknown parameter space especially for high-order plants. If found its dimension may be so large that makes it impractical. Another drawback that is present in all adaptive schemes is that in the absence of persistently exciting signals which guarantee that the input/output data have sufficient information about the unknown plant parameters, there is no guarantee that the controller the scheme converged to is indeed a stabilizing one. In other words, if switching is disengaged or the adaptive law is switched off, there is no guarantee that a small disturbance will not drive the corresponding LTI scheme unstable. Nevertheless these techniques allow the incorporation of well-established robust control techniques in designing a priori the set of controller candidates. The problem is that if the plant parameters change in a way not accounted for a priori, no controller from the set may be stabilizing leading to an unstable system.

Robust Adaptive Control

Search Methods, Multiple Models, and Switching Schemes

One of the drawbacks of APPC is the stabilizability condition which requires the estimated plant at each time t to satisfy the detectability and stabilizability condition that is necessary for the controller parameters to exist. Since

The MRAC and APPC schemes presented above are designed for LTI systems. Due to the adaptive law, the closed-loop system is no longer LTI but nonlinear and time varying. It has been shown using simple examples that the pure integral action of the adaptive law could cause parameter drift in the presence of small disturbances and/or unmodeled dynamics (Ioannou and Fidan

2006; Ioannou and Kokotovic 1983; Ioannou and Sun 1996) which could then excite the unmodeled dynamics and lead to instability. Modifications to counteract these possible instabilities led to the field of robust adaptive control whose focus was to modify the adaptive law in order to guarantee robustness with respect to disturbances, unmodeled dynamics, time-varying parameters, classes of nonlinearities, etc., by using techniques such as normalizing signals, projection, fixed and switching sigma modification, etc.

Cross-References

- ▶ [Adaptive Control: Overview](#)
- ▶ [History of Adaptive Control](#)
- ▶ [Model Reference Adaptive Control](#)
- ▶ [Robust Adaptive Control](#)
- ▶ [Switching Adaptive Control](#)

Bibliography

- Astrom K, Wittenmark B (1995) Adaptive control. Addison-Wesley, Reading
- Egardt B (1979) Stability of adaptive controllers. Springer, New York
- Fekri S, Athans M, Pascoal A (2007) Robust multiple model adaptive control (RMMAC): a case study. *Int J Adapt Control Signal Process* 21(1):1–30
- Goodwin G, Sin K (1984) Adaptive filtering prediction and control. Prentice-Hall, Englewood Cliffs
- Hespanha JP, Liberzon D, Morse A (2003) Hysteresis-based switching algorithms for supervisory control of uncertain systems. *Automatica* 39(2):263–272
- Ioannou P, Fidan B (2006) Adaptive control tutorial. SIAM, Philadelphia
- Ioannou P, Kokotovic P (1983) Adaptive systems with reduced models. Springer, Berlin/New York
- Ioannou P, Sun J (1996) Robust adaptive control. Prentice-Hall, Upper Saddle River
- Kuipers M, Ioannou P (2010) Multiple model adaptive control with mixing. *IEEE Trans Autom Control* 55(8):1822–1836
- Landau Y (1979) Adaptive control: the model reference approach. Marcel Dekker, New York
- Landau I, Lozano R, M'Saad M (1998) Adaptive control. Springer, New York
- Morse A (1996) Supervisory control of families of linear set-point controllers part I: exact matching. *IEEE Trans Autom Control* 41(10):1413–1431
- Narendra K, Annaswamy A (1989) Stable adaptive systems. Prentice Hall, Englewood Cliffs
- Narendra K, Balakrishnan J (1997) Adaptive control using multiple models. *IEEE Trans Autom Control* 42(2):171–187
- Sastry S, Bodson M (1989) Adaptive control: stability, convergence and robustness. Prentice Hall, Englewood Cliffs
- Stefanovic M, Safonov M (2011) Safe adaptive control: data-driven stability analysis and robust synthesis. Lecture notes in control and information sciences, vol 405. Springer, Berlin
- Tao G (2003) Adaptive control design and analysis. Wiley-Interscience, Hoboken

Adaptive Control of PDEs

Henrik Anfinssen and Ole Morten Aamo
Department of Engineering Cybernetics, NTNU,
Trondheim, Norway

Abstract

The infinite-dimensional backstepping method has recently been used for adaptive control of partial differential equations (PDEs). We will in this article briefly explain the main ideas for the three most commonly used methods for backstepping-based adaptive control of PDEs: Lyapunov-based design, identifier-based design, and swapping-based design. Swapping-based design is also demonstrated on a simple, scalar hyperbolic PDE with an uncertain in-domain parameter, clearly showing all the steps involved in using this method.

Keywords

Partial differential equations · Hyperbolic systems · Stabilization · Adaptive control · Parameter estimation · Infinite-dimensional backstepping

Introduction

The backstepping method has for the last decade and a half been used with great success for controller and observer design of partial differential equations (PDEs). Starting with nonadaptive results for parabolic PDEs (Liu

2003; Smyshlyaev and Krstić 2004, 2005; Krstić and Smyshlyaev 2008c), the method was extended to hyperbolic PDEs in Krstić and Smyshlyaev (2008b), where a controller for a scalar 1-D system was designed. Extensions to coupled hyperbolic PDEs followed in Vazquez et al. (2011), Di Meglio et al. (2013), and Hu et al. (2016). The backstepping approach offers a systematic way of designing controllers and observers for linear PDEs. One of its key strengths is that state feedback control laws and state observers can be derived for the infinite-dimensional system directly, with all analysis carried out in the infinite-dimensional framework, avoiding any artifacts caused by spatial discretization.

Backstepping has turned out to be well suited for dealing with uncertain systems by adaptive designs. The first result appeared in Smyshlyaev and Krstić (2006) where a parabolic PDE with an uncertain parameter is stabilized by backstepping. Extensions in several directions subsequently followed (Krstić and Smyshlyaev 2008a; Smyshlyaev and Krstić 2007a, b), culminating in the book *Adaptive Control of Parabolic PDEs* (Smyshlyaev and Krstić 2010). Most of the ideas developed for parabolic PDEs carry over to the hyperbolic case, with the first result appearing in Bernard and Krstić (2014) for a scalar equation. Extensions to coupled PDEs followed in Anfinssen and Aamo (2017a, b, c) and Yu et al. (2017), to mention a few.

Methods for Adaptive Control of PDEs

In Smyshlyaev and Krstić (2010) and Anfinssen and Aamo (2019), three main types of control design methods for adaptive control of PDEs are mentioned. These are

1. *Lyapunov-based design*: This approach directly addresses the problem of closed-loop stability, with the controller and adaptive law designed simultaneously from Lyapunov functions. It has been used for adaptive control design for parabolic PDEs in Krstić and Smyshlyaev (2008a) and for a scalar

hyperbolic PDE in Xu and Liu (2016). The Lyapunov method produces the simplest adaptive law in the sense of the controller's dynamical order, but the stability proof quickly becomes complicated as the systems get more complex. For this reason, no results extending those of Xu and Liu (2016) to coupled PDEs have appeared in the literature.

2. *Identifier-based design*: An identifier is usually a copy of the system dynamics with uncertain system parameters replaced by estimates and with error signals in terms of differences between measured and computed signals injected. Adaptive laws are chosen so that boundedness of the identifier error is ensured, before a control law is designed with the aim of stabilizing the identifier. Boundedness of the original system follows from boundedness of the identifier and the identifier error. Identifier-based designs have been developed for parabolic PDEs in Smyshlyaev and Krstić (2007a) and hyperbolic PDEs in Anfinssen and Aamo (2016a) and Anfinssen and Aamo (2018). The control law is designed independently of the adaptive laws, and hence the identifier-based method is based on certainty equivalence (CE). Since the method employs a copy of the system dynamics along with the adaptive laws, the dynamical order of the controller is larger than the dynamical order of the system.
3. *Swapping-based design*: In swapping-based designs, filters are carefully constructed so that the system states can be statically expressed in terms of the filter states, the unknown parameters and some error terms that are proved to converge to zero. The key property of the static parameterization is that the uncertain parameters appear linearly, so that well-established parameter identification laws such as the gradient law or the least squares method can be used to generate estimates. It also allows for normalization, so that the update laws remain bounded regardless of the boundedness of the system states. Adaptive estimates of the states can then be generated from the filters and parameter estimates by substituting the uncertain

parameters in the static parameterization with their respective estimates. A controller is designed for stabilization of the adaptive state estimates, meaning that this method, like the identifier-based method, is based on the certainty equivalence principle. The number of filters required when using this method typically equals the number of unknown parameters plus one, rendering the swapping-based adaptive controller of higher dynamical order than Lyapunov-based or identifier-based designs. Nevertheless, swapping-based design is the one most prominently used for adaptive control of hyperbolic PDEs since it extends to more complex systems in a natural way, as illustrated by the series of papers (Bernard and Krstić 2014; Anfinson and Aamo 2016b, 2017a, d).

Application of the Swapping Design Method

To give the reader a flavor of what adaptive control designs for PDEs involve, we will demonstrate swapping-based design for stabilization of a simple linear hyperbolic PDE. First, we introduce some notation. We will by $\|z\|$ and $\|z\|_\infty$ denote the L_2 -norm and ∞ -norm respectively, that is $\|z\| = \sqrt{\int_0^1 z^2(x)dx}$ and $\|z\|_\infty = \sup_{x \in [0,1]} |z(x)|$ for some function $z(x)$ defined for $x \in [0, 1]$. For a time-varying, real signal $f(t)$, the following vector spaces are used: $f \in \mathcal{L}_p \Leftrightarrow (\int_0^\infty |f(t)|^p dt)^{\frac{1}{p}} < \infty$ for $1 \leq p < \infty$, and $f \in \mathcal{L}_\infty \Leftrightarrow \sup_{t \geq 0} |f(t)| < \infty$. For two functions $u(x), v(x)$ on $x \in [0, 1]$, we define the operator $\equiv$ as $u \equiv v \Leftrightarrow \|u - v\|_\infty = 0$, $u \equiv 0 \Leftrightarrow \|u\|_\infty = 0$. Lastly, the following space is defined $\mathcal{B}([0, 1]) = \{u(x) \mid \|u\|_\infty < \infty\}$.

Consider the linear hyperbolic PDE

$$u_t(x, t) - u_x(x, t) = \theta u(0, t), \quad (1a)$$

$$u(1, t) = U(t), \quad (1b)$$

$$u(x, 0) = u_0(x) \quad (1c)$$

where $\theta \in \mathbb{R}$ is an uncertain parameter, while $u_0 \in \mathcal{B}([0, 1])$. Although θ is uncertain, we assume that we have some a priori bound, as formally stated in the following assumption:

Assumption 1 *A bound on θ is known. That is, we are in knowledge of a nonnegative constant $\bar{\theta}$ so that $|\theta| \leq \bar{\theta}$.*

Nonadaptive Control

A finite-time convergent stabilizing nonadaptive controller is Krstić and Smyshlyaev (2008b)

$$U(t) = -\theta \int_0^1 e^{\theta(1-\xi)} u(\xi, t) d\xi. \quad (2)$$

This can be proved using the *backstepping transformation*

$$w(x, t) = T[k, u(t)](x)$$

$$= u(x, t) - \int_0^x k(x - \xi) u(\xi, t) d\xi \quad (3)$$

with inverse

$$\begin{aligned} u(x, t) &= T^{-1}[\theta, w(t)](x) \\ &= w(x, t) - \theta \int_0^x w(\xi, t) d\xi \end{aligned} \quad (4)$$

where

$$k(x) = -\theta e^{\theta x} \quad (5)$$

is the *kernel function*. T maps system (1a) into the *target system*

$$w_t(x, t) - w_x(x, t) = 0, \quad w(x, 0) = w_0(x) \quad (6a)$$

$$w(1, t) = U(t)$$

$$+ \theta \int_0^1 e^{\theta(1-\xi)} u(\xi, t) d\xi. \quad (6b)$$

Choosing the control law $U(t)$ as (2) yields $w(1, t) = 0$, for which it trivially follows that $w \equiv 0$ for $t \geq 1$. From (4), it is clear that $w \equiv 0$ implies $u \equiv 0$.

Adaptive Control

Filter Design and Nonadaptive State Estimate

We introduce the input filter ψ and parameter filter ϕ as

$$\begin{aligned} \psi_t(x, t) - \psi_x(x, t) &= 0, \\ \psi(1, t) &= U(t), \\ \psi(x, 0) &= \psi_0(x) \\ \phi_t(x, t) - \phi_x(x, t) &= 0, \\ \phi(1, t) &= u(0, t), \\ \phi(x, 0) &= \phi_0(x). \end{aligned} \quad (7a) \quad (7b)$$

Using the filters, a *nonadaptive estimate* $\bar{u}$ of the system state u can be generated as

$$\begin{aligned} \bar{u}(x, t) &= \psi(x, t) + \theta F[\phi(t)](x), \\ F[\phi(t)](x) &= \int_x^1 \phi(1 - \xi + x, t) d\xi. \end{aligned} \quad (8)$$

Notice that $\bar{u}$ is linear in the uncertain parameter θ and that F , ϕ , and ψ are known.

Lemma 1 Consider system (1a), filters (7), and the variable $\bar{u}$ generated from (8). For $t \geq 1$, we have $\bar{u} \equiv u$.

Proof. Defining, $e(x, t) = u(x, t) - \bar{u}(x, t) = u(x, t) - \psi(x, t) - \theta F[\phi(t)](x)$, it is straightforward to show, using the dynamics (1a) and (7), that e satisfies the dynamics

$$\begin{aligned} e_t(x, t) - e_x(x, t) &= 0, \\ e(1, t) &= 0, e(x, 0) = e_0(x) \end{aligned} \quad (9)$$

and hence $e \equiv 0$ and $\bar{u} \equiv u$ for $t \geq 1$. $\square$

Adaptive Law

From the linear relationship $u(x, t) = \bar{u}(x, t) = \psi(x, t) + \theta F[\phi(t)](x)$ for $t \geq 1$ ensured by Lemma 1, we have $u(0, t) = \psi(0, t) + \theta F[\phi(t)](0)$ from which we propose the gradient adaptive law with normalization

$$\dot{\hat{\theta}}(t) = \begin{cases} \text{proj}_{\bar{\theta}} \left\{ \gamma \frac{\hat{e}(0, t) \int_0^1 \phi(x, t) dx}{1 + \|\phi(t)\|^2}, \hat{\theta}(t) \right\}, & \text{for } t \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad \hat{\theta}(0) = \hat{\theta}_0, \quad |\hat{\theta}_0| \leq \bar{\theta} \quad (10)$$

for some gain $\gamma > 0$, $\hat{\theta}_0$ chosen inside the feasible domain as given by Assumption 1 and where

$$\hat{e}(x, t) = u(x, t) - \hat{u}(x, t), \quad \hat{u}(x, t) = \psi(x, t) + \hat{\theta}(t) F[\phi(t)](x), \quad (11)$$

and the projection operator is defined as

$$\text{proj}_a(\tau, \omega) = \begin{cases} 0 & \text{if } (\omega = -a \text{ and } \tau \leq 0) \text{ or } (\omega = a \text{ and } \tau \geq 0) \\ \tau & \text{otherwise.} \end{cases} \quad (12)$$

Notice that the *adaptive state estimate* $\hat{u}$ is obtained from the nonadaptive estimate by simply substituting the uncertain parameter θ with its adaptive estimate provided by (10). The following lemma states important properties regarding the adaptive law which are crucial in the closed-loop stability analysis.

Lemma 2 *Subject to Assumption 1, the adaptive law (10) with initial condition $|\hat{\theta}_0| \leq \bar{\theta}$ provides the following signal properties*

$$|\hat{\theta}(t)| \leq \bar{\theta}, \forall t \geq 0 \quad (13a)$$

$$\dot{\hat{\theta}}, \sigma \in \mathcal{L}_2 \cap \mathcal{L}_\infty, \quad \sigma(t) = \frac{\hat{e}(0, t)}{\sqrt{1 + \|\phi(t)\|^2}}. \quad (13b)$$

Proof. Property (13a) follows from the initial condition (10) and the projection operator (Anfinssen and Aamo 2019, Lemma A.1). Consider

$$V(t) = \frac{1}{2\gamma} \tilde{\theta}^2(t), \quad \tilde{\theta}(t) = \theta - \hat{\theta}(t). \quad (14)$$

For $t \in [0, 1)$, we will trivially have $\dot{V}(t) = 0$. For $t \geq 1$, we have, using the property $-\tilde{\theta} \text{proj}_{\bar{\theta}}(\tau, \hat{\theta}) \leq -\tilde{\theta}\tau$ (Anfinssen and Aamo 2019, Lemma A.1), that

$$\dot{V}(t) \leq -\tilde{\theta}(t) \frac{\hat{e}(0, t) \int_0^1 \phi(x, t) dx}{1 + \|\phi(t)\|^2}. \quad (15)$$

From (8), (11) and Lemma 1, we have, for $t \geq 1$ $\hat{e}(0, t) = u(0, t) - \hat{u}(0, t) = \psi(0, t) + \theta F[\phi(t)](0) - \psi(0, t) - \hat{\theta}(t) F[\phi(t)](0) = \tilde{\theta}(t) \int_0^1 \phi(x, t) dx$, and substituting this into (15), we obtain

$$\dot{V}(t) \leq -\sigma^2(t), \quad (16)$$

where we have used the definition of σ stated in (13b). This proves that V is bounded and nonincreasing and hence has a limit, V_∞ , as $t \rightarrow \infty$. Integrating (16) in time from zero to infinity gives

$$\begin{aligned} \int_0^\infty \sigma^2(t) dt &\leq - \int_0^\infty \dot{V}(t) dt \\ &= V(0) - V_\infty \leq V(0) \leq \frac{2}{\gamma} \bar{\theta}^2 < \infty, \end{aligned} \quad (17)$$

and hence $\sigma \in \mathcal{L}_2$. Moreover, using $\hat{e}(0, t) = \tilde{\theta}(t) \int_0^1 \phi(x, t) dx$, we have

$$\begin{aligned} \sigma^2(t) &= \frac{\hat{e}^2(0, t)}{1 + \|\phi(t)\|^2} \\ &\leq \tilde{\theta}^2(t) \frac{(\int_0^1 \phi(x, t) dx)^2}{1 + \|\phi(t)\|^2} \\ &\leq \tilde{\theta}^2(t) \frac{\int_0^1 \phi^2(x, t) dx}{1 + \|\phi(t)\|^2} \\ &\leq \tilde{\theta}^2(t) \frac{\|\phi(t)\|^2}{1 + \|\phi(t)\|^2} \leq \tilde{\theta}^2(t) \leq 4\bar{\theta}^2 < \infty \end{aligned} \quad (18)$$

where we used Cauchy-Schwarz' inequality and hence $\sigma \in \mathcal{L}_\infty$. For $t \in [0, 1)$ or if the projection is active, the adaptive law (10) is zero, and hence boundedness of $\dot{\hat{\theta}}$ follows. For $t \geq 1$, we have for an inactive projection and inserting $\hat{e}(0, t) = \tilde{\theta}(t) \int_0^1 \phi(x, t) dx$, that

$$\begin{aligned} |\dot{\hat{\theta}}(t)| &= \gamma \left| \frac{\hat{e}(0, t) \int_0^1 \phi(x, t) dx}{1 + \|\phi(t)\|^2} \right| \\ &= \gamma \left| \frac{\hat{e}(0, t)}{\sqrt{1 + \|\phi(t)\|^2}} \frac{\int_0^1 \phi(x, t) dx}{\sqrt{1 + \|\phi(t)\|^2}} \right| \\ &\leq \gamma |\sigma(t)| \end{aligned} \quad (19)$$

and since $\sigma \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, it follows that $\dot{\hat{\theta}} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$. $\square$

Adaptive Control Law

As mentioned, the swapping-based design employs certainty equivalence. An adaptive stabilizing control law is hence obtained from the nonadaptive control law (2) by simply substituting the uncertain parameter θ by its

adaptive estimate $\hat{\theta}$ and the unmeasured state u by its adaptive state estimate $\hat{u}$ to obtain

$$U(t) = -\hat{\theta}(t) \int_0^1 e^{\hat{\theta}(t)(1-\xi)} \hat{u}(\xi, t) d\xi. \quad (20)$$

We obtain the following main result for the closed-loop dynamics.

Theorem 1 *Consider system (1a), filters (7), the adaptive law (10), and the adaptive state estimate (11). The control law (20) ensures*

$$\|u\|, \|\psi\|, \|\phi\|, \|u\|_\infty, \|\psi\|_\infty, \|\phi\|_\infty \in \mathcal{L}_2 \cap \mathcal{L}_\infty \quad (21a)$$

$$\|u\|, \|\psi\|, \|\phi\|, \|u\|_\infty, \|\psi\|_\infty, \|\phi\|_\infty \rightarrow 0. \quad (21b)$$

The proof of Theorem 1 involves several steps. First, a backstepping transformation with a time-varying kernel function brings the adaptive state estimate's dynamics into a target system which involves the parameter estimate as well as its time derivatives (adaptive laws). Next, a relatively comprehensive Lyapunov analysis involving heavy use of Cauchy-Schwarz' and Young's inequalities, along with the properties of Lemma 2, follows, which establishes that the Lyapunov function

$$V_1(t) = 4 \int_0^1 (1+x) \eta^2(x, t) dx + \int_0^1 (1+x) \phi^2(x, t) dx \quad (22)$$

satisfies

$$\dot{V}_1(t) \leq -cV_1(t) + l_1(t)V_1(t) + l_2(t) \quad (23)$$

where c is a positive constant, $\eta(x, t) = T[\hat{k}, \hat{u}(t)](x)$ with $\hat{k}$ given by (5) with θ replaced by $\hat{\theta}$ and l_1 and l_2 are non-negative and integrable (i.e., $l_1, l_2 \geq 0, l_1, l_2 \in \mathcal{L}_1$). Finally, established stability and convergence results from the literature on adaptive control are applied.

Summary and Future Directions

It is apparent from the example that the swapping-based method is quite involved even for the simple scalar system. However, the method extends to more complex systems in a natural way and can be modified (see Anfinson and Aamo 2019) to accommodate spatially varying parameters, any number of coupled hyperbolic equations carrying information in either direction, interface with ODE's at the boundary or in the interior, etc. The backstepping transformation induces a complicated relationship between the uncertain parameters of the system in its original coordinates and the uncertainties appearing in the target system, often leading to vast over-parameterization accompanied by poor robustness properties of the closed loop. Robustness in general is a topic for further investigation before the method is ripe for application to practical problems (Lamare et al. 2018). How to deal with nonlinearities is another important direction of future research. While backstepping was invented for systems described by nonlinear ordinary differential equations, it has had little success on the nonlinear arena for PDEs. Some local stability results exist for linear backstepping designs (applied to semi-linear PDEs without uncertainty, Coron et al. 2013), but it is unclear if backstepping can be helpful in providing truly nonlinear control laws in the PDE case. For semi-linear systems without uncertainty, nonlinear state feedback laws and state observers have been designed by an approach not using backstepping (Strecker and Aamo 2017). Interestingly, though, the state feedback law reduces to the backstepping control law in the linear case.

Cross-References

- [Backstepping for PDEs](#)
- [Boundary Control of 1-D Hyperbolic Systems](#)

Recommended Reading

The book (Krstić and Smyshlyaev 2008c) gives an introduction to both nonadaptive and adaptive control of PDEs using the backstepping method. For a more in-depth study of adaptive control of PDEs, the books (Smyshlyaev and Krstić 2010) and (Anfinson and Aamo 2019) provide the state-of-the-art for the parabolic and hyperbolic case, respectively.

Bibliography

- Anfinson H, Aamo OM (2016a) Stabilization of linear 2×2 hyperbolic systems with uncertain coupling coefficients – part I: identifier-based design. In: Australian control conference 2016, Newcastle
- Anfinson H, Aamo OM (2016b) Stabilization of linear 2×2 hyperbolic systems with uncertain coupling coefficients – part II: swapping design. In: Australian control conference 2016, Newcastle
- Anfinson H, Aamo OM (2017a) Adaptive output feedback stabilization of $n + m$ coupled linear hyperbolic PDEs with uncertain boundary conditions. *SIAM J Control Optim* 55(6):3928–3946
- Anfinson H, Aamo OM (2017b) Adaptive output-feedback stabilization of linear 2×2 hyperbolic systems using anti-collocated sensing and control. *Syst Control Lett* 104:86–94
- Anfinson H, Aamo OM (2017c) Adaptive stabilization of 2×2 linear hyperbolic systems with an unknown boundary parameter from collocated sensing and control. *IEEE Trans Autom Control* 62(12):6237–6249
- Anfinson H, Aamo OM (2017d) Adaptive stabilization of $n + 1$ coupled linear hyperbolic systems with uncertain boundary parameters using boundary sensing. *Syst Control Lett* 99:72–84
- Anfinson H, Aamo OM (2018) Adaptive control of linear 2×2 hyperbolic systems. *Automatica* 87:69–82
- Anfinson H, Aamo OM (2019) Adaptive control of hyperbolic PDEs. Springer International Publishing, Cham
- Bernard P, Krstić M (2014) Adaptive output-feedback stabilization of non-local hyperbolic PDEs. *Automatica* 50:2692–2699
- Coron JM, Vazquez R, Krstić M, Bastin G (2013) Local exponential H^2 stabilization of a 2×2 quasilinear hyperbolic system using backstepping. *SIAM J Control Optim* 51(3):2005–2035
- Di Meglio F, Vazquez R, Krstić M (2013) Stabilization of a system of $n + 1$ coupled first-order hyperbolic linear PDEs with a single boundary input. *IEEE Trans Autom Control* 58(12):3097–3111
- Hu L, Di Meglio F, Vazquez R, Krstić M (2016) Control of homodirectional and general heterodirectional linear coupled hyperbolic PDEs. *IEEE Trans Autom Control* 61(11):3301–3314
- Krstić M, Smyshlyaev A (2008a) Adaptive boundary control for unstable parabolic PDEs – part I: Lyapunov design. *IEEE Trans Autom Control* 53(7):1575–1591
- Krstić M, Smyshlyaev A (2008b) Backstepping boundary control for first-order hyperbolic PDEs and application to systems with actuator and sensor delays. *Syst Control Lett* 57(9):750–758
- Krstić M, Smyshlyaev A (2008c) Boundary control of PDEs: a course on backstepping designs. *Soc Ind Appl Math*
- Lamare PO, Auriol J, Aarsnes UJ, Di Meglio F (2018) Robust output regulation of 2×2 hyperbolic systems. In: American control conference 2018, Milwaukee
- Liu W (2003) Boundary feedback stabilization of an unstable heat equation. *SIAM J Control Optim* 42:1033–1043
- Smyshlyaev A, Krstić M (2004) Closed form boundary state feedbacks for a class of 1-D partial integro-differential equations. *IEEE Trans Autom Control* 49:2185–2202
- Smyshlyaev A, Krstić M (2005) Backstepping observers for a class of parabolic PDEs. *Syst Control Lett* 54:613–625
- Smyshlyaev A, Krstić M (2006) Output-feedback adaptive control for parabolic PDEs with spatially varying coefficients. In: 45th IEEE conference on decision & control, San Diego
- Smyshlyaev A, Krstić M (2007a) Adaptive boundary control for unstable parabolic PDEs – part II: estimation-based designs. *Automatica* 43:1543–1556
- Smyshlyaev A, Krstić M (2007b) Adaptive boundary control for unstable parabolic PDEs – part III: output feedback examples with swapping identifiers. *Automatica* 43:1557–1564
- Smyshlyaev A, Krstić M (2010) Adaptive control of parabolic PDEs. Princeton University Press, Princeton
- Strecker T, Aamo OM (2017) Output feedback boundary control of 2×2 semilinear hyperbolic systems. *Automatica* 83:290–302
- Vazquez R, Krstić M, Coron JM (2011) Backstepping boundary stabilization and state estimation of a 2×2 linear hyperbolic system. In: 2011 50th IEEE conference on, decision and control and european control conference (CDC-ECC), pp 4937–4942
- Xu Z, Liu Y (2016) Adaptive boundary stabilization for first-order hyperbolic PDEs with unknown spatially varying parameter. *Int J Robust Nonlinear Control* 26(3):613–628
- Yu H, Vazquez R, Krstić M (2017) Adaptive output feedback for hyperbolic PDE pairs with non-local coupling. In: 2017 American control conference, Seattle

Adaptive Control: Overview

Richard Hume Middleton
School of Electrical Engineering and Computer
Science, The University of Newcastle,
Callaghan, NSW, Australia

Abstract

Adaptive control describes a range of techniques for altering control behavior using measured signals to achieve high control performance under uncertainty. The theory and practice of adaptive control has matured in many areas. This entry gives an overview of adaptive control with pointers to more detailed specific topics.

Keywords

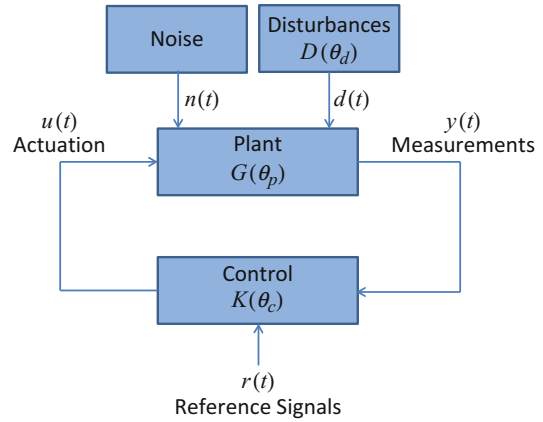
Adaptive control · Estimation

Introduction

What Is Adaptive Control

Feedback control has a long history of using sensing, decision, and actuation elements to achieve an overall goal. The general structure of a control system may be illustrated in Fig. 1. It has long been known that high fidelity control relies on knowledge of the system to be controlled. For example, in most cases, knowledge of the plant gain and/or time constants (represented by θ_p in Fig. 1) is important in feedback control design. In addition, disturbance characteristics (e.g., frequency of a sinusoidal disturbance), θ_d in Fig. 1, are important in feedback compensator design.

Many control design and synthesis techniques are model based, using prior knowledge of both model structure and parameters. In other cases, a fixed controller structure is used, and the controller parameters, θ_c in Fig. 1, are tuned empirically during control system



Adaptive Control: Overview, Fig. 1 General control and adaptive control diagram

commissioning. However, if the plant parameters vary widely with time or have large uncertainties, these approaches may be inadequate for high-performance control.

There are two main ways of approaching high-performance control with unknown plant and disturbance characteristics:

1. Robust control ([► Optimization-Based Robust Control](#)), wherein a controller is designed to perform adequately despite the uncertainties. Variable structure control may have very high levels of robustness in some cases and therefore is a special class of robust nonlinear control.
2. Adaptive control, where the controller learns and adjusts its strategy based on measured data. This frequently takes the form where the controller parameters, θ_c , are time-varying functions that depend on the available data ($y(t)$, $u(t)$, and $r(t)$). Adaptive control has close links to intelligent control (including neural control ([► Optimal Control and the Dynamic Programming Principle](#)), where specific types of learning are considered) and also to stochastic adaptive control ([► Stochastic Adaptive Control](#)).

Robust control is most useful when there are large unmodeled dynamics (i.e., structural uncertainties), relatively high levels of noise, or rapid and unpredictable parameter changes. Conversely, for slow or largely predictable parameter variations, with relatively well-known model structure and limited noise levels, adaptive control may provide a very useful tool for high-performance control (Åström and Wittenmark 2008).

Varieties of Adaptive Control

One practical variant of adaptive control is controller auto-tuning (► [Autotuning](#)). Auto-tuning is particularly useful for PID and similar controllers and involves a specific phase of signal injection, followed by analysis, PID gain computation, and implementation. These techniques are an important aid to commissioning and maintenance of distributed control systems.

There are also large classes of adaptive controllers that are continuously monitoring the plant input-output signals to adjust the strategy. These adjustments are often parametrized by a relatively small number of coefficients, θ_C . These include schemes where the controller parameters are directly adjusted using measureable data (also referred to as “implicit,” since there is no explicit plant model generated). Early examples of this often included model reference adaptive control (► [Model Reference Adaptive Control](#)). Other schemes (Middleton et al. 1988) explicitly estimate a plant model θ_P ; thereafter, performing online control design and, therefore, the adaptation of controller parameters θ_C are indirect. This then led on to a range of other adaptive control techniques applicable to linear systems (► [Adaptive Control of Linear Time-Invariant Systems](#)).

There have been significant questions concerning the sensitivity of some adaptive control algorithms to unmodeled dynamics, time-varying systems, and noise (Ioannou and Kokotovic

1984; Rohrs et al. 1985). This prompted a very active period of research to analyze and redesign adaptive control to provide suitable robustness (► [Robust Adaptive Control](#)) (e.g., Anderson et al. 1986; Ioannou and Sun 2012) and parameter tracking for time-varying systems (e.g., Kresselmeier 1986; Middleton and Goodwin 1988).

Work in this area further spread to nonparametric methods, such as switching, or supervisory adaptive control (► [Switching Adaptive Control](#)) (e.g., Fu and Barmish 1986; Morse et al. 1992). In addition, there has been a great deal of work on the more difficult problem of adaptive control for nonlinear systems (► [Nonlinear Adaptive Control](#)).

A further adaptive control technique is extremum seeking control (► [Extremum Seeking Control](#)). In extremum seeking (or self optimizing) control, the desired reference for the system is unknown, instead we wish to maximize (or minimize) some variable in the system (Ariyur and Krstic 2003). These techniques have quite distinct modes of operation that have proven important in a range of applications.

A final control algorithm that has nonparametric features is iterative learning control (► [Iterative Learning Control](#)) (Amann et al. 1996; Moore 1993). This control scheme considers a system with a highly structured, namely, repetitive finite run, control problem. In this case, by taking a nonparametric approach of utilizing information from previous run(s), in many cases, near-perfect asymptotic tracking can be achieved.

Adaptive control has a rich history (► [History of Adaptive Control](#)) and has been established as an important tool for some classes of control problems.

Cross-References

- [Adaptive Control of Linear Time-Invariant Systems](#)
- [Autotuning](#)

- ▶ Extremum Seeking Control
- ▶ History of Adaptive Control
- ▶ Iterative Learning Control
- ▶ Model Reference Adaptive Control
- ▶ Nonlinear Adaptive Control
- ▶ Optimal Control and the Dynamic Programming Principle
- ▶ Optimization-Based Robust Control
- ▶ Robust Adaptive Control
- ▶ Stochastic Adaptive Control
- ▶ Switching Adaptive Control

Bibliography

- Amann N, Owens DH, Rogers E (1996) Iterative learning control using optimal feedback and feedforward actions. *Int J Control* 65(2):277–293
- Anderson BDO, Bitmead RR, Johnson CR, Kokotovic PV, Kosut RL, Mareels IMY, Praly L, Riedle BD (1986) Stability of adaptive systems: passivity and averaging analysis. MIT, Cambridge
- Ariyur KB, Krstic M (2003) Real-time optimization by extremum-seeking control. Wiley, New Jersey
- Åström KJ, Wittenmark B (2008) Adaptive control. Courier Dover Publications, Mineola
- Fu M, Barmish BR (1986) Adaptive stabilization of linear systems via switching control. *IEEE Trans Autom Control* 31(12):1097–1103
- Ioannou PA, Kokotovic PV (1984) Instability analysis and improvement of robustness of adaptive control. *Automatica* 20(5):583–594
- Ioannou PA, Sun J (2012) Robust adaptive control. Dover Publications, Mineola/New York
- Kreisselmeier G (1986) Adaptive control of a class of slowly time-varying plants. *Syst Control Lett* 8(2):97–103
- Middleton RH, Goodwin GC (1988) Adaptive control of time-varying linear systems. *IEEE Trans Autom Control* 33(2):150–155
- Middleton RH, Goodwin GC, Hill DJ, Mayne DQ (1988) Design issues in adaptive control. *IEEE Trans Autom Control* 33(1):50–58
- Moore KL (1993) Iterative learning control for deterministic systems. *Advances in industrial control series*. Springer, London/New York
- Morse AS, Mayne DQ, Goodwin GC (1992) Applications of hysteresis switching in parameter adaptive control. *IEEE Trans Autom Control* 37(9):1343–1354
- Rohrs C, Valavani L, Athans M, Stein G (1985) Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics. *IEEE Trans Autom Control* 30(9):881–889

Adaptive Cruise Control

Rajesh Rajamani

Department of Mechanical Engineering,
University of Minnesota, Twin Cities,
Minneapolis, MN, USA

Abstract

This chapter discusses advanced cruise control automotive technologies, including adaptive cruise control (ACC) in which spacing control, speed control, and a number of transitional maneuvers must be performed. The ACC system must satisfy difficult performance requirements of vehicle stability and string stability. The technical challenges involved and the control design techniques utilized in ACC system design are presented.

Keywords

Collision avoidance · String stability ·
Traffic stability · Vehicle following

Introduction

Adaptive cruise control (ACC) is an extension of cruise control. An ACC vehicle includes a radar, a lidar, or other sensor that measures the distance to any preceding vehicle in the same lane on the highway. In the absence of preceding vehicles, the speed of the car is controlled to a driver-desired value. In the presence of a preceding vehicle, the controller determines whether the vehicle should switch from speed control to spacing control. In spacing control, the distance to the preceding car is controlled to a desired value.

A different form of advanced cruise control is a *forward collision avoidance* (FCA) system. An FCA system uses a distance sensor to determine if the vehicle is approaching a car ahead too quickly and will automatically apply brakes to minimize the chances of a forward collision. For the 2013 model year, 29 % vehicles have

forward collision warning as an available option and 12 % include autonomous braking for a full FCA system. Examples of models in which an FCA system is standard are the Mercedes Benz G-class and the Volvo S-60, S-80, XC-60, and XC-70.

It should be noted that an FCA system does not involve steady-state vehicle following. An ACC system on the other hand involves control of speed and spacing to desired steady-state values.

ACC systems have been in the market in Japan since 1995, in Europe since 1998, and in the US since 2000. An ACC system provides enhanced driver comfort and convenience by allowing extended operation of the cruise control option even in the presence of other traffic.

Controller Architecture

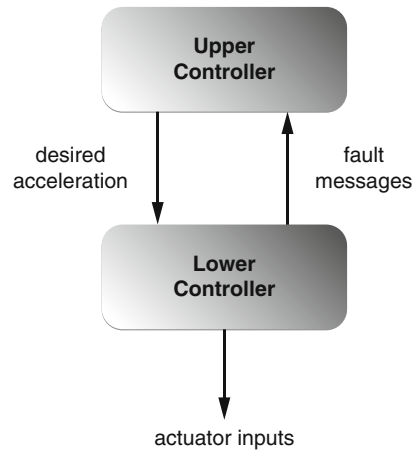
The ACC system has two modes of steady state operation: speed control and vehicle following (i.e., spacing control). Speed control is traditional cruise control and is a well-established technology. A proportional-integral controller based on feedback of vehicle speed (calculated from rotational wheel speeds) is used in cruise control (Rajamani 2012).

Controller design for vehicle following is the primary topic of discussion in the sections titled “Vehicle Following Requirements” and “String Stability Analysis” in this chapter.

Transitional maneuvers and transitional control algorithms are discussed in the section titled “Transitional Maneuvers” in this chapter.

The longitudinal control system architecture for an ACC vehicle is typically designed to be hierarchical, with an upper-level controller and a lower-level controller, as shown in Fig. 1.

The upper-level controller determines the desired acceleration for the vehicle. The lower level controller determines the throttle and/or brake commands required to track the desired acceleration. Vehicle dynamic models, engine maps, and nonlinear control synthesis techniques are used in the design of the lower controller



Adaptive Cruise Control, Fig. 1 Structure of longitudinal control system

(Rajamani 2012). This chapter will focus only on the design of the upper controller, also known as the ACC controller.

As far as the upper-level controller is concerned, the plant model for control design is

$$\ddot{x}_i = u \quad (1)$$

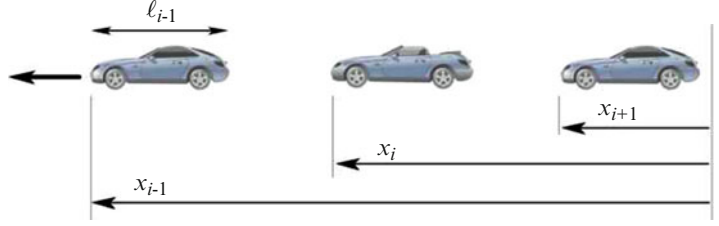
where the subscript i denotes the i th car in a string of consecutive ACC cars. The acceleration of the car is thus assumed to be the control input. However, due to the finite bandwidth associated with the lower level controller, each car is actually expected to track its desired acceleration imperfectly. The objective of the upper level controller design is therefore stated as that of meeting required performance specifications robustly in the presence of a first order lag in the lower-level controller performance:

$$\ddot{x}_i = \frac{1}{\tau s + 1} \ddot{x}_{i,\text{des}} = \frac{1}{\tau s + 1} u_i. \quad (2)$$

Equation (1) is thus assumed to be the nominal plant model while the performance specifications have to be met even if the actual plant model were given by Eq. (2). The lag τ typically has a value between 0.2 and 0.5 s (Rajamani 2012).

Adaptive Cruise Control,

Fig. 2 String of adaptive cruise control vehicles



Vehicle Following Requirements

In the vehicle following mode of operation, the ACC vehicle maintains a desired spacing from the preceding vehicle. The two important performance specifications that the vehicle following control system must satisfy are: individual vehicle stability and string stability.

(a) Individual vehicle stability

Consider a string of vehicles on the highway using a longitudinal control system for vehicle following, as shown in Fig. 2. Let x_i be the location of the i th vehicle measured from an inertial reference. The spacing error for the i th vehicle (the ACC vehicle under consideration) is then defined as

$$\delta_i = x_i - x_{i-1} + L_{\text{des}}. \quad (3)$$

Here, L_{des} is the desired spacing and includes the preceding vehicle length ℓ_{i-1} . L_{des} could be chosen as a function of variables such as the vehicle speed $\dot{x}_i$. The ACC control law is said to provide individual vehicle stability if the spacing error of the ACC vehicle converges to zero when the preceding vehicle is operating at constant speed:

$$\ddot{x}_{i-1} \rightarrow 0 \Rightarrow \delta_i \rightarrow 0. \quad (4)$$

(b) String stability

The spacing error is expected to be non-zero during acceleration or deceleration of the preceding vehicle. It is important then to describe how the spacing error would propagate from vehicle

to vehicle in a string of ACC vehicles during acceleration. The string stability of a string of ACC vehicles refers to a property in which spacing errors are guaranteed not to amplify as they propagate towards the tail of the string (Swaroop and Hedrick 1996).

String Stability Analysis

In this section, mathematical conditions that ensure string stability are provided.

Let δ_i and δ_{i-1} be the spacing errors of consecutive ACC vehicles in a string. Let $\hat{H}(s)$ be the transfer function relating these errors:

$$\hat{H}(s) = \frac{\hat{\delta}_i}{\hat{\delta}_{i-1}}(s). \quad (5)$$

The following two conditions can be used to determine if the system is string stable:

(a) The transfer function $\hat{H}(s)$ should satisfy

$$\|\hat{H}(s)\|_{\infty} \leq 1. \quad (6)$$

(b) The impulse response function $h(t)$ corresponding to $\hat{H}(s)$ should not change sign (Swaroop and Hedrick 1996), i.e.,

$$h(t) > 0 \quad \forall t \geq 0. \quad (7)$$

The reasons for these two requirements to be satisfied are described in Rajamani (2012). Roughly speaking, Eq. (6) ensures that $\|\delta_i\|_2 \leq \|\delta_{i-1}\|_2$, which means that the energy in the spacing error signal decreases as the spacing error propagates towards the tail of the string. Equation (7) ensures that the steady state spacing

errors of the vehicles in the string have the same sign. This is important because a positive spacing error implies that a vehicle is closer than desired while a negative spacing error implies that it is further apart than desired. If the steady state value of δ_i is positive while that of δ_{i-1} is negative, then this might be dangerous due to the vehicle being closer, even though in terms of magnitude δ_i might be smaller than δ_{i-1} .

If conditions (6) and (7) are both satisfied, then $\|\delta_i\|_\infty \leq \|\delta_{i-1}\|_\infty$ (Rajamani 2012).

Constant Inter-vehicle Spacing

The ACC system only utilizes on board sensors like radar and does not depend on inter-vehicle communication from other vehicles. Hence the only variables available as feedback for the upper controller are inter-vehicle spacing, relative velocity and the ACC vehicle's own velocity.

Under the constant spacing policy, the spacing error of the i th vehicle was defined in Eq. (3).

If the acceleration of the vehicle can be instantaneously controlled, then it can be shown that a linear control system of the type

$$\ddot{x}_i = -k_p \delta_i - k_v \dot{\delta}_i \quad (8)$$

results in the following closed-loop transfer function between consecutive spacing errors

$$\hat{H}(s) = \frac{\hat{\delta}_i}{\hat{\delta}_{i-1}}(s) = \frac{k_p + k_v s}{s^2 + k_v s + k_p}. \quad (9)$$

Equation (9) describes the propagation of spacing errors along the vehicle string.

All positive values of k_p and k_v guarantee individual vehicle stability. However, it can be shown that there are no positive values of k_p and k_v for which the magnitude of $G(s)$ can be guaranteed to be less than unity at all frequencies. The details of this proof are available in Rajamani (2012).

Thus, the constant spacing policy will always be string unstable.

Constant Time-Gap Spacing

Since the constant spacing policy is unsuitable for autonomous control, a better spacing policy that can ensure both individual vehicle stability and string stability must be used. The constant time-gap (CTG) spacing policy is such a spacing policy. In the CTG spacing policy, the desired inter-vehicle spacing is not constant but varies with velocity. The spacing error is defined as

$$\delta_i = x_i - x_{i-1} + L_{\text{des}} + h\dot{x}_i. \quad (10)$$

The parameter h is referred to as the time-gap.

The following controller based on the CTG spacing policy can be used to regulate the spacing error at zero (Swaroop et al. 1994):

$$\ddot{x}_{i_des} = -\frac{1}{h}(\dot{x}_i - \dot{x}_{i-1} + \lambda\delta_i) \quad (11)$$

With this control law, it can be shown that the spacing errors of successive vehicles δ_i and δ_{i-1} are independent of each other:

$$\dot{\delta}_i = -\lambda\delta_i \quad (12)$$

Thus, δ_i is independent of δ_{i-1} and is expected to converge to zero as long as $\lambda > 0$. However, this result is only true if any desired acceleration can be instantaneously obtained by the vehicle i.e., if $\tau = 0$.

In the presence of the lower controller and actuator dynamics given by Eq. (2), it can be shown that the dynamic relation between δ_i and δ_{i-1} in the transfer function domain is

$$\hat{H}(s) = \frac{s + \lambda}{h\tau s^3 + hs^2 + (1 + \lambda h)s + \lambda} \quad (13)$$

The string stability of this system can be analyzed by checking if the magnitude of the above transfer function is always less than or equal to 1. It can be shown that this is the case at all frequencies if and only if (Rajamani 2012)

$$h \geq 2\tau. \quad (14)$$

Further, if Eq. (14) is satisfied, then it is also guaranteed that one can find a value of λ such that Eq. (7) is satisfied. Thus the condition (14) is necessary (Swaroop and Hedrick 1996) for string stability.

Since the typical value of τ is of the order of 0.5 s, Eq. (14) implies that ACC vehicles must maintain at least a 1-s time gap between vehicles for string stability.

Transitional Maneuvers

While under speed control, an ACC vehicle might suddenly encounter a new vehicle in its lane (either due to a lane change or due to a slower moving preceding vehicle). The ACC vehicle must then decide whether to continue to operate under the speed control mode or transition to the vehicle following mode or initiate hard braking. If a transition to vehicle following is required, a transitional trajectory that will bring the ACC vehicle to its steady state following distance needs to be designed. Similarly, a decision on the mode of operation and design of a transitional trajectory are required when an ACC vehicle loses its target.

The regular CTG control law cannot directly be used to follow a newly encountered vehicle, see Rajamani (2012) for illustrative examples.

When a new target vehicle is encountered by the ACC vehicle, a “range – range rate” diagram can be used (Fancher and Bareket 1994) to decide if

- The vehicle should use speed control.
- The vehicle should use spacing control (with a defined transition trajectory in which desired spacing varies slowly with time)
- The vehicle should brake as hard as possible in order to avoid a crash.

The maximum allowable values for acceleration and deceleration need to be taken into account in making these decisions.

For the range – range rate ($R - \dot{R}$) diagram, define range R and range rate $\dot{R}$ as

$$R = x_{i-1} - x_i \quad (15)$$

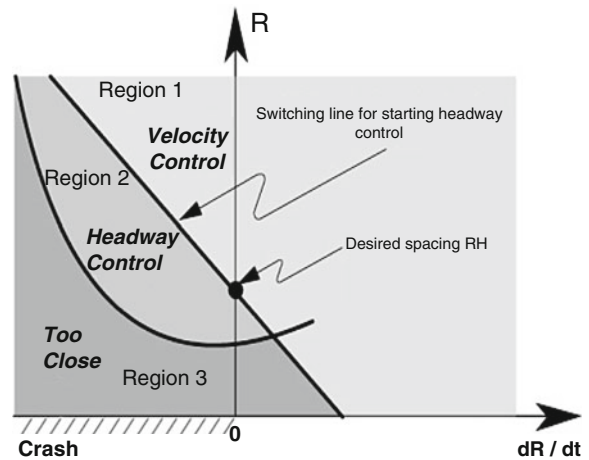
$$\dot{R} = \dot{x}_{i-1} - \dot{x}_i = V_{i-1} - V_i \quad (16)$$

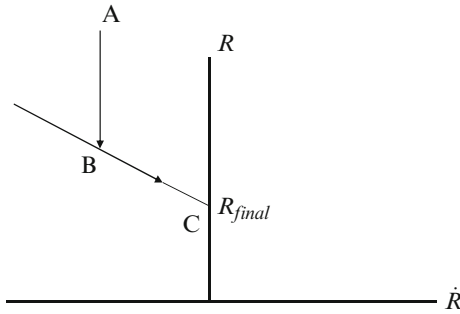
where x_{i-1} , x_i , V_{i-1} , and V_i are inertial positions and velocities of the preceding vehicle and the ACC vehicle respectively.

A typical $R - \dot{R}$ diagram is shown in Fig. 3 (Fancher and Bareket 1994). Depending on the measured real-time values of R and $\dot{R}$, and the $R - \dot{R}$ diagram in Fig. 3, the ACC system determines the mode of longitudinal control. For instance, in region 1, the vehicle continues

Adaptive Cruise Control,

Fig. 3 Range vs. range-rate diagram





Adaptive Cruise Control, Fig. 4 Switching line for spacing control

to operate under speed control. In region 2, the vehicle operates under spacing control. In region 3, the vehicle decelerates at the maximum allowed deceleration so as to try and avoid a crash.

The switching line from speed to spacing control is given by

$$R = -T\dot{R} + R_{\text{final}} \quad (17)$$

where T is the slope of the switching line. When a slower vehicle is encountered at a distance larger than the desired final distance R_{final} , the switching line shown in Fig. 4 can be used to determine when and whether the vehicle should switch to spacing control. If the distance R is greater than that given by the line, speed control should be used.

The overall strategy (shown by trajectory ABC) is to first reduce gap at constant $\dot{R}$ and then follow the desired spacing given by the switching line of Eq. (17).

The control law during spacing control on this transitional trajectory is as follows. Depending on the value of $\dot{R}$, determine R from Eq. (17). Then use R as the desired inter-vehicle spacing in the PD control law

$$\ddot{x}_{\text{des}} = -k_p(x_i - R) - k_d(\dot{x}_i - \dot{R}) \quad (18)$$

The trajectory of the ACC vehicle during constant deceleration is a parabola on the $R - \dot{R}$ diagram (Rajamani 2012).

The switching line should be such that travel along the line is comfortable and does not constitute high deceleration. The deceleration during coasting (zero throttle and zero braking) can be used to determine the slope of the switching line (Rajamani 2012).

Note that string stability is not a concern during transitional maneuvers (Rajamani 2012).

Traffic Stability

In addition to individual vehicle stability and string stability, another type of stability analysis that has received significant interest in ACC literature is traffic flow stability. Traffic flow stability refers to the stable evolution of traffic velocity and traffic density on a highway section, for given inflow and outflow conditions. One well-known result in this regard in literature is that traffic flow is defined to be stable if $\frac{\partial q}{\partial \rho}$ is positive, i.e., as the density ρ of traffic increases, traffic flow rate q must increase (Swaroop and Rajagopal 1999). If this condition is not satisfied, the highway section would be unable to accommodate any constant inflow of vehicles from an oncoming ramp. The steady state traffic flow on the highway section would come to a stop, if the ramp inflow did not stop (Swaroop and Rajagopal 1999).

It has been shown that the constant time-gap spacing policy used in ACC systems has a negative $q - \rho$ slope and thus does not lead to traffic flow stability (Swaroop and Rajagopal 1999). It has also been shown that it is possible to design other spacing policies (in which the desired spacing between vehicles is a nonlinear function of speed, instead of being proportional to speed) that can provide stable traffic flow (Santhanakrishnan and Rajamani 2003).

The importance of traffic flow stability has not been fully understood by the research community. Traffic flow stability is likely to become important when the number of ACC vehicles

on the highway increase and their penetration percentage into vehicles on the road becomes significant.

Recent Automotive Market Developments

The latest versions of ACC systems on the market have been enhanced with collision warning, integrated brake support, and stop-and-go operation functionality.

The collision warning feature uses the same radar as the ACC system to detect moving vehicles ahead and determine whether driver intervention is required. In this case, visual and audio warnings are provided to alert the driver and brakes are pre-charged to allow quick deceleration. On Ford's ACC-equipped vehicles, brakes are also automatically applied when the driver lifts the foot off from the accelerator pedal in a detected collision warning scenario.

When enabled with stop-and-go functionality, the ACC system can also operate at low vehicle speeds in heavy traffic. The vehicle can be automatically brought to a complete stop when needed and restarted automatically. Stop-and-go is an expensive option and requires the use of multiple radar sensors on each car. For instance, the BMW ACC system uses two short range and one long range radar sensor for stop-and-go operation.

The 2013 versions of ACC on the Cadillac ATS and on the Mercedes Distronic systems are also being integrated with camera based lateral lane position measurement systems. On the Mercedes Distronic systems, a camera steering assist system provides automatic steering, while on the Cadillac ATS, a camera based system provides lane departure warnings.

Future Directions

Current ACC systems use only on-board sensors and do not use wireless communication with

other vehicles. There is a likelihood of evolution of current systems into co-operative adaptive cruise control (CACC) systems which utilize wireless communication with other vehicles and highway infrastructure. This evolution could be facilitated by the dedicated short-range communications (DSRC) capability being developed by government agencies in the US, Europe and Japan. In the US, DSRC is being developed with a primary goal of enabling communication between vehicles and with infrastructure to reduce collisions and support other safety applications. In CACC, wireless communication could provide acceleration signals from several preceding downstream vehicles. These signals could be used in better spacing policies and control algorithms to improve safety, ensure string stability, and improve traffic flow.

Cross-References

- [Lane Keeping Systems](#)
- [Vehicle Dynamics Control](#)
- [Vehicular Chains](#)

Bibliography

- Fancher P, Bareket Z (1994) Evaluating headway control using range versus range-rate relationships. *Veh Syst Dyn* 23(8):575–596
- Rajamani R (2012) *Vehicle dynamics and control*, 2nd edn. Springer, New York. ISBN:978-1461414322
- Santhanakrishnan K, Rajamani R (2003) On spacing policies for highway vehicle automation. *IEEE Trans Intell Transp Syst* 4(4):198–204
- Swaroop D, Hedrick JK (1996) String stability of interconnected dynamic systems. *IEEE Trans Autom Control* 41(3):349–357
- Swaroop D, Rajagopal KR (1999) Intelligent cruise control systems and traffic flow stability. *Transp Res C Emerg Technol* 7(6):329–352
- Swaroop D, Hedrick JK, Chien CC, Ioannou P (1994) A comparison of spacing and headway control laws for automatically controlled vehicles. *Veh Syst Dyn* 23(8):597–625

Adaptive Horizon Model Predictive Control and Al'brekht's Method

Arthur J. Krener

Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA, USA

Abstract

A standard way of finding a feedback law that stabilizes a control system to an operating point is to recast the problem as an infinite horizon optimal control problem. If the optimal cost and the optimal feedback can be found on a large domain around the operating point, then a Lyapunov argument can be used to verify the asymptotic stability of the closed loop dynamics. The problem with this approach is that it is usually very difficult to find the optimal cost and the optimal feedback on a large domain for nonlinear problems with or even without constraints, hence the increasing interest in Model Predictive Control (MPC). In standard MPC a finite horizon optimal control problem is solved in real time, but just at the current state, the first control action is implemented, the system evolves one time step, and the process is repeated. A terminal cost and terminal feedback found by Al'brekht's method and defined in a neighborhood of the operating point can be used to shorten the horizon and thereby make the nonlinear programs easier to solve because they have less decision variables. Adaptive Horizon Model Predictive Control (AHMPC) is a scheme for varying the horizon length of Model Predictive Control as needed. Its goal is to achieve stabilization with horizons as small as possible so that MPC methods can be used on faster and/or more complicated dynamic processes.

Keywords

Adaptive horizon · Model predictive control · Al'brekht's method · Optimal stabilization

Introduction

Model Predictive Control (MPC) is a way to steer a discrete time control system to a desired operating point. We will present an extension of MPC that we call Adaptive Horizon Model Predictive Control (AHMPC) which adjusts the length of the horizon in MPC while nearly verifying in real time that stabilization is occurring for a nonlinear system.

We are not the first to consider adaptively changing the horizon length in MPC; see Michalska and Mayne (1993), Polak and Yang (1993a, b, c). In these papers the horizon is changed so that a terminal constraint is satisfied by the predicted state at the end of horizon. In Giselsson (2010) the horizon length is adaptively changed to ensure that the infinite horizon cost of using the finite horizon MPC scheme is not much more than the cost of the corresponding infinite horizon optimal control problem.

Adaptive horizon tracking is discussed in Page et al. (2006) and Droge and Egerstedt (2011). In Kim and Sugie (2008) an adaptive parameter estimation algorithm suitable for MPC was proposed, which uses the available input and output signals to estimate the unknown system parameters. In Gruene et al. (2010) a detailed analysis of the impact of the optimization horizon and the time varying control horizon on stability and performance of the closed loop is given.

Review of Model Predictive Control

We briefly describe MPC following the definitive treatise of Rawlings and Mayne (2009). We largely follow their notation.

We are given a controlled, nonlinear dynamics in discrete time

$$x^+ = f(x, u) \quad (1)$$

where the state $x \in \mathbb{R}^{n \times 1}$, the control $u \in \mathbb{R}^{m \times 1}$, and $x^+(k) = x(k+1)$. Typically this a time discretization of a controlled, nonlinear dynamics in continuous time. The goal is to find a feedback law $u(k) = \kappa(x(k))$ that drives the

state of the system to some desired operating point. A pair (x^e, u^e) is an operating point if $f(x^e, u^e) = x^e$. We conveniently assume that, after state and control coordinate translations, the operating point of interest is $(x^e, u^e) = (0, 0)$.

The controlled dynamics may be subject to constraints such as

$$x \in \mathbb{X} \subset \mathbb{R}^{n \times 1} \quad (2)$$

$$u \in \mathbb{U} \subset \mathbb{R}^{m \times 1} \quad (3)$$

and possibly constraints involving both the state and control

$$y = h(x, u) \in \mathbb{Y} \subset \mathbb{R}^{p \times 1} \quad (4)$$

A control u is said to be feasible at $x \in \mathbb{X}$ if

$$u \in \mathbb{U}, f(x, u) \in \mathbb{X}, h(x, u) \in \mathbb{Y} \quad (5)$$

Of course the stabilizing feedback $\kappa(x)$ that we seek needs to be feasible, that is, for every $x \in \mathbb{X}$,

$$\kappa(x) \in \mathbb{U}, f(x, \kappa(x)) \in \mathbb{X}, h(x, \kappa(x)) \in \mathbb{Y}$$

An ideal way to find a stabilizing feedback is to choose a Lagrangian $l(x, u)$ (aka running cost), that is, nonnegative definite in x, u and positive definite in u , and then to solve the infinite horizon optimal control problem of minimizing the quantity

$$\sum_{k=0}^{\infty} l(x(k), u(k))$$

over all choices of infinite control sequences $\mathbf{u} = (u(0), u(1), \dots)$ subject to the dynamics (1), the constraints (2, 3, 4), and the initial condition $x(0) = x^0$. Assuming the minimum exists for each $x^0 \in \mathbb{X}$, we define the optimal cost function

$$V(x^0) = \min_u \sum_{k=0}^{\infty} l(x(k), u(k)) \quad (6)$$

Let $\mathbf{u}^* = (u^*(0), u^*(1), \dots)$ be a minimizing control sequence with corresponding state sequence $\mathbf{x}^* = (x^*(0) = x^0, x^*(1), \dots)$. Minimizing control and state sequences need

not be unique, but we shall generally ignore this problem because we are using optimization as a path to stabilization. The key question is whether the possibly nonunique solution is stabilizing to the desired operating point. As we shall see, AHMPC nearly verifies stabilization in real time.

If a pair $V(x) \in \mathbb{R}, \kappa(x) \in \mathbb{R}^{m \times 1}$ of functions satisfy the infinite horizon Bellman Dynamic Program Equations (BDP)

$$\begin{aligned} V(x) &= \min_u \{V(f(x, u)) + l(x, u)\} \\ \kappa(x) &= \operatorname{argmin}_u \{V(f(x, u)) + l(x, u)\} \end{aligned} \quad (7)$$

and the constraints

$$\kappa(x) \in \mathbb{U}, f(x, \kappa(x)) \in \mathbb{X}, h(x, \kappa(x)) \in \mathbb{Y} \quad (8)$$

for all $x \in \mathbb{X}$, then it is not hard to show that $V(x)$ is the optimal cost and $\kappa(x)$ is an optimal feedback on $\mathbb{X}$. Under suitable conditions a Lyapunov argument can be used to show that the feedback $\kappa(x)$ is stabilizing.

The difficulty with this approach is that it is generally impossible to solve the BDP equations on a large domain $\mathbb{X}$ if the state dimension n is greater than 2 or 3. So both theorists and practitioners have turned to Model Predictive Control (MPC).

In MPC one chooses a Lagrangian $l(x, u)$, a horizon length N , a terminal domain $\mathbb{X}_f \subset \mathbb{X}$ containing $x = 0$, and a terminal cost $V_f(x)$ defined and positive definite on $\mathbb{X}_f$. Then one considers the problem of minimizing

$$\sum_{k=0}^{N-1} l(x(k), u(k)) + V_f(x(N))$$

by choice of feasible

$$\mathbf{u}_N = (u_N(0), u_N(1), \dots, u_N(N-1))$$

subject to the dynamics (1), the constraints (8), the final condition $x(N) \in \mathbb{X}_f$, and the initial condition $x(0) = x^0$. Assuming this problem is solvable, let $V_N(x^0)$ denote the optimal cost

$$V_N(x^0) = \min_{u_N} \sum_{k=0}^{N-1} l(x(k), u(k)) + V_f(x(N)) \quad (9)$$

and let

$$\begin{aligned} u_N^* &= (u_N^*(0), u_N^*(1), \dots, u_N^*(N-1)) \\ x_N^* &= (x_N^*(0) = x^0, x_N^*(1), \dots, x_N^*(N)) \end{aligned}$$

denote optimal control and state sequences starting from x^0 when the horizon length is N . We then define the MPC feedback law $\kappa_N(x^0) = u_N^*(0)$.

The terminal set $\mathbb{X}_f$ is controlled invariant (aka viable) if for each $x \in \mathbb{X}_f$ there exists a $u \in \mathbb{U}$ such that $f(x, u) \in \mathbb{X}_f$ and $h(x, u) \in \mathbb{Y}$ are satisfied. If $V_f(x)$ is a control Lyapunov function on $\mathbb{X}_f$, then, under suitable conditions, a Lyapunov argument can be used to show that the feedback $\kappa_N(x)$ is stabilizing on $\mathbb{X}_f$. See Rawlings and Mayne (2009) for more details.

AHMPC requires a little more, the existence of terminal feedback $u = \kappa_f(x)$ defined on the terminal set $\mathbb{X}_f$ that leaves it positively invariant, if $x \in \mathbb{X}_f$ then $f(x, \kappa_f(x)) \in \mathbb{X}_f$, and which makes $V_f(x)$ a strict Lyapunov function on $\mathbb{X}_f$ for the closed loop dynamics, if $x \in \mathbb{X}_f$ and $x \neq 0$ then

$$V_f(x) > V_f(f(x, \kappa_f(x))) \geq 0$$

If $x \in \mathbb{X}_f$ then AHMPC does not need to solve (9) to get u , it just takes $u = \kappa_f(x)$. A similar scheme has been called dual mode control in Michalska and Mayne (1993).

The advantage of solving the finite horizon optimal control problem (9) over solving the infinite horizon problem (6) is that it may be possible to solve the former online as the process evolves. If it is known that the terminal set $\mathbb{X}_f$ can be reached from the current state x in N or fewer steps, then the finite horizon N optimal control problem is a feasible nonlinear program with finite dimensional decision variable $u_N \in \mathbb{R}^{m \times N}$. If the time step is long enough, if f, h, l are reasonably simple, and if N is small enough, then this nonlinear program possibly can be solved in

a fraction of one time step for u_N^* . Then the first element of this sequence $u_N^*(0)$ is used as the control at the current time. The system evolves one time step, and the process is repeated at the next time. Conceptually MPC computes a feedback law $\kappa_N(x) = u_N^*(0)$ but only at values of x when and where it is needed.

Some authors like Grimm et al. (2005) and Gruene (2012) do away with the terminal cost $V_f(x)$, but there is a theoretical reason and a practical reason to use one. The theoretical reason is that a control Lyapunov terminal cost facilitates a proof of asymptotic stability via a simple Lyapunov argument; see Rawlings and Mayne (2009). But this is not a binding reason because under suitable assumptions, asymptotic stability can be shown even when there is no terminal cost provided the horizon is sufficiently long. The practical reason is more important; when there is a terminal cost one can usually use a shorter horizon N . A shorter horizon reduces the dimension mN of the decision variables in the nonlinear programs that need to be solved online. Therefore MPC with a suitable terminal cost can be used for faster and more complicated systems.

An ideal terminal cost $V_f(x)$ is $V(x)$ of the corresponding infinite horizon optimal control provided that the latter can be accurately computed off-line on a reasonably large terminal set $\mathbb{X}_f$. For then the infinite horizon cost (6) and (9) will be the same. One should not make too much of this fact as stabilization is our goal; the optimal control problems are just a means to accomplish this. This in contrast to Economic MPC where the cost and the associated Lagrangian are chosen to model real-world costs.

Adaptive Horizon Model Predictive Control

For AHMPC we assume that we have the following;

- A discrete time dynamics $f(x, u)$ with operating point $x = 0, u = 0$.
- A Lagrangian $l(x, u)$, nonnegative definite in (x, u) and positive definite in u .

- State constraints $x \in \mathbb{X}$ where $\mathbb{X}$ is a neighborhood of $x = 0$.
- Control constraints $u \in \mathbb{U}$ where $\mathbb{U}$ is a neighborhood of $u = 0$.
- Mixed constraints $h(x, u) \in \mathbb{Y}$ which are not active at the operating point $x = 0, u = 0$.
- The dynamics is recursively feasible on $\mathbb{X}$, that is, for every $x \in \mathbb{X}$ there is a $u \in \mathbb{U}$ satisfying $h(x, u) \in \mathbb{Y}$ and $f(x, u) \in \mathbb{X}$.
- A terminal cost $V_f(x)$ defined and nonnegative definite on some neighborhood $\mathbb{X}_f$ of the operating point $x = 0, u = 0$. The neighborhood $\mathbb{X}_f$ need not be known explicitly.
- A terminal feedback $u = \kappa_f(x)$ defined on $\mathbb{X}_f$ such that the terminal cost is a valid Lyapunov function on $\mathbb{X}_f$ for the closed loop dynamics using the terminal feedback $u = \kappa_f(x)$.

One way of obtaining a terminal pair $V_f(x), \kappa_f(x)$ is to approximately solve the infinite horizon dynamic program equations (BDP) on some neighborhood of the origin. For example, if the linear part of the dynamics and the quadratic part of the Lagrangian constitute a linear quadratic regulator (LQR) problem satisfying the standard conditions, then one can let $V_f(x)$ be the quadratic optimal cost and $\kappa_f(x)$ be the linear optimal feedback of this LQR problem. Of course the problem with such terminal pairs $V_f(x), \kappa_f(x)$ is that generally there is no way to estimate the terminal set $\mathbb{X}_f$ on which the feasibility and Lyapunov conditions are satisfied. It is reasonable to expect that they are satisfied on some terminal set but the extent of this terminal set is difficult to estimate.

In the next section we show how higher degree Taylor polynomials for the optimal cost and optimal feedback can be computed by the discrete time extension (Aguilar and Krener 2012) of Al'brekht's method (Al'brecht 1961) because this can lead to a larger terminal set $\mathbb{X}_f$ on which the feasibility and the Lyapunov conditions are satisfied. It would be very difficult to determine what this terminal set is, but fortunately in AHMPC we do not need to do this.

AHMPC mitigates this last difficulty just as MPC mitigates the problem of solving the infinite horizon Bellman Dynamic Programming

equations BDP (7). MPC does not try to compute the optimal cost and optimal feedback everywhere; instead it computes them just when and where they are needed. AHMPC does not try to compute the set $\mathbb{X}_f$ on which $\kappa_f(x)$ is feasible and stabilizing; it just tries to determine if the end state $x_N^*(N)$ of the currently computed optimal trajectory is in a terminal set $\mathbb{X}_f$ where the feasibility and Lyapunov conditions are satisfied.

Suppose the current state is x and we have solved the horizon N optimal control problem for $u_N^* = (u_N^*(0), \dots, u_N^*(N-1))$, $x_N^* = (x_N^*(0) = x, \dots, x_N^*(N))$. The terminal feedback $u = \kappa_f(x)$ is used to compute M additional steps of this state trajectory

$$x_N^*(k+1) = f(x_N^*(k), \kappa_f(x_N^*(k))) \quad (10)$$

for $k = N, \dots, N+M-1$.

Then one checks that the feasibility and Lyapunov conditions hold for the extended part of the state sequence,

$$\kappa_f(x_N^*(k)) \in \mathbb{U} \quad (11)$$

$$f(x_N^*(k), \kappa_f(x_N^*(k))) \in \mathbb{X} \quad (12)$$

$$h(x_N^*(k), \kappa_f(x_N^*(k))) \in \mathbb{Y} \quad (13)$$

$$V_f(x_N^*(k)) \geq \alpha(|x_N^*(k)|) \quad (14)$$

$$V_f(x_N^*(k)) - V_f(x_N^*(k+1)) \geq \alpha(|x_N^*(k)|) \quad (15)$$

for $k = N, \dots, N+M-1$ and some Class K function $\alpha(s)$. For more on Class K functions, we refer the reader to Khalil (1996).

If (11–15) hold for all for $k = N, \dots, N+M-1$, then we presume that $x_N^*(N) \in \mathbb{X}_f$, a set where $\kappa_f(x)$ is stabilizing and we use the control $u_N^*(0)$ to move one time step forward to $x^+ = f(x, u_N^*(0))$. At this next state x^+ , we solve the horizon $N-1$ optimal control problem and check that the extension of the new optimal trajectory satisfies (11–15).

If (11–15) does not hold for all for $k = N, \dots, N+M-1$, then we presume that $x_N^*(N) \notin \mathbb{X}_f$. We extend current horizon N

to $N + L$ where $L \geq 1$, and if time permits, we solve the horizon $N + L$ optimal control problem at the current state x and then check the feasibility and Lyapunov conditions again. We keep increasing N by L until these conditions are satisfied on the extension of the trajectory. If we run out of time before they are satisfied, then we use the last computed $u_N^*(0)$ and move one time step forward to $x^+ = f(x, u_N^*(0))$. At x^+ we solve the horizon $N + L$ optimal control problem.

How does one choose the extended horizon M and the class K function $\alpha(\cdot)$? If the extended part of the state sequence is actually in the region where the terminal cost $V_f(x)$ and the terminal feedback $\kappa_f(x)$ well approximate the solution to infinite horizon optimal control problem, then the dynamic programming equations (7) should approximately hold. In other words

$$\begin{aligned} V_f(x_N^*(k)) - V_f(x_N^*(k+1)) \\ \approx l(x_N^*(k), \kappa_f(x_N^*(k))) \geq 0 \end{aligned}$$

If this does not hold throughout the extended trajectory, we should increase the horizon N . We can also increase the extended horizon M , but this may not be necessary. If the Lyapunov and feasibility conditions are going to fail somewhere on the extension, it is most likely this will happen at the beginning of the extension. Also we should choose $\alpha(\cdot)$ so that $\alpha(|x|) < |l(x, \kappa_f(x))|/2$.

The nonlinear programming problems generated by employing MPC on a nonlinear system are generally nonconvex so the solver might return local rather than global minimizers. In which case there is no guarantee that an MPC approach is actually stabilizing. AHMPC mitigates this difficulty by nearly checking that stabilization is occurring. If (11–15) don't hold even after the horizon N has been increased substantially, then this is a strong indication that the solver is returning locally rather than globally minimizing solutions, and these local solutions are not stabilizing. To change this behavior one needs to start the solver at a substantially different initial guess. Just how one does this is an open research question. It is essentially the same ques-

tion as which initial guess should one pass to the solver at the first step of MPC.

The actual computation of the M additional steps (10) can be done very quickly because the closed loop dynamics function $f(x, \kappa_f(x))$ can be computed and compiled beforehand. Similarly the feasibility and Lyapunov conditions (11–15) can be computed and compiled beforehand. The number M of additional time steps is a design parameter. One choice is to take M a positive integer; another choice is a positive integer plus a fraction of the current N .

Choosing a Terminal Cost and a Terminal Feedback

A standard way of obtaining a terminal cost $V_f(x)$ and a terminal feedback $\kappa_f(x)$ is to solve the linear quadratic regulator (LQR) using the quadratic part of the Lagrangian and the linear part of dynamics around the operating point $(x^e, u^e) = (0, 0)$. Suppose

$$\begin{aligned} l(x, u) &= \frac{1}{2} (x' Q x + 2x' S u + u' R u) + O(x, u)^3 \\ f(x, u) &= Fx + Gu + O(x, u)^2 \end{aligned}$$

Then the LQR problem is to find P, K such that

$$\begin{aligned} P &= F' P F - (F' P G + S) \\ &\quad (R + G' P G)^{-1} (G' P F + S') + Q \end{aligned} \quad (16)$$

$$K = -(R + G' P G)^{-1} (G' P F + S') \quad (17)$$

Under mild assumptions, the stabilizability of F, G , the detectability of $Q^{1/2}, F$, the nonnegative definiteness of $[Q, S; S', R]$, and the positive definiteness of R , there exist a unique nonnegative definite P satisfying the first equation (16) which is called the discrete time algebraic Riccati equation (DARE). Then K given by (17) puts all the poles of the closed loop linear dynamics

$$x^+ (F + GK) x$$

inside of the open unit disk. See Antsaklis and Michel (1997) for details. But $l(x, u)$ is a design parameter so we choose $[Q, S; S', R]$ to be positive definite and then P will be positive definite.

If we define the terminal cost to be $V_f(x) = \frac{1}{2}x'Px$, then we know that it is positive definite for all $x \in \mathbb{R}^{n \times 1}$. If we define the terminal feedback to be $\kappa_f(x) = Kx$, then we know by a Lyapunov argument that the nonlinear closed loop dynamics

$$x^+ = f(x, \kappa(x))$$

is locally asymptotically stable around $x^e = 0$. The problem is that we don't know what is the neighborhood $\mathbb{X}_f$ of asymptotic stability and computing it off-line can be very difficult in state dimensions higher than two or three.

There are other possible choices for the terminal cost and terminal feedback. Al'brecht (1961) showed how the Taylor polynomials of the optimal cost and the optimal feedback could be computed for some smooth, infinite horizon optimal control problems in continuous time. Aguilar and Krener (2012) extended this to some smooth, infinite horizon optimal control problems in discrete time. The discrete time Taylor polynomials of the optimal cost and the optimal feedback may be used as the terminal cost and terminal feedback in an AHMPC scheme, so we briefly review (Aguilar and Krener 2012).

Since we assumed that $f(x, u)$ and $l(x, u)$ are smooth and the constraints are not active at the origin, we can simplify the BDP equations. The simplified Bellman Dynamic Programming equations (sBDP) are obtained by setting the derivative with respect to u of the quantity to be minimized in (7) to zero. The result is

$$V(x) = V(f(x, \kappa(x))) + l(x, \kappa(x)) \quad (18)$$

$$0 = \frac{\partial V}{\partial x}(f(x, u)) \frac{\partial f}{\partial u}(x, \kappa(x)) + \frac{\partial l}{\partial u}(x, \kappa(x)) \quad (19)$$

If the quantity to be minimized is strictly convex in u , then the BDP equations and sBDP equations

are equivalent. But if not then BDP implies sBDP but not vice versa.

Suppose the discrete time dynamics and Lagrangian have Taylor polynomials around the operating point $x = 0, u = 0$ of the form

$$\begin{aligned} f(x, u) &= Fx + Gu + f^{[2]}(x, u) \\ &\quad + \dots + f^{[d]}(x, u) + O(x, u)^{d+1} \\ l(x, u) &= \frac{1}{2}(x'Qx + 2x'Su + u'Ru) + l^{[3]}(x, u) \\ &\quad + \dots + l^{[d+1]}(x, u) + O(x, u)^{d+2} \end{aligned}$$

for some integer $d \geq 1$ where $^{[j]}$ indicates the homogeneous polynomial terms of degree j .

Also suppose the infinite horizon optimal cost and optimal feedback have similar Taylor polynomials

$$\begin{aligned} V(x) &= \frac{1}{2}x'Px + V^{[3]}(x) + \dots \\ &\quad + V^{[d+1]}(x) + O(x, u)^{d+2} \\ \kappa(x) &= Kx + \kappa^{[2]}(x, u) + \dots \\ &\quad + \kappa^{[d]}(x, u) + O(x, u)^{d+1} \end{aligned}$$

We plug these polynomials into sBDP and collect terms of lowest degree. The lowest degree in (18) is two while in (19) it is one. The result is the discrete time Riccati equations (16, 17).

At the next degrees we obtain the equations

$$\begin{aligned} V^{[3]}(x) &= V^{[3]}((F + GK)x) \\ &= ((F + GK)x)' P f^{[2]}(x, Kx) \\ &\quad + l^{[3]}(x, Kx) \end{aligned} \quad (20)$$

$$\begin{aligned} &\left(\kappa^{[2]}(x) \right)' (R + G'PG) \\ &= -\frac{\partial V^{[3]}}{\partial x}((F + GK)x)G \\ &\quad - ((F + GK)x)' P \frac{\partial f^{[2]}}{\partial u}(x, Kx) \\ &\quad - \frac{\partial l^{[3]}}{\partial u}(x, Kx) \end{aligned} \quad (21)$$

Notice these are linear equations in the unknowns $V^{[3]}(x)$ and $\kappa^{[2]}(x)$ and the right sides of these equations involve only known quantities. Moreover $\kappa^{[2]}(x)$ does not appear in the first equation. The mapping

$$V^{[3]}(x) \mapsto V^{[3]}(x) - V^{[3]}((F + GK)x) \quad (22)$$

is a linear operator on the space of polynomials of degree three in x . Its eigenvalues are of the form of 1 minus the products of three eigenvalues of $F + GK$. Since the eigenvalues of $F + GK$ are all inside the open unit disk, zero is not an eigenvalue of the operator (22), so it is invertible. Having solved (20) for $V^{[3]}(x)$, we can readily solve (21) for $\kappa^{[2]}(x)$ since we have assumed R is positive definite

The higher degree equations are similar, at degrees $d + 1, d$ they take the form

$$\begin{aligned} & V^{[j+1]}(x) - V^{[j+1]}((F + GK)x) \\ &= \text{Known Quantities} \\ & \left(\kappa^{[j]}(x) \right)' (R + G'PG) \\ &= \text{Known Quantities} \end{aligned}$$

The “Known Quantities” involve the terms of the Taylor polynomials of f, l and previously computed $V^{[i+1]}, \kappa^{[i]}$ for $1 \leq i < j$. Again the equations are linear in the unknowns $V^{[k+1]}, \kappa^{[k]}$, and the first equation does not involve $\kappa^{[k]}$. The eigenvalues of the linear operator

$$V^{[k+1]}(x) \mapsto V^{[k+1]}(x) - V^{[k+1]}((F + GK)x)$$

are of the form of 1 minus the products of $k + 1$ eigenvalues of $F + GK$.

We have written MATLAB code to solve these equations to any degree and in any dimensions. The code is quite fast. Later we shall present an example where $n = 4$ and $m = 2$. The code found the Taylor polynomials of the optimal cost to degree 6 and the optimal feedback to degree 5 in 0.12 s. on a laptop using 3.1 GHz Intel Core i5.

Completing the Squares

The infinite horizon optimal cost is certainly non-negative definite, and if we choose $Q > 0$ then it is positive definite. That implies that its quadratic part $V^{[2]}(x) = \frac{1}{2}x'Px$ is positive definite.

But its Taylor polynomial of degree $d + 1$,

$$V^{[2]}(x) + V^{[3]}(x) + \dots + V^{[d+1]}(x)$$

need not be positive definite for $d > 1$. This can lead to problems if we define this Taylor polynomial to be our terminal cost $V_f(x)$ because then the nonlinear program solver might return a negative cost $V_N(x)$ (9). The way around this difficulty is to “complete the squares.”

Theorem Suppose a polynomial $V(x)$ is of degrees two through $d + 1$ in n variables $x_1, \dots, x_n$. If the quadratic part of $V(x)$ is positive definite, then there exist a nonnegative definite polynomial $W(x)$ of degrees two through $2d$ such that the part of $W(x)$ that is of degrees two through $d + 1$ equals $V(x)$. Moreover, we know that $W(x)$ is nonnegative definite because it is the sum of n squares.

Proof. We start with the quadratic part of $V(x)$, because it is positive definite it must be of the form $\frac{1}{2}x'Px$ where P is a positive definite $n \times n$ matrix. We know that there is an orthogonal matrix T that diagonalizes P

$$T'PT = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix}$$

where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > 0$. We make the linear change of coordinates $x = Tz$. We shall show that $V(z) = V(Tz)$ can be extended with higher degree terms to a polynomial $W(z)$ of degrees two through $2d$ which is a sum of n squares. We do this degree by degree. We have already showed that the degree two part of $V(z)$ is a sum of n squares

$$\frac{1}{2} \sum_{i=1}^n \lambda_i z_i^2 = \frac{1}{2} (\lambda_1 z_1^2 + \dots + \lambda_n z_n^2)$$

The degrees two and three parts of $V(z)$ are of the form

$$\frac{1}{2} \sum_{i=1}^n \lambda_i z_i^2 + \sum_{i_1=1}^n \sum_{i_2=i_1}^n \sum_{i_3=i_2}^n \gamma_{i_1, i_2, i_3} z_{i_1} z_{i_2} z_{i_3}$$

Consider the expression

$$\delta_1(z) = z_1 + \sum_{i_2=1}^n \sum_{i_3=i_2}^n \delta_{1, i_2, i_3} z_{i_2} z_{i_3}$$

then

$$\begin{aligned} \frac{\lambda_1}{2} (\delta_1(z))^2 &= \frac{\lambda_1}{2} \left(z_1 + \sum_{i_2=1}^n \sum_{i_3=i_2}^n \delta_{1, i_2, i_3} z_{i_2} z_{i_3} \right)^2 \\ &= \frac{\lambda_1}{2} \left(z_1^2 + 2 \sum_{i_2=1}^n \sum_{i_3=i_2}^n \delta_{1, i_2, i_3} z_1 z_{i_2} z_{i_3} + \frac{1}{2} \left(\sum_{i_2=1}^n \sum_{i_3=i_2}^n \delta_{1, i_2, i_3} z_{i_2} z_{i_3} \right)^2 \right) \end{aligned}$$

Let $\delta_{1, i_2, i_3} = \frac{\gamma_{1, i_2, i_3}}{\lambda_1}$ then the degrees two and three parts of

$$V(z) - \frac{\lambda_1}{2} (\delta_1(z))^2$$

have no terms involving z_1 .

□

Next consider the expression

$$\delta_2(z) = z_2 + \sum_{i_2=2}^n \sum_{i_3=i_2}^n \delta_{2, i_2, i_3} z_{i_2} z_{i_3}$$

then

$$\begin{aligned} \frac{\lambda_2}{2} (\delta_2(z))^2 &= \frac{\lambda_2}{2} \left(z_2^2 + 2 \sum_{i_2=2}^n \sum_{i_3=i_2}^n \delta_{2, i_2, i_3} z_2 z_{i_2} z_{i_3} \right. \\ &\quad \left. + \frac{1}{2} \left(\sum_{i_2=2}^n \sum_{i_3=i_2}^n \delta_{2, i_2, i_3} z_{i_2} z_{i_3} \right)^2 \right) \end{aligned}$$

Let $\delta_{2, i_2, i_3} = \frac{\gamma_{2, i_2, i_3}}{\lambda_2}$ then the degrees two and three parts of

$$V(z) - \frac{\lambda_1}{2} (\delta_1(z))^2 - \frac{\lambda_2}{2} (\delta_2(z))^2$$

have no terms involving either z_1 or z_2 .

We continue on in this fashion defining $\delta_3(z), \dots, \delta_n(z)$ such that

$$V(z) - \sum_{i=1}^n \frac{\lambda_i}{2} (\delta_i(z))^2 = \sum_{i_1=1}^n \sum_{i_2=i_1}^n \sum_{i_3=i_2}^n \sum_{i_4=i_3}^n \gamma_{i_1, i_2, i_3, i_4} z_{i_1} z_{i_2} z_{i_3} z_{i_4} + O(z)^5$$

has no terms of degrees either two or three.

We redefine

$$\begin{aligned} \delta_1(z) &= z_1 + \sum_{i_2=1}^n \sum_{i_3=i_2}^n \delta_{1, i_2, i_3} z_{i_2} z_{i_3} \\ &\quad + \sum_{i_2=1}^n \sum_{i_3=i_2}^n \sum_{i_4=i_3}^n \delta_{1, i_2, i_3, i_4} z_{i_2} z_{i_3} z_{i_4} \end{aligned}$$

This does not change the degree two and three terms of $\frac{\lambda_1}{2} (\delta_1(z))^2$, and its degree four terms are of the form

$$\lambda_1 \sum_{i_2=1}^n \sum_{i_3=i_2}^n \sum_{i_4=i_3}^n \delta_{1, i_2, i_3, i_4} z_1 z_{i_2} z_{i_3} z_{i_4}$$

If we let $\delta_{1,i_2,i_3,i_4} = \frac{\gamma_{1,i_2,i_3,i_4}}{\lambda_1}$, then we cancel the degree four terms involving z_1 in

$$V(z) - \sum_{j=1}^n \frac{\lambda_j}{2} (\delta_j(z))^2$$

Next we redefine

$$\begin{aligned} \delta_2(z) = & z_2 + \sum_{i_2=2}^n \sum_{i_3=i_2}^n \delta_{2,i_2,i_3} z_{i_2} z_{i_3} \\ & + \sum_{i_2=2}^n \sum_{i_3=i_2}^n \sum_{i_4=i_3}^n \delta_{2,i_2,i_3,i_4} z_{i_2} z_{i_3} z_{i_4} \end{aligned}$$

Again this does not change the degree two and three terms of $\frac{\lambda_2}{2} (\delta_2(z))^2$, and its degree four terms are of the form

$$\lambda_2 \sum_{i_2=2}^n \sum_{i_3=i_2}^n \sum_{i_4=i_3}^n \delta_{2,i_2,i_3,i_4} z_{i_2} z_{i_3} z_{i_4}$$

If we let $\delta_{2,i_2,i_3,i_4} = \frac{\gamma_{2,i_2,i_3,i_4}}{\lambda_2}$ then we cancel the degree four terms involving z_2 in

$$V(z) - \sum_{j=1}^n \frac{\lambda_j}{2} (\delta_j(z))^2$$

We continue on in this fashion. The result is a sum of squares whose degree two through four terms equal $V(z)$.

Eventually we define

$$\begin{aligned} \delta_j(z) = & z_j + \sum_{i_2=j}^n \sum_{i_3=i_2}^n \delta_{j,i_2,i_3} z_{i_2} z_{i_3} + \cdots \\ & + \sum_{i_2=j}^n \cdots \sum_{i_d=i_{d-1}}^n \delta_{j,i_2,\dots,i_d} z_{i_2} \cdots z_{i_d} \end{aligned}$$

and

$$W(z) = \sum_{j=1}^n \frac{\lambda_j}{2} (\delta_j(z))^2$$

At degree $d = 3$, we solved a linear equation from the quadratic coefficients of $\delta_1(z), \dots, \delta_n(z)$

to the cubic coefficients of $V(z)$. We restricted the domain of this mapping by requiring that the quadratic part of $\delta_i(z)$ does not depend on $z_1, \dots, z_{i-1}$. This made the restricted mapping square; the dimensions of the domain and the range of the linear mapping are the same. We showed that the restricted mapping has a unique solution.

If we drop this restriction, then at degree d the overall dimension of the domain is

$$n \binom{n+d-1}{d}$$

while the dimension of the range is

$$\binom{n+d}{d+1}$$

So the unrestricted mapping has more unknowns than equations, and hence there are multiple solutions.

But the restricted solution that we constructed is a least squares solution to the unrestricted equations because $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > 0$. To see this consider the coefficient $\gamma_{1,1,2}$ of $z_1^2 z_2$. If we allow $\gamma_2(z)$ to have a term of the form $\delta_{2,1,1} z_1^2$, we can also cancel $\gamma_{1,1,2}$ by choosing $\delta_{1,1,2}$ and $\delta_{2,1,1}$ so that

$$\gamma_{1,1,2} = \lambda_1 \delta_{1,1,2} + \lambda_2 \delta_{2,1,1}$$

Because $\lambda_1 \geq \lambda_2$, a least squares solution to this equation is $\delta_{1,1,2} = \frac{\gamma_{1,1,2}}{\lambda_1}$ and $\delta_{2,1,1} = 0$. Because T is orthogonal, $W(x) = W(T'z)$ is also a least squares solution.

Example

Suppose we wish to stabilize a double pendulum to straight up. The first two states are the angles between the two links and straight up measured in radians counterclockwise. The other two states are their angular velocities. The controls are the torques applied at the base of the lower link and at the joint between the links. The links are assumed to be massless, the base link is one meter long,

and the other link is two meters long. There is a mass of two kilograms at the joint between the links and a mass of one kilogram at the tip of the upper link. There is linear damping at the base and the joint with both coefficients equal to 0.5 s^{-1} . The resulting continuous time dynamics is discretized using Euler's method with time step 0.1 s .

We simulated AHMPC with two different terminal costs and terminal feedbacks. In both cases the Lagrangian was

$$\frac{0.1}{2} (|x|^2 + |u|^2) \quad (23)$$

The first pair $V_f^2(x), \kappa_f^1(x)$ was found by solving the infinite horizon LQR problem obtained by taking the linear part of the dynamics around the operating point $x = 0$ and the quadratic Lagrangian (23). Then $V_f^2(x)$ is quadratic and positive definite and $\kappa_f^1(x)$ is a linear.

The second pair $V_f^6(x), \kappa_f^5(x)$ was found using the discrete time version of Al'brekht's method. Then $V_f^6(x)$ is the Taylor polynomial of the optimal cost to degree 6 and $\kappa_f^5(x)$ is the Taylor polynomial of the optimal feedback to degree 5. But $V_f^6(x)$ is not positive definite so we completed the squares as above to get degree 10 polynomial $V_f^{10}(x)$ which is positive definite.

In all the simulations we imposed the control constraint $|u|_\infty \leq 4$ and started at $x(0) = (0.9\pi, 0.9\pi, 0, 0)$ with an initial horizon of $N =$

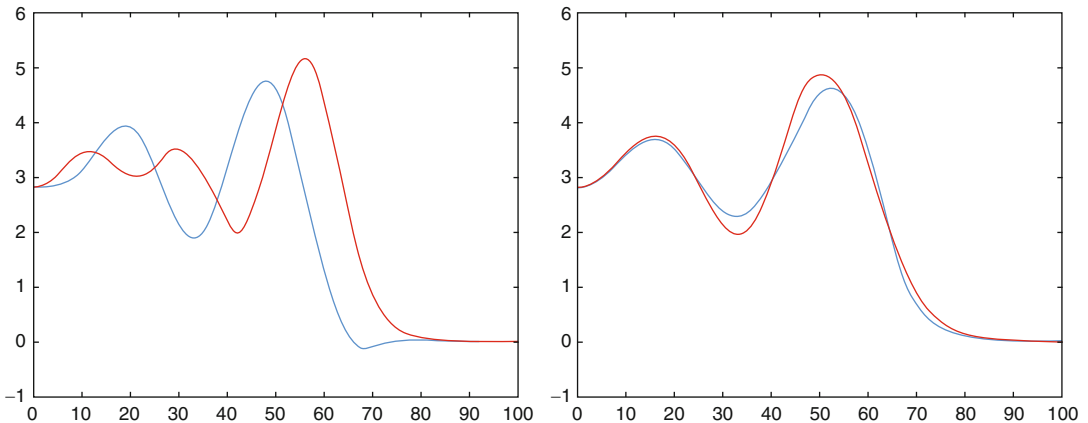
50 time steps. The extended horizon was kept constant at $M = 5$. The class K function was taken to be $\alpha(s) = s^2/10$.

If the Lyapunov or feasibility conditions were violated, then the horizon N was increased by 5 and the finite horizon nonlinear program was solved again without advancing the system. If after three tries the Lyapunov or feasibility conditions were still not satisfied, then the first value of the control sequence was used, the simulation was advanced one time step, and the horizon was increased by 5 (Fig. 1).

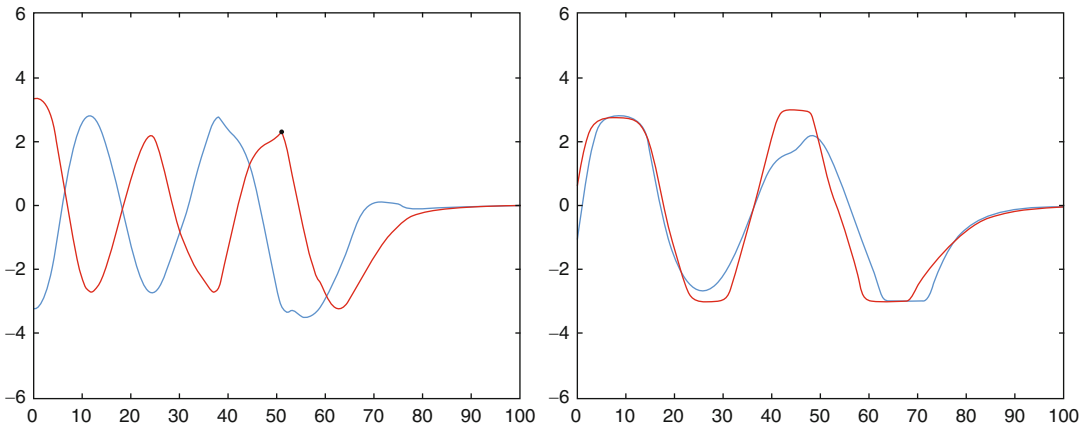
If the Lyapunov and feasibility conditions were satisfied over the extended horizon, then the simulation was advanced one time step, and the horizon N was decreased by 1.

The simulations were first run with no noise, and the results are shown in the following figures. Both methods stabilized the links to straight up in about $t = 80$ times steps (8 s). The degree $2d = 10$ terminal cost and the degree $d = 5$ terminal feedback seem to do it a little more smoothly with no overshoot and with shorter maximum horizon $N = 65$ versus $N = 75$ for LQR ($d = 1$) (Fig. 2).

The simulations were done on a MacBook Pro with a 3.1 GHz Intel Core i5 using MATLAB's `fmincon.m` with its default settings. We did supply `fmincon.m` the gradients of the objective functions, but we did not give it the Hessians. The cpu time for the degree $2d = 10$ terminal cost and the degree $d = 5$ terminal feedback was 5.01 s. This is probably too slow to control



Adaptive Horizon Model Predictive Control and Al'brekht's Method, Fig. 1 Angles, $d = 1$ on left, $d = 5$ on right



Adaptive Horizon Model Predictive Control and Al'brekht's Method, Fig. 2 Controls, $d = 1$ on left, $d = 5$ on right

pendulum in real time because the solver needs to return $u^*(0)$ in a fraction of a time step. But by using a faster solver than `fmincon.m`, coding the objective, the gradient, and the Hessian in C and compiling them, we probably could control the double pendula in real time. The CPU time for the LQR terminal cost and terminal feedback was 24.56 s so it is probably not possible to control the double pendula in real time using LQR terminal cost and terminal feedback (Fig. 3). Then we added noise to the simulations. At each advancing step a Gaussian random vector with mean zero and covariance 0.0001 times the identity was added to the state. The next figures show the results using the degree 10 terminal cost and degree 5 terminal feedback. Notice that the horizon starts at $T = 50$ but immediately jumps to $T = 65$, declines to $T = 64$, then jumps to $T = 69$ before decaying monotonically to zero. When the horizon $T = 0$ the terminal feedback is used. The LQR terminal cost and feedback failed to stabilize the noisy pendula. We stopped the simulation when $T > 1000$ (Fig. 4).

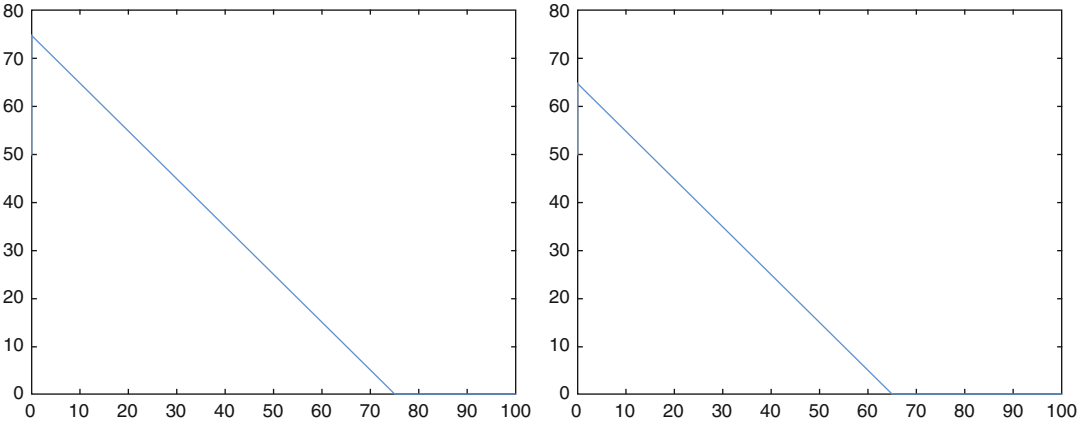
We also considered $d = 3$ so that after completing the squares the terminal cost is degree $2d = 6$ and the terminal feedback is degree $d = 3$. It stabilized the noiseless simulation with a maximum horizon of $N = 80$ which is greater than the maximum horizons for both $d = 1$ and $d = 5$. But it did not stabilize the noisy simulation. Perhaps the reason is revealed by Taylor polynomial approximations to $\sin x$ as

shown in Figure 5. The linear approximation in green overestimates the magnitude of $\sin x$, so the linear feedback is stronger than it needs to be to overcome gravity. The cubic approximation in blue underestimates the magnitude of $\sin x$ so the cubic feedback is weaker than it needs to be to overcome gravity. The quintic approximation in orange overestimates the magnitude of $\sin x$, so the quintic feedback is also stronger than it needs to be but by a lesser margin than the linear feedback to overcome gravity. This may explain why the degree 5 feedback stabilizes the noise free pendula in a smoother fashion than the linear feedback (Fig. 5).

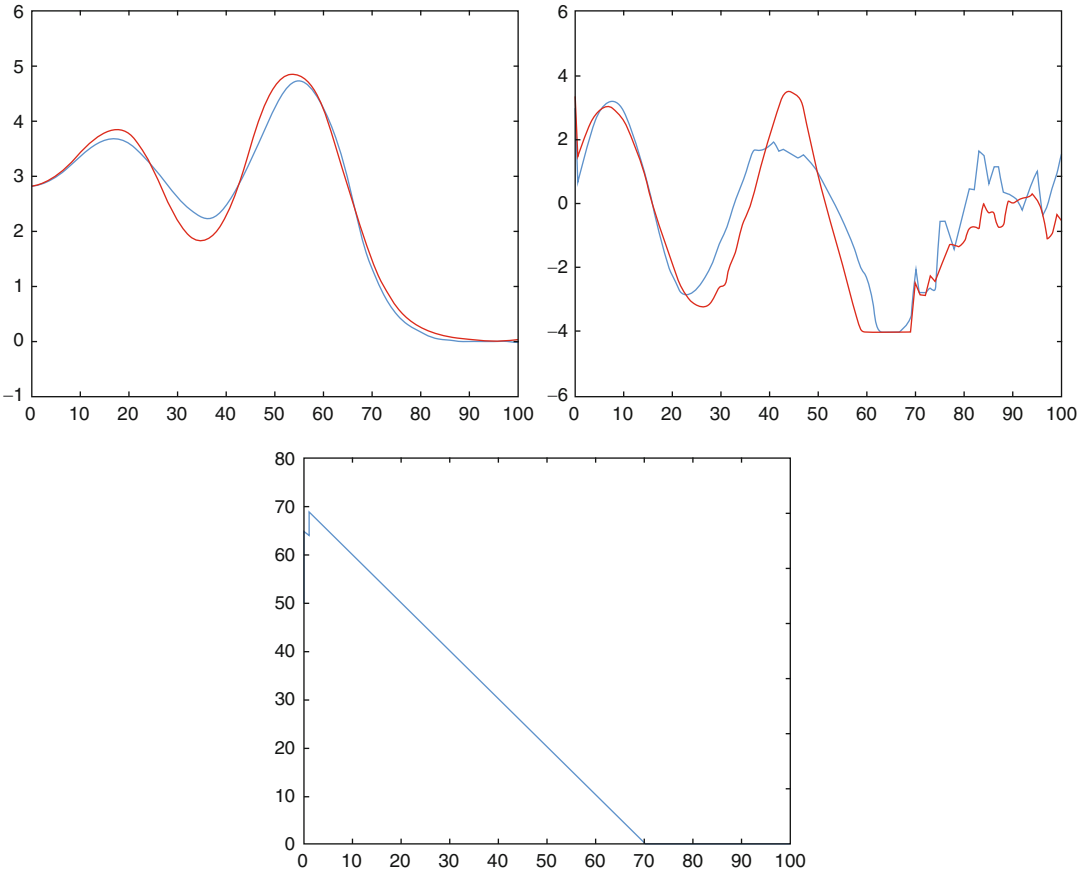
Conclusion

Adaptive Horizon Model Predictive Control is a scheme for varying the horizon length in Model Predictive Control as the stabilization process evolves. It adapts the horizon in real time by testing Lyapunov and feasibility conditions on extensions of optimal trajectories returned by the nonlinear program solver. In this way it seeks the shortest horizons consistent with stabilization.

AHMPC requires a terminal cost and terminal feedback that stabilizes the plant in some neighborhood of the operating point but that neighborhood need not be known explicitly. Higher degree Taylor polynomial approximations to the optimal cost and the optimal feedback of the cor-



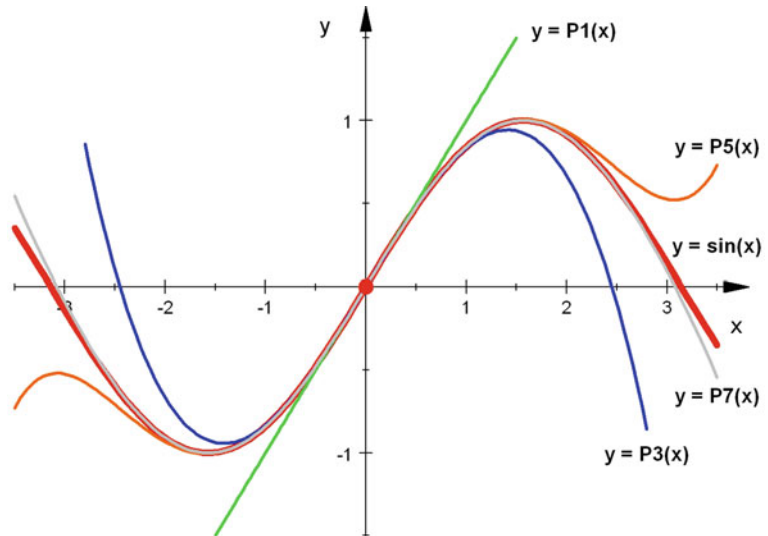
Adaptive Horizon Model Predictive Control and Al'brekht's Method, Fig. 3 Horizons, $d = 1$ on left, $d = 5$ on right



Adaptive Horizon Model Predictive Control and Al'brekht's Method, Fig. 4 Noisy angles, controls and horizons

Adaptive Horizon Model Predictive Control and Al'brekht's Method,

Fig. 5 Taylor approximations to $y = \sin x$



responding infinite horizon optimal control problems can be found by an extension of Al'brekht's method (Aguilar and Krener 2012). The higher degree Taylor polynomial approximations to optimal cost need not be positive definite, but they can be extended to nonnegative definite polynomials by completing the squares. These nonnegative definite extensions and the Taylor polynomial approximations to the optimal feedback can be used as terminal costs and terminal feedbacks in AHMPC. We have shown by an example that a higher degree terminal cost and feedback can outperform using LQR to define a degree two terminal cost and a degree one terminal feedback.

Cross-References

- [Linear Quadratic Optimal Control](#)
- [Lyapunov's Stability Theory](#)
- [Nominal Model-Predictive Control](#)
- [Optimization Algorithms for Model Predictive Control](#)

Bibliography

- Aguilar C, Krener AJ (2012) Numerical solutions to the dynamic programming equations of optimal control. In: Proceedings of the 2012 American control conference
- Antsaklis PJ, Michel AN (1997) Linear systems. McGraw Hill, New York
- Al'brecht EG (1961) On the optimal stabilization of nonlinear systems. PMM-J Appl Math Mech 25:1254–1266
- Droge G, Egerstedt M (2011) Adaptive time horizon optimization in model predictive control. In: Proceedings of the 2011 American control conference
- Giselsson P (2010) Adaptive nonlinear model predictive control with suboptimality and stability guarantees. In: Proceedings of the 2010 conference on decision and control
- Grimm G, Messina MJ, Tuna SE, Teel AR (2005) Model predictive control: for want of a local control Lyapunov function, all is not lost. IEEE Trans Autom Control 50:546–558
- Guene L (2012) NMPC without terminal constraints. In: Proceedings of the 2012 IFAC conference on nonlinear model predictive control, pp 1–13
- Guene L, Pannek J, Sehafer M, Worthmann K (2010) Analysis of unconstrained nonlinear MPC schemes with time varying control horizon. SIAM J Control Optim 48:4938–4962
- Khalil HK (1996) Nonlinear systems, 2nd edn. Prentice Hall, Englewood
- Kim T-H, Sugie T (2008) Adaptive receding horizon predictive control for constrained discrete-time linear systems with parameter uncertainties. Int J Control 81:62–73
- Krener AJ (2018) Adaptive horizon model predictive control. In: Proceedings of the IFAC conference on modeling, identification and control of nonlinear systems, Guadalajara, 2018
- Michalska H, Mayne DQ (1993) Robust receding horizon control of constrained nonlinear systems. IEEE Trans Autom Control 38:1623–1633

- Page SF, Dolia AN, Harris CJ, White NM (2006) Adaptive horizon model predictive control based sensor management for multi-target tracking. In: Proceedings of the 2006 American control conference
- Pannek J, Worthmann K (2011) Reducing the Predictive Horizon in NMPC: an algorithm based approach. Proc IFAC World Congress 44:7969–7974
- Polak E, Yang TH (1993a) Moving horizon control of linear systems with input saturation and plant uncertainty part 1. robustness. Int J Control 58: 613–638
- Polak E, Yang TH (1993b) Moving horizon control of linear systems with input saturation and plant uncertainty part 2. disturbance rejection and tracking. Int J Control 58:639–663
- Polak E, Yang TH (1993c) Moving horizon control of nonlinear systems with input saturation, disturbances and plant uncertainty. Int J Control 58: 875–903
- Rawlings JB, Mayne DQ (2009) Model predictive control: theory and design. Nob Hill Publishing, Madison

Adaptive Human Pilot Models for Aircraft Flight Control

Ronald A. Hess
University of California, Davis, CA, USA

Abstract

Loss-of-control incidents in commercial and military aircraft have prompted a renewed interest in the manner in which the human pilot adapts to sudden changes or anomalies in the flight control system which augments the fundamental aircraft dynamics. A simplified control-theoretic model of the human pilot can be employed to describe adaptive pilot behavior in the so-called primary control loops and to shed light upon the limits of human control capabilities.

Keywords

Control-theoretic pilot models · Pilot crossover model · Aircraft loss of control · Single and multi-axis pilot control

An Introduction to Control-Theoretic Models of the Human Pilot

A control-theoretic model of the human pilot refers to a representation in which the pilot is modeled as an element in a control system. Typically, this means that the dynamics of the pilot are represented by a transfer function, and this transfer function is placed in a block diagram where the pilot's control inputs affect the aircraft's responses. In its simplest form, such a representation is shown in Fig. 1.

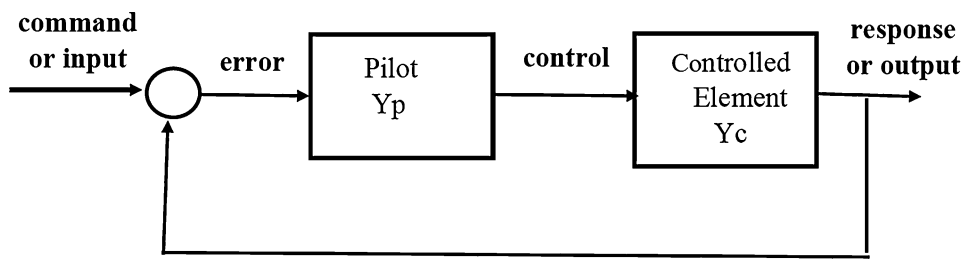
As a more concrete representation, “command” might indicate a command to the pilot/aircraft system in the form of a desired aircraft pitch attitude. “Control” might represent an aircraft elevator motion produced by the pilot, whose dynamics are represented by the linear, time-invariant transfer function $Y_p(s)$. “Controlled element” is the pertinent aircraft transfer function, $Y_c(s)$ representing the manner in which aircraft pitch attitude (“response”) is dependent upon the pilot's control input.

Perhaps the most definitive representation of human pilot dynamics in tasks such as that shown in Fig. 1 is the “crossover” model of the combined human pilot and controlled element (McRuer and Krendel 1974), shown in Eq. 1.

$$Y_p(s)Y_c(s) \approx \omega_c \frac{e^{-\tau_e s}}{s} \quad (1)$$

This is called the *crossover model* of the human pilot because it is a neighborhood of the frequency region where $|Y_p(j\omega)Y_c(j\omega)| \approx 1.0$, where the model is valid. Figure 2 shows a typical measured $Y_p Y_c(s)$ plotted on a Bode diagram and compared to the crossover model. The controlled element dynamics in this case were $Y_c = K_c/s$ (McRuer et al. 1965). The phase lags apparent for $\omega > \omega_c$ are attributable to the time delay τ_e in Eq. 1 and limit the magnitude of an open-loop crossover frequency for stable closed-loop operation of the pilot/aircraft system.

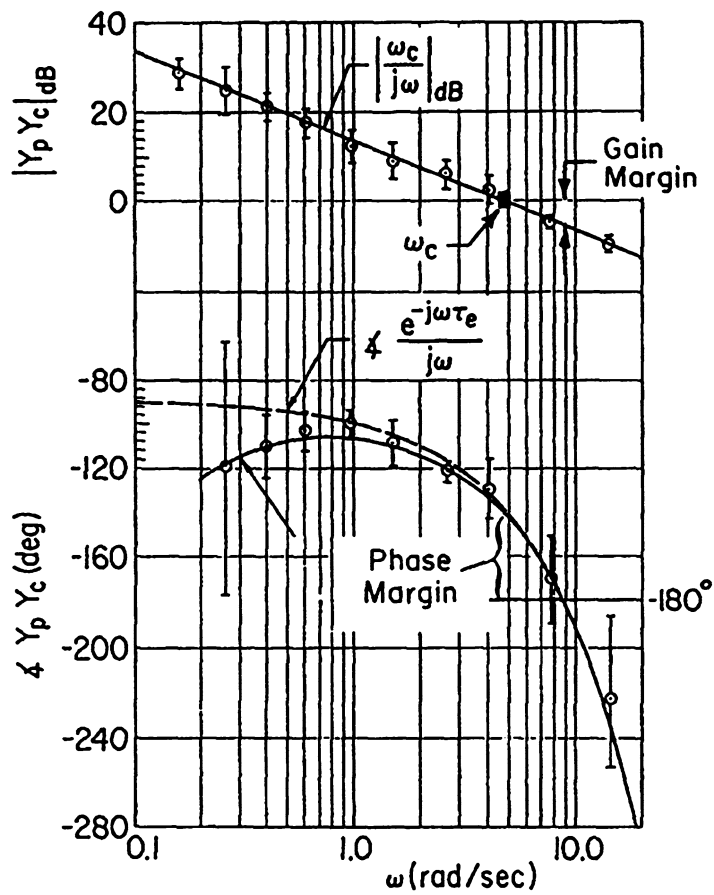
A more complete discussion of linear, time-invariant control-theoretic models of the human pilot can be found in Hess (1997).



Adaptive Human Pilot Models for Aircraft Flight Control, Fig. 1 A block diagram of a control-theoretic model of the human pilot in a single-loop tracking task. Both Y_p and Y_c are represented by transfer functions

Adaptive Human Pilot Models for Aircraft Flight Control,

Fig. 2 A Bode diagram of the measured $Y_p Y_c$ in a single-loop tracking task is shown. In this example, $Y_c = K_c/s$

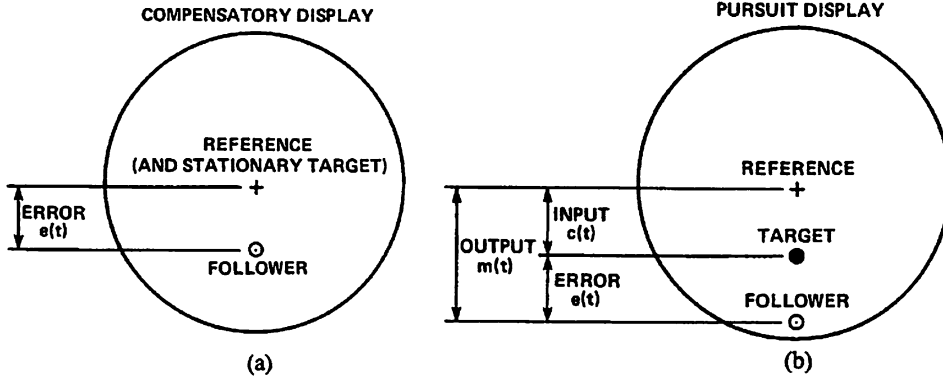


A Simplified Pursuit Control Model of the Human Pilot

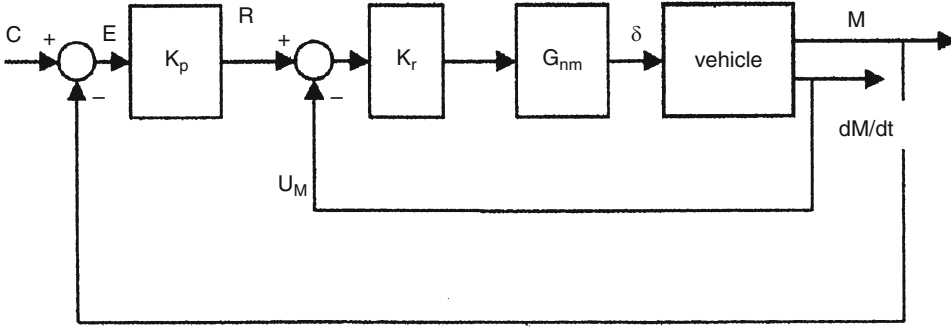
Pilot control behavior in command-following tracking tasks is often categorized as either *compensatory* or *pursuit* depending upon the type of visual information available to the pilot (Krendel and McRuer 1960; Hess 1981). Figure 3 compares two cockpit displays, one providing

compensatory information (error only) and one providing pursuit information (command and error). Figure 4 is a block diagram representation of the pursuit tracking pilot model to be utilized herein (Hess 2006).

It should be emphasized that the analysis procedure to be described utilizes a sequential loop closure formulation for piloted control. For example, in Fig. 1 the command or input may



Adaptive Human Pilot Models for Aircraft Flight Control, Fig. 3 Two cockpit displays can be created showing the difference between compensatory and pursuit tracking formats



Adaptive Human Pilot Models for Aircraft Flight Control, Fig. 4 A pursuit tracking model of the human pilot from Hess (2006). In pursuit tracking behavior, vehicle

output and output rate are assumed to be directly available to the pilot, here represented by K_p , K_r , and G_{nm}

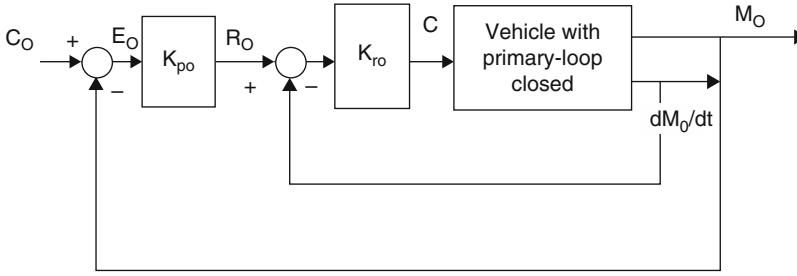
be an altitude command in which the pilot is using pitch attitude as a primary loop closure to control aircraft altitude. The element denoted G_{nm} in Fig. 4 represents the simplified dynamics of the neuromuscular system driving the cockpit inceptor. Further information on neuromuscular system models can be found in McRuer and Magdalenio (1966). In the model of Fig. 4, G_{nm} is given by

$$G_{nm} = \frac{10^2}{s^2 + 2(707)s + 10^2} \quad (2)$$

Figure 5 shows the manner in which Fig. 4 can be expanded to include such sequential closures. In Fig. 5, the “vehicle with primary loop closed” would simply be the M/C transfer function in Fig. 4. As discussed in Hess (2006),

the adjustment rules for selecting K_{ro} and K_{po} are as follows: K_{ro} is chosen as the gain value that results in a crossover frequency for the transfer function $K_{ro} \cdot [\dot{M}_o/C]$ equal to that for the adjacent inner loop and M/E in Fig. 4. K_{po} is chosen to provide a desired open-loop crossover frequency in the outer-loop transfer function M_o/E_o . Nominally, this crossover frequency will be one-third the value for the crossover frequency in the inner loop of Fig. 4.

A distinct advantage of the pilot model of Fig. 4 lies in the fact that only two parameters, the gains K_p and K_r , are necessary to implement the model in a primary control loop. Hess (2006) demonstrates the manner in which this is accomplished. Basically, K_r is chosen so that the Bode diagram of the transfer function $\frac{\dot{M}}{R}(s)$ exhibits a 10 dB amplitude peaking near 10 rad/s. K_p is



Adaptive Human Pilot Models for Aircraft Flight Control, Fig. 5 A multi-loop extension of the pilot model of Fig. 4. The block “vehicle with primary loop closed”

implies that the pilot model of Fig. 4 has been used in a primary-loop closure

then chosen to yield a 2.0 rad/s crossover frequency in the Bode diagram of $\frac{M}{C}(s)$. Hess (2006) demonstrates the ability of the two-parameter pilot model to reproduce pilot crossover model characteristics for a variety of vehicle dynamics in Fig. 4. The fact that only two gain values are necessary to create the pilot model suggests that extending the representation of Fig. 4 to encompass adaptive pilot behavior may be warranted.

A Rationale for Modeling the Adaptive Human Pilot

Loss of control is the leading cause of jet fatalities worldwide (Jacobson 2010). Aside from their frequency of occurrence, aviation accidents resulting from loss of control seize the public’s attention by yielding large number of fatalities in a single event. The rationale for modeling the human pilot in potential loss of control accidents can be best summarized as follows (McRuer and Jex 1967):

- (1) To summarize behavioral data
- (2) To provide a basis for rationalization and understanding of pilot control actions
- (3) To be used in conjunction with vehicle dynamics in forming predictions or in explaining the behavior of pilot-vehicle systems

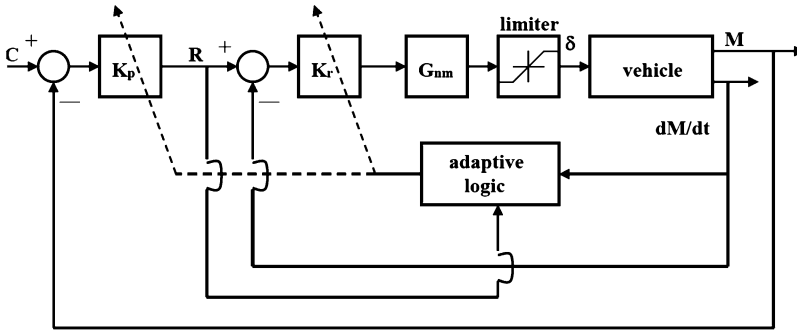
To this list might be added: to be used in concert with human-in-the-loop flight simulation to offer a fundamental rationale for aircraft loss of control.

Modeling the Adaptive Human Pilot

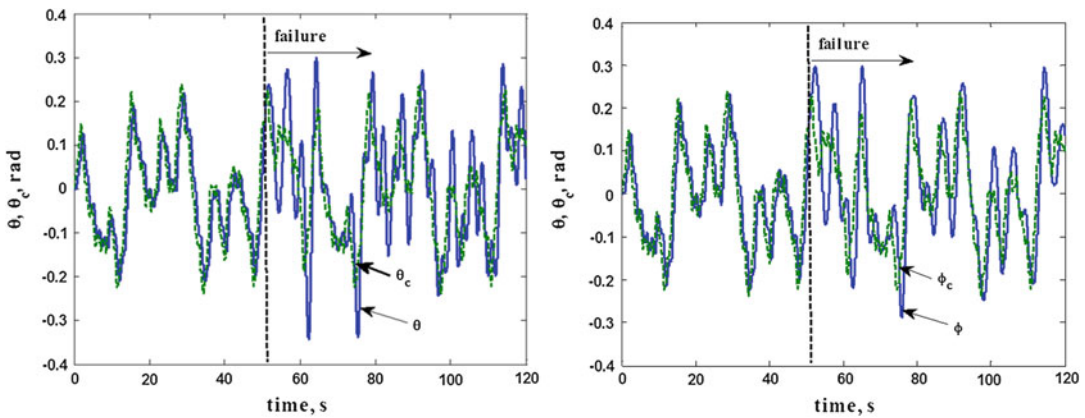
Studies devoted to modeling the human pilot’s adaptive capabilities date from the mid-1960s. See, for example, Elkind and Miller (1966), Weir (1968), Young (1969), Niemela and Krendel (1974), and Hess (2016). As used herein, “adaptive behavior” will refer to the human pilot’s ability to adopt dynamic characteristics that allow control of sudden changes in the dynamics of the aircraft being controlled. This is in contrast to a human “learning” to control a dynamic system through a training process. Figure 6 from Hess (2016) shows the fundamental adaptive structure of the adaptive human pilot in a primary control loop.

In Hess (2016) the adaptive logic governing changes in K_p and K_r are defined. The reader is referred to that document for specifics. The logic is based upon certain guidelines:

- (1) The adjustments to K_p and K_r must be predicated upon observations that can easily be made by the human pilot.
- (2) The logic driving the adjustments must be predicated on information available to the human pilot.
- (3) The post-adapted pilot models must follow the dictates of the crossover model of the human pilot (McRuer and Krendel 1974).
- (4) Performance improvement with adaptation must occur relatively quickly. This guideline is supported by the research reported by Hess (2009).



Adaptive Human Pilot Models for Aircraft Flight Control, Fig. 6 Modifying the model of Fig. 4 to enable modeling of pilot adaptive behavior



Adaptive Human Pilot Models for Aircraft Flight Control, Fig. 7 Simulink® simulation of helicopter pitch and roll responses before and after system “failure.” Note

that, although there have been significant reductions in stability augmentation gains and control inceptor sensitivities, tracking performance remains reasonably constant

It should be noted that in Hess (2016), the adaptive pilot model is not limited to one axis of control. Examples in which the pilot is controlling two axes, e.g., aircraft pitch and roll, are presented.

An Example of Adaptive Pilot Modeling

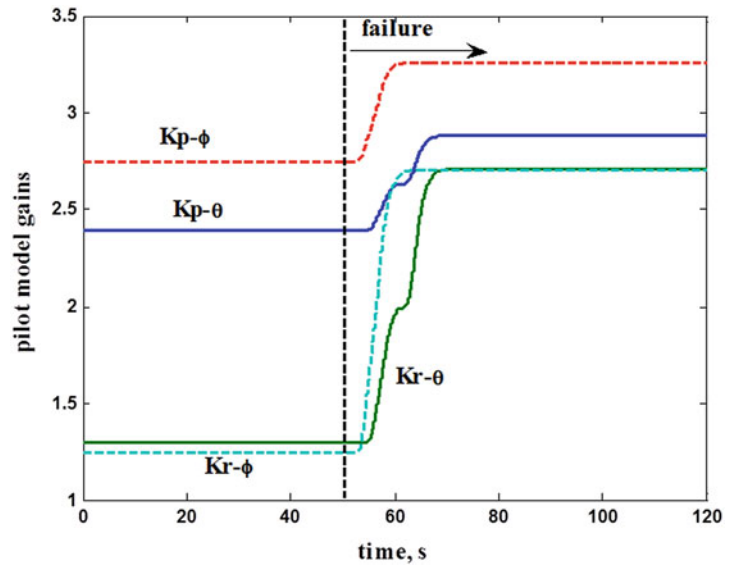
The following example deals with piloted control of a hovering helicopter. The helicopter dynamics and flight control system are taken from Hall and Bryson (1973). Two vehicle axes are being controlled by the pilot, pitch attitude (θ), and roll attitude (ϕ). Random-appearing sums of sinusoids provided commands to the pitch and roll axes (θ_c and ϕ_c). Of particular importance is

the fact that control response coupling was in evidence in the vehicle model, i.e. inputs to control pitch attitude also affected roll attitude and vice-versa. The vehicle “failures” were created by reducing the gain of the pitch and roll stability augmentation systems by a factor of 10 and reducing the sensitivity of the cockpit inceptors by a factor of 5. The failures were introduced at $t=50$ s in a 120 s simulation conducted using MATLAB Simulink®. Details can be found in Hess (2016).

Figure 7 shows the pitch and roll attitude responses before and after the failure with the dashed line representing the pitch and roll command (θ_c , ϕ_c) and the solid lines representing the corresponding pilot/vehicle responses. Figure 8 shows the adaptive pilot model gains K_p and K_r for each loop.

Adaptive Human Pilot Models for Aircraft Flight Control, Fig. 8

Adaptive pilot model gains in the pitch and roll attitude control loops. As might be expected, the sharp reduction in control and inceptor sensitivities in the failure is accommodated by the adaptive model by increases in K_p and K_r in each control axis. Of equal importance is the fact that crossover model characteristics were in evidence when the K_p and K_r assumed their final values, as they were at the initiation of the adaptive changes



Cross-References

- [Aircraft Flight Control](#)
- [Pilot-Vehicle System Modeling](#)

Recommended Reading

An excellent textbook aimed at advanced undergraduates and graduate student interested in manual control of dynamic systems has been authored by Jagacinski and Flach (2003).

Bibliography

- Elkind JJ, Miller DC (1966) Process of adaptation by the human controller. In: Proceedings of the second annual NASA-university conference on manual control
- Hall WE Jr, Bryson AE Jr (1973) Inclusion of rotor dynamics in controller design for helicopters. *J Aircr* 10(4):200–206
- Hess RA (1981) Pursuit tracking and higher levels of skill development in the human pilot. *IEEE Trans Syst Man Cybern SMC* 11(4):262–273
- Hess RA (1997) Feedback control models – manual control and tracking. In: Salvendy G (ed) *Handbook of human factors and ergonomics*, 2nd edn, Chap. 38. Wiley, New York
- Hess RA (2006) Simplified approach for modelling pilot pursuit control behavior in multi-loop flight control tasks. In: *Proc Inst Mech Eng J Aerosp Eng* 220(G2):85–102
- Hess RA (2009) Modeling pilot control behavior with sudden changes in vehicle dynamics. *J Aircr* 46(5): 1584–1592
- Hess RA (2016) Modeling human pilot adaptation to flight control anomalies and changing task demands. *J Guid Control Dyn* 39(3):655–666
- Jacobson SR (2010) Aircraft loss of control causal factors and mitigation challenges. In: *AIAA guidance navigation and control conference*, 2–5 Aug 2010
- Jagacinski RJ, Flach JM (2003) *Control theory for humans: quantitative approaches to modeling performance*. Lawrence Erlbaum Associates, Mahwah
- Krendel ES, McRuer DT (1960) A servomechanisms approach to skill development. *J Frankl Inst* 269(1): 24–42
- McRuer DT, Jex HR (1967) Review of quasi-linear pilot models. *IEEE Trans Hum Factors Electron HFE* 8(3):231–249
- McRuer DT, Krendel E (1974) *Mathematical models of human pilot behavior*, AGARDograph No. 188
- McRuer DT, Magdaleno RE (1966) Experimental validation and analytical elaboration for models of the pilot's neuromuscular subsystem in tracking tasks, NASA CR-1757
- McRuer D, Graham D, Krendel E (1965) *Human pilot dynamics in compensatory systems*. Air Force Flight Dynamics Lab, report AFFDL-TR-65-15
- Niemela R, Krendel ES (1974) Detection of a change in plant dynamics in a man-machine system. In: *Proceedings of the tenth annual NASA-university conference on manual control*
- Weir DH (1968) Applications of the pilot transition response model to flight control system failure analysis. In: *Proceedings of the fourth annual NASA-university conference on manual control*
- Young LR (1969) On adaptive manual control. *Ergonomics* 12(4):292–331

Advanced Manipulation for Underwater Sampling

Giuseppe Casalino
University of Genoa, Genoa, Italy

Keywords

Kinematic control law (KCL) ·
Manipulator · Motion priorities

Abstract

This entry deals with the kinematic self-coordination aspects to be managed by parts of underwater floating manipulators, whenever employed for sample collections at the seafloor.

Kinematic self-coordination is here intended as the autonomous ability exhibited by the system in closed loop specifying the most appropriate reference velocities for its main constitutive parts (i.e., the supporting vehicle and the arm) in order to execute the sample collection with respect to both safety and best operability conditions for the system while also guaranteeing the needed “execution agility” in performing the task, particularly useful in case of underwater repeated collections. To this end, the devising and employment of a unifying control framework capable of guaranteeing the above properties will be outlined.

Such a framework is however intended to only represent the so-called Kinematic Control Layer (KCL) overlaying a Dynamic Control Layer (DCL), where the overall system dynamic and hydrodynamic effects are suitably accounted for, to the benefit of closed loop tracking of the reference system velocities. Since the DCL design is carried out in a way which is substantially independent from the system mission(s), it will not constitute a specific topic of this entry, even if some orienting references about it will be provided.

At this entry’s end, as a follow-up of the resulting structural invariance of the devised KCL framework, future challenges addressing much wider and complex underwater applications will be foreseen, beyond the here-considered sample collection one.

Introduction

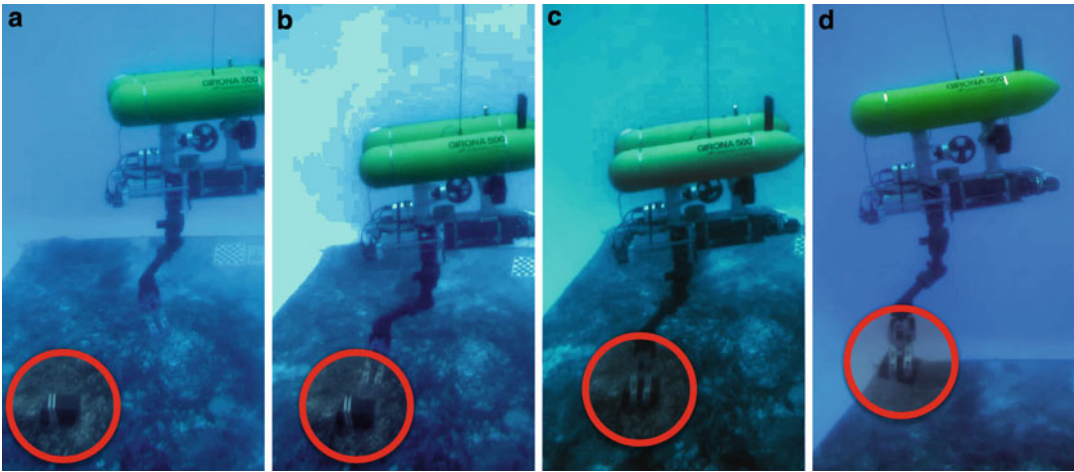
An automated system for underwater sampling is here intended to be an autonomous underwater floating manipulator (see Fig. 1) capable of collecting samples corresponding to an a priori assigned template. The snapshots of Fig. 1 outline the most recent realization of a system of this kind (completed in 2012 within the EU-funded project TRIDENT; Sanz et al. 2012) when in operation, which is characterized by a vehicle and an endowed 7-dof arm exhibiting comparable masses and inertia, thus resulting in potentially faster and more agile designs than the very few similar previous realizations.

Its general the operational mode consists in exploring an assigned area of the seafloor, while executing a collection each time a feature corresponding to the assigned template is recognized (by the vehicle endowed with a stereovision system) as a sample to be collected.

Thus the autonomous functionalities to be exhibited are the following (to be sequenced as they are listed on an event-driven basis): (1) explore an assigned seabed area while visually performing model-based sample recognitions, (2) suspend the exploration and grasping a recognized sample, (3) deposit the sample inside an endowed container, and (4) then restart exploring till the next recognized sample.

Functionalities (1) and (4), since they do not require the arm usage, naturally reenter within the topics of navigation, patrolling, visual mapping, etc., which are typical of traditional AUVs and consequently will not be discussed here. Only functionality (2) will be discussed, since it is most distinctive of the considered system (often termed as I-AUV, with “I” for “Intervention”) and because functionality (3) can be established along the same lines of (2) as a particular simpler case.

By then focusing on functionality (2), we must note how the sample grasping ultimate



Advanced Manipulation for Underwater Sampling, Fig. 1 Snapshots showing the underwater floating manipulator TRIDENT when autonomously picking an identified object

objective, which translates into a specific position/attitude to be reached by the end-effector, must however be achieved within the preliminary fulfillment of also other objectives, each one reflecting the need of guaranteeing the system operating within both its safety and best operability conditions. For instance, the arm's joint limits must be respected and the arm singular postures avoided. Moreover, since the sample position is estimated via the vehicle with a stereo camera, the sample must stay grossly centered inside its visual cone, since otherwise the visual feedback would be lost and the sample search would need to start again. Also, the sample must stay within suitable horizontal and vertical distance limits from the camera frame, in order for the vision algorithm to be well performing. And furthermore, in these conditions the vehicle should be maintained with an approximately horizontal attitude, for energy savings.

With the exception of the objective of making the end-effector position/attitude reaching the grasping position, which is clearly an equality condition, its related safety/enabling objectives are instead represented by a set of inequality conditions (involving various system variables) whose achievement (accordingly with their safety/enabling role) must therefore deserve the highest priority.

System motions guaranteeing such prioritized objective achievements should moreover allow for a concurrent management of them (i.e., avoiding a sequential motion management whenever possible), which means requiring each objective progressing toward its achievement, by at each time instant only exploiting the residual system mobility allowed by the current progresses of its higher priority objectives. Since the available system mobility will progressively increase during time, accordingly with the progressive achievement of all inequality objectives, this will guarantee the grasping objective to be also completed by eventually progressing within adequate system safety and best operability conditions. In this way the system will also exhibit the necessary “agility” in executing its maneuvers, in a way faster than in case they were executed on a sequential motion basis.

The devising of an effective way to incorporate all the inequality and equality objectives within a uniform and computationally efficient task-priority-based algorithmic framework for underwater floating manipulators has been the result of the developments outlined in the next section.

The developed framework however solely represents the so-called Kinematic Control Layer (KCL) of the overall control architecture, that is, the one in charge of closed-loop real-time

control generating the system velocity vector y as a reference signal, to be in turn concurrently tracked, via the action of the arm joint torques and vehicle thrusters, by an adequate underlying Dynamic Control Layer (DCL), where the overall dynamic and hydrodynamic effects are kept into account to the benefit of such velocity tracking. Since the DCL can actually be designed in a way substantially independent from the system mission(s), it will not constitute a specific topic of this entry. Its detailed dynamic-hydrodynamic model-based structuring, also including a stability analysis, can be found in Casalino (2011), together with a more detailed description of the upper-lying KCL, while more general references on underwater dynamic control aspects can be found, for instance, in Antonelli (2006).

Task-Priority-Based Control of Floating Manipulators

The above-outlined typical set of objectives (of inequality and/or equality types) to be achieved within a sampling mission are here formalized. Then some helpful generalizing definitions are given, prior to presenting the related unifying task-priority-based algorithmic framework to be used.

Inequality and Equality Objectives

One of the objectives, of inequality type, related to both arm safety and its operability is that of maintaining each joint within corresponding minimum and maximum limits, that is,

$$q_{lm} < q_i < q_{im}; \quad i = 1, 2, \dots, 7$$

Moreover, in order to have the arm operating with dexterity, its manipulability measure (Yoshikawa 1985; Nakamura 1991) must ultimately stay above a minimum threshold value, thus also requiring the achievement of the inequality type objective

$$\mu > \mu_m$$

While the above objectives arise from inherently scalar variables, other objectives instead arise as

conditions to be achieved within the Cartesian space, where each one of them can be conveniently expressed in terms of the modulus associated to a corresponding Cartesian vector variable.

To be more specific, let us, for instance, refer to the need of avoiding the occlusions between the sample and the stereo camera, which might occasionally occur due to the arm link motions. Then such need can be, for instance, translated into the ultimate achievement of the following set of inequalities, for suitable chosen values of the boundaries

$$\|l\| > l_m; \quad \|\tau\| > \tau_m; \quad \|\eta\| < \eta_M$$

where l is the vector lying on the vehicle x-y plane, joining the arm elbow with the line parallel to the vehicle z-axis and passing through camera frame origin, as sketched in Fig. 2a. Moreover η is the misalignment vector formed by vector τ also lying on the vehicle x-y plane, joining the lines parallel to the vehicle z-axis and, respectively, passing through the elbow and the end-effector origin.

As for the vehicle, it must keep the object of interest grossly centered in the camera frame (see Fig. 2b), thus meaning that the modulus of the orientation error ξ , formed by the unit vector n_p of vector p from the sample to the camera frame and the unit vector k_c of the z-axis of the camera frame itself, must ultimately satisfy the inequality

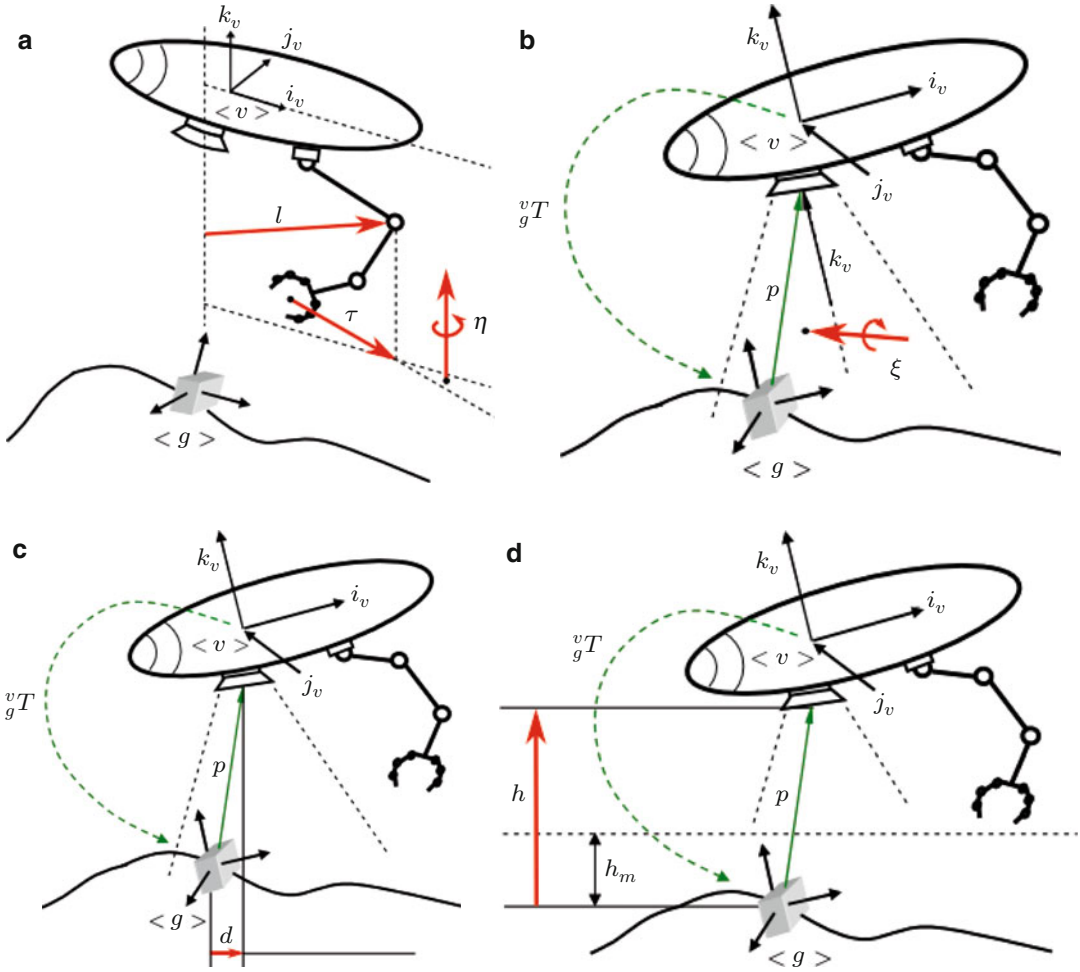
$$\|\xi\| < \xi_M$$

Furthermore, the camera must also be closer than a given horizontal distance d_M to the vertical line passing through the sample, and it must lie between a maximum and minimum height with respect to the sample itself, thus implying the achievement of the following inequalities (Fig. 2c, d):

$$\|d\| < d_M; \quad h_m < \|h\| < h_M$$

Also since the vehicle should exhibit an almost horizontal attitude, this further requires the achievement of the following additional inequality:

$$\|\phi\| < \phi_M$$



Advanced Manipulation for Underwater Sampling, Fig. 2 Vectors allowing for the definition of some inequality objectives in the Cartesian space: (a) camera occlusion, (b) camera centering, (c) camera distance, (d) camera height

with ϕ the misalignment vector formed by the absolute vertical unit vector k_o with the vehicle z-axis one k_v .

And finally the end-effector must eventually reach the sample, for then picking it. Thus the following, now of equality type, objectives must also be ultimately achieved, where r is the position error and θ the orientation one of the end-effector frame with respect to the sample frame

$$\|r\| = 0; \quad \|\vartheta\| = 0$$

As already repeatedly remarked, the achievement of the above inequality objectives (since related to the system safety and/or its best operability)

must globally deserve a priority higher than the last equality.

Basic Definitions

The following definitions only regard a generic vector $s \in R^3$ characterizing a corresponding generic objective defined in the Cartesian space (for instance, with the exclusion of the joint and manipulability limits, all the other above-reported objectives). In this case the vector is termed to be the error vector of the objective, and it is assumed measured with components on the vehicle frame. Then its modulus

$$\sigma \doteq \|s\|$$

is termed to be the error, while its unit vector

$$n \doteq s/\sigma; \quad \sigma \neq 0$$

is accordingly denoted as the unit error vector. Then the following differential Jacobian relationship can always be evaluated for each of them:

$$\dot{s} = Hy$$

where $y \in R^N$ ($N = (7 + 6)$ for the system of Fig. 1) is the stacked vector composed of the joint velocity vector $\dot{q} \in R^7$, plus the stacked vector $v \in R^6$ of the absolute vehicle velocities (linear and angular) with components on the vehicle frame and with $\dot{s}$ clearly representing the time derivative of vector s itself, as seen from the vehicle frame and with components on it (see Casalino (2011) for details on the real-time evaluation of Jacobian matrices H).

Obviously, for the time derivative $\dot{\sigma}$ of the error, also the following differential relationship holds

$$\dot{\sigma} = n^T Hy$$

Further, to each error variable σ , a so-called error reference rate is real time assigned of the form

$$\dot{\sigma} = -\gamma(\sigma - \sigma^o)\alpha(\sigma)$$

where for equality objectives σ^o is the target value and $\alpha(\sigma) \equiv 1$, while for inequality ones, σ^o is the threshold value and $\alpha(\sigma)$ is a left-cutting or right-cutting (in correspondence of σ^o) smooth sigmoidal activation function, depending on whether the objective is to force σ to be below or above σ^o , respectively.

In case $\dot{\sigma}$ could be exactly assigned to its corresponding error rate $\dot{\sigma}$, it would consequently smoothly drive σ toward the achievement of its associated objective. Note however that for inequality objectives, it would necessarily impose $\dot{\sigma} = 0$ in correspondence of a point located inside the interval of validity of the inequality objective itself, while instead such an error rate zeroing effect should be relaxed, for allowing the helpful subsequent system mobility increase, which allows for further progress toward other

lower priority control objectives. Such a relaxation aspect will be dealt with soon.

Furthermore, in correspondence of a reference error rate $\dot{\sigma}$, the so-called reference error vector rate can also be defined as

$$\dot{s} \doteq n\dot{\sigma}$$

that for equality objectives requiring the zeroing of their error σ simply becomes

$$\dot{s} \doteq -\gamma s$$

whose evaluation, since not requiring its unit vector n , will be useful for managing equality objectives.

Finally note that for each objective not defined in the Cartesian space (like, for instance, the above joint limits and manipulability), the corresponding scalar error variable, its rate, and its reference error rate can instead be managed directly, since obviously they do not require any preliminary scalar reduction process.

Managing the Higher Priority Inequality Objectives

A prioritized list of the various scalar inequality objectives, to be concurrently progressively achieved, is suitably established in a descending priority order.

Then, by starting to consider the highest priority one, we have that the linear manifold of the system velocity vector y (i.e., the arm joints velocity vector $\dot{q}$ stacked with vector v of the vehicle linear and angular velocities), capable of driving toward its achievement, results at each time instant as the set of solution of the following minimization problem with scalar argument, with row vector $G_1 \doteq \alpha_1 n_1^T H_1$ and scalar α_1 the same activation function embedded within the reference error rate $\dot{\sigma}_1$

$$S_1 \doteq \left\{ \underset{y}{\operatorname{argmin}} \left\| \dot{\sigma}_1 - G_1 y \right\|^2 \right\} \Leftrightarrow$$

$$y = G_1^\# \dot{\sigma}_1 + (I - G_1^\# G_1) z_1 \doteq \rho_1 + Q_1 z_1; \quad \forall z_1$$

(1)

The above minimization, whose solution manifold appears at the right (also expressed in a concise notation with an obvious correspondence of terms) parameterized by the arbitrary vector z_1 , has to be assumed executed without extracting the common factor α_1 , that is, by evaluating the pseudo-inverse matrix $G_1^\#$ via the regularized form

$$G_1^\# = \left(\alpha_1^2 n_1^T H_1 H_1^T n_1 + p_1 \right)^{-1} \alpha_1 H_1^T n_1$$

with p_1 , a suitably chosen bell-shaped, finite support and centered on zero, regularizing function of the norm of row vector G_1 .

In the above solution manifold, when $\alpha_1 = 1$ (i.e., when the first inequality is still far to be achieved), the second arbitrary term $Q_1 z_1$ is orthogonal to the first, thus having no influence on the generated $\dot{\sigma}_1 = \dot{\bar{\sigma}}_1$ and consequently suitable to be used for also progressing toward the achievement of other lower priority objectives, without perturbing the current progressive achievement of the first one. Note however that, since in this condition the span of the second term results one dimension less than the whole system velocity space $y \in R^N$, this implies that the lower priority objectives can be progressed by only acting within a one-dimension reduced system velocity subspace.

When $\alpha_1 = 0$ (i.e., when the first inequality is achieved) since $G_1^\# = 0$ (as granted by the regularization) and consequently $y = z_1$, the lower priority objectives can instead be progressed by now exploiting the whole system velocity space.

When instead α_1 is within its transition zone $0 < \alpha_1 < 1$ (i.e., when the first inequality is near to be achieved), since the two terms of the solution manifold now become only approximately orthogonal, this can make the usage of the second term for managing lower priority tasks, possibly counteracting the first, currently acting in favor of the highest priority one, but in any case without any possibility of making the primary error variable σ_1 getting out of its enlarged boundaries (i.e., the ones inclusive of the transition zone), thus meaning that once the primary variable σ_1

has entered within such larger boundaries, it will definitely never get out of them.

With the above considerations in mind, managing the remaining priority-descending sequence of inequality objectives can then be done by applying the same philosophy to each of them and within the mobility space left free by its preceding ones, that is, as the result of the following sequence of nested minimization problems:

$$S_i \doteq \left\{ \underset{y \in S_{i-1}}{\operatorname{argmin}} \left\| \dot{\bar{\sigma}}_i - G_i y \right\|^2 \right\}; \quad i = 1, 2, \dots, k$$

with $G_i \doteq \alpha_i n_i^T H_i$ and with k indexing the lowest priority inequality objective and where the highest priority objective has been also included for the sake of completeness (upon letting $S_0 = R^N$). In this way the procedure guarantees the concurrent prioritized convergence (actually occurring as a sort of “domino effect” scattering along the prioritized objective list) toward the ultimate fulfillment of all inequality objectives, each one within its enlarged bounds at worse and with no possibility of getting out of them, once reached.

Further, a simple algebra allows translating the above sequence of k nested minimizations into the following algorithmic structure, with initialization $\rho_0 = 0$; $Q_0 = I$ (see Casalino et al. 2012a, b for more details):

$$\hat{G}_1 \doteq G_1 Q_i$$

$$T_i = \left(I - Q_{i-1} G_i^\# G_i \right)$$

$$\rho_i = T_i \rho_{i-1} + Q_{i-1} G_i^\# \dot{\bar{\sigma}}_i$$

$$Q_i = Q_{i-1} \left(I - G_i^\# G_i \right)$$

ending with the last k -th iteration with the solution manifold

$$y = \rho_k + Q_k z_k; \quad \forall z_k$$

where the residual arbitrariness space $Q_k z_k$ has to be then used for managing the remaining equality objectives, as hereafter indicated.

Managing the Lower Priority Equality Objectives and Subsystem Motion Priorities

For managing the lower priority equality objectives when these require the zeroing of their associated error σ_i (as, for instance, for the end-effector sample reaching task), the following sequence of nested minimization problems has to be instead considered (with initialization ρ_k ; Q_k):

$$S_i \doteq \left\{ \operatorname{argmin}_{y \in S_{i-1}} \|\dot{\hat{s}}_i - H_i y\|^2 \right\}; \quad i = (k+1), \dots, m$$

with m indexing the last priority equality objective and where the whole reference error vector rates $\dot{\hat{s}}_i$ and associated whole error vectors $\hat{s}_i$ have now to be used, since for $\alpha_i \equiv 1$ (as it is for any equality objective) the otherwise needed evaluation of unit vectors n_i (which become ill defined for the relevant error σ_i approaching zero) would most probably provoke unwanted chattering phenomena around $\sigma_i = 0$, while instead the above avoids such risk (since $\dot{\hat{s}}_i$ and $\hat{s}_i$ can be evaluated without requiring n_i), even if at the cost of requiring, for each equality objective, three degrees of mobility instead of a sole one, as it instead is for each inequality objectives. However, note how the algorithmic translation of the above procedure remains structurally the same as the one for the inequality objectives (obviously with the substitutions $\dot{\hat{s}}_i \rightarrow \dot{\hat{\sigma}}_i$, $H_i \rightarrow G_i$, and with initialization ρ_k , Q_k), thus ending in correspondence of the m -th last equality objective with the solution manifold

$$y = \rho_m + Q_m z_m; \quad \forall z_m$$

where the still possibly existing residual arbitrariness space $Q_m z_m$ can be further used for assigning motion priorities between the arm and the vehicle, for instance, via the following additional least-priority ending task

$$y = \operatorname{argmin}_{y \in S_m} \|v\|^2 = \rho_{m+1}$$

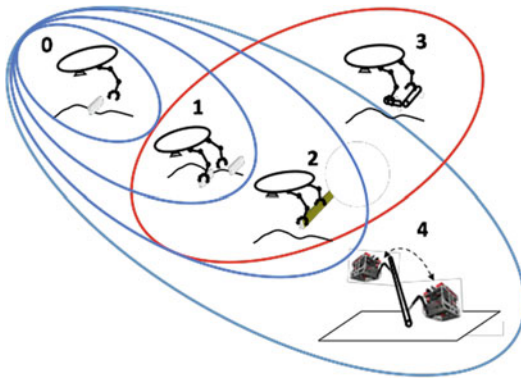
whose solution ρ_{m+1} (with no more arbitrariness required) finally assures (while respecting all previous priorities) a motion minimality of the vehicle, thus implicitly assigning to the arm a greater mobility, which in turn allows the exploitation of its generally higher motion precision, especially during the ultimate convergence toward the final grasping.

Implementations

The recently realized TRIDENT system of Fig. 1, embedding the above introduced task-priority-based control architecture, has been operating at sea in 2012 (Port Soller Harbor, Mallorca, Spain). A detailed presentation of the preliminary performed simulations, then followed by pool experiments, and finally followed by field trials executed within a true underwater sea environment can be found in Simetti et al. (2013). The related EU-funded TRIDENT project (Sanz et al. 2012) is the first one where agile manipulation could be effectively achieved by part of an underwater floating manipulator, not only as the consequence of the comparable masses and inertia exhibited by the vehicle and arm, but mainly due to the adopted unified task-priority-based control framework. Capabilities for autonomous underwater floating manipulation were however already achieved for the first time in 2009 at the University of Hawaii, within the SAUVIM project (Yuh et al. 1998; Marani et al. 2009, 2014) even if without effective agility (the related system was in fact a 6-t vehicle endowed with a less than 35 kg arm).

Future Directions

The presented task-priority-based KCL structure is invariant with the addition, deletion, and substitution (even on-the-fly) of the various objectives, as well as invariant to changes in their priority ordering, thus constituting an invariant core potentially capable of supporting intervention tasks beyond the sole sample collection ones. On this basis, more complex systems and operational



Advanced Manipulation for Underwater Sampling, Fig. 3 A sketch of the foreseen roadmap for future development of marine intervention robotics

cases, such as, for instance, multi-arm systems and/or even cooperating ones, can be foreseen to be developed along the lines established by the roadmap of Fig. 3 (with case 0 the current development state).

The future availability of agile floating single-arm or multi-arm manipulators, also implementing cooperative interventions in force of a unified control and coordination structure (to this aim purposely extended), might in fact pave the way toward the realization of underwater hard-work robotized places, where different intervention agents might individually or cooperatively perform different object manipulation and transportation activities, also including assembly ones, thus far beyond the here considered case of sample collection. Such scenarios deserve the attention not only of the science community when needing to execute underwater works (excavation, coring, instrument handling, etc., other than sample collection) at increasing depths but obviously also those of the offshore industry.

Moreover, by exploiting the current and future developments on underwater exploration and survey mission performed by normal AUVs (i.e., nonmanipulative), a possible work scenario might also include the presence of these last, for accomplishing different service activities supporting the intervention ones, for instance, relays with the surface, then informative activities

(for instance, the delivery of the area model built during a previous survey phase or the delivery of the intervention mission, both downloaded when in surface and then transferred to the intervention agents upon docking), or even when hovering on the work area (for instance, close to a well-recognized feature) behaving as a local reference system for the self-localization of the operative agents via twin USBL devices.

Cross-References

- Control of Networks of Underwater Vehicles
- Control of Ship Roll Motion
- Dynamic Positioning Control Systems for Ships and Underwater Vehicles
- Mathematical Models of Marine Vehicle-Manipulator Systems
- Mathematical Models of Ships and Underwater Vehicles
- Motion Planning for Marine Control Systems
- Redundant Robots
- Robot Grasp Control
- Robot Teleoperation
- Underactuated Marine Control Systems
- Underactuated Robots

Bibliography

- Antonelli G (2006) Underwater robotics. Springer tracts in advanced robotics. Springer, New York
- Casalino G (2011) Trident overall system modeling, including all needed variables for reactive coordination. Technical report ISME-2011. Available at <http://www.grasal.dist.unige.it/files/89>
- Casalino G, Zereik E, Simetti E, Torelli S, Sperindè A, Turetta A (2012a) Agility for uunderwater floating manipulation task and subsystem priority based control strategies. In: International conference on intelligent robots and systems (IROS 2012), Vilamoura-Algarve
- Casalino G, Zereik E, Simetti E, Torelli S, Sperindè A, Turetta A (2012b) A task and subsystem priority based-Ccontrol strategy for underwater floating manipulators. In: IFAC workshop on navigation, guidance and control of underwater vehicles (NGCUV 2012), Porto
- Marani G, Choi SK, Yuh J (2009) Underwater autonomous manipulation for intervention missions AUVs. Ocean Eng 36(1):15–23

- Marani G, Yuh J (2014) Introduction to autonomous manipulation - case study with an underwater robot, SAUVIM. Springer Tracts in Advanced Robotics 102, Springer, pp. 1-156
- Nakamura Y (1991) Advanced robotics: redundancy and optimization. Addison Wesley, Reading
- Sanz P, Ridao P, Oliver G, Casalino G, Insurralde C, Silvestre C, Melchiorri M, Turetta A (2012) TRIDENT: recent improvements about autonomous underwater intervention missions. In: IFAC workshop on navigation, guidance and control of underwater vehicles (NGCUV 2012), Porto
- Simetti E, Casalino G, Torelli S, Sperinde A, Turetta A (2013) Experimental results on task priority and dynamic programming based approach to underwater floating manipulation. In: OCEANS 2013, Bergen, June 2013
- Yoshikawa T (1985) Manipulability of robotic mechanisms. *Int J Robot Res* 4(1):3-9. 1998
- Yuh J, Cho SK, Ikehara C, Kim GH, McMurty G, Ghasemi-Nejhad M, Sarkar N, Sugihara K (1998) Design of a semi-autonomous underwater vehicle for intervention missions (SAUVIM). In: Proceedings of the 1998 international symposium on underwater technology, Tokyo, Apr 1998

Air Traffic Management Modernization: Promise and Challenges

Christine Haissig

Honeywell International Inc., Minneapolis, MN, USA

Abstract

This entry provides a broad overview of how air traffic for commercial air travel is scheduled and managed throughout the world. The major causes of delays and congestion are described, which include tight scheduling, safety restrictions, infrastructure limitations, and major disturbances. The technical and financial challenges to air traffic management are outlined, along with some of the promising developments for future modernization.

Keywords

Air traffic management · Air traffic control · Airport capacity · Airspace management · Flight safety

Synonyms

[ATM Modernization](#)

Introduction: How Does Air Traffic Management Work?

This entry focuses on air traffic management for commercial air travel, the passenger- and cargo-carrying operations with which most of us are familiar. This is the air travel with a pressing need for modernization to address current and future congestion. Passenger and cargo traffic is projected to double over the next 20 years, with growth rates of 3–4 % annually in developed markets such as the USA and Europe and growth rates of 6 % and more in developing markets such as Asia Pacific and the Middle East.

In most of the world, air travel is a distributed, market-driven system. Airlines schedule flights based on when people want to fly and when it is optimal to transport cargo. Most passenger flights are scheduled during the day; most package carrier flights are overnight. Some airports limit how many flights can be scheduled by having a slot system, others do not. This decentralized schedule of flights to and from airports around the world is controlled by a network of air navigation service providers (ANSPs) staffed with air traffic controllers, who ensure that aircraft are separated safely.

The International Civil Aviation Organization (ICAO) has divided the world's airspace into flight information regions. Each region has a country that controls the airspace, and the ANSP for each country can be a government department, state-owned company, or private organization. For example, in the United States, the ANSP is the Federal Aviation Administration (FAA), which is a government department. The Canadian

ANSP is NAV CANADA, which is a private company.

Each country is different in terms of the services provided by the ANSP, how the ANSP operates, and the tools available to the controllers. In the USA and Europe, the airspace is divided into sectors and areas around airports. An air traffic control center is responsible for traffic flow within its sector and rules and procedures are in place to cover transfer of control between sectors. The areas around busy airports are usually handled by a terminal radar approach control. The air traffic control tower personnel handle departing aircraft, landing aircraft, and the movement of aircraft on the airport surface.

Air traffic controllers in developed air travel markets like the USA and Europe have tools that help them with the business of controlling and separating aircraft. Tower controllers operating at airports can see aircraft directly through windows or on computer screens through surveillance technology such as radar and Automatic Dependent Surveillance-Broadcast (ADS-B). Tower controllers may have additional tools to help detect and prevent potential collisions on the airport surface. En route controllers can see aircraft on computer screens and may have additional tools to help detect potential losses of separation between aircraft. Controllers can communicate with aircraft via radio and some have datalink communication available such as Controller-Pilot Datalink Communications (CPDLC).

Flight crews have tools to help with navigating and flying the airplane. Autopilots and autothrottles off-load the pilot from having to continuously control the aircraft; instead the pilot can specify the speed, altitude, and heading and the autopilot and autothrottle will maintain those commands. Flight management systems (FMS) assist in flight planning in addition to providing lateral and vertical control of the airplane. Many aircraft have special safety systems such as the Traffic Alert and Collision Avoidance System, which alerts the flight crew to potential collisions with other airborne aircraft, and the Terrain Avoidance Warning Systems (TAWS), which alert the flight crew to potential flight into terrain.

Causes of Congestion and Delays

Congestion and delays are caused by multiple reasons. These include tight scheduling, safety limitations on how quickly aircraft can take off and land and how closely they can fly, infrastructure limitations such as the number of runways at an airport and the airway structure, and disturbances such as weather and unscheduled maintenance.

Tight Scheduling

Tight scheduling is a major contributor to congestion and delays. The hub and spoke system that many major airlines operate with to minimize connection times means that aircraft arrive and depart in multiple banks during the day. During the arrival and departure banks, airports are very busy. As mentioned previously, passengers have preferred times to travel, which also increase demand at certain times. At airports that do not limit flight schedules by using slot scheduling, the number of flights scheduled can actually exceed the departure and arrival capacity of the airport even in best-case conditions. One of the reasons that airlines are asked to report on-time statistics is to make the published airline schedules more reflective of the average time from departure to arrival, not the best-case time.

Aircraft themselves are also tightly scheduled. Aircraft are an expensive capital asset. Since customers are very sensitive to ticket prices, airlines need to have their aircraft flying as many hours as possible per day. Airlines also limit the number of spare aircraft and flight crews available to fill in when operations are disrupted to control costs.

Safety Restrictions

Safety restrictions contribute to congestion. There is a limit to how quickly aircraft can take off from and land on a runway. Sometimes runways are used for both departing and arriving aircraft; at other times a runway may be used for departures only or arrivals only. Either way, the rule that controllers follow for safety is that only one aircraft can occupy the runway at one time. Thus, a landing aircraft must turn off of the runway before another aircraft can take off

or land. This limitation and other limitations like the ability of controllers to manage the arrival and departure aircraft propagate backwards from the airport. Aircraft need to be spaced in an orderly flow and separated no closer than what can be supported by airport arrival rates. The backward propagation can go all the way to the departure airports and cause aircraft to be held on the ground as a means to regulate the traffic flow into a congested airport or through a congested air traffic sector.

There is a limit on how close aircraft can fly. Aircraft produce a wake that can be dangerous for other aircraft that are following too closely behind. Pilots are aware of this limitation and space safely when doing visual separation. Rules that controllers apply for separation take into account wake turbulence limitations, surveillance limitations, and limitations on how well aircraft can navigate and conform to the required speed, altitude, and heading.

The human is a safety limitation. Controllers and pilots are human. Being human, they have excellent reasoning capability. However, they are limited as to the number of tasks they can perform and are subject to fatigue. The rules and procedures in place to manage and fly aircraft take into account human limitations.

Infrastructure Limitations

Infrastructure limitations contribute to congestion and delays. Airport capacity is one infrastructure limitation. The number of runways combined with the available aircraft gates and capacity to process passengers through the terminal limit the airport capacity.

The airspace itself is a limitation. The airspace where controllers provide separation services is divided into an orderly structure of airways. The airways are like one-way, one-lane roads in the sky. They are stacked at different altitudes, which are usually separated by either 1,000 ft. or 2,000 ft. The width of the airways depends on how well aircraft can navigate. In the US domestic airspace where there are regular navigation aids and direct surveillance of aircraft, the airways have a plus or minus 4 NM width. Over the ocean, airways may need to be separated

laterally by as much as 120 NM since there are fewer navigation aids and aircraft are not under direct control but separated procedurally. The limited number of airways that the airspace can support limits available capacity.

The airways themselves have capacity limitations just as traditional roads do. There are special challenges for airways since aircraft need a minimum separation distance, aircraft cannot slow down to a stop, and airways do not allow passing. So, although it may look like there is a lot of space in which aircraft can fly, there are actually a limited number of routes between a city pair or over oceanic airspace.

The radio that is used for pilots and controllers to communicate is another infrastructure limitation. At busy airports, there is significant radio congestion and pilots may need to wait to get an instruction or response from a controller.

Disturbances

Weather is a significant disturbance in air traffic management. Weather acts negatively in many ways. Wet or icy pavement affects the braking ability of aircraft so they cannot vacate a runway as quickly as in dry conditions. Low cloud ceilings mean that all approaches must be instrument approaches rather than visual approaches, which also reduces runway arrival rates. Snow must be cleared from runways, closing them for some period of time. High winds can mean that certain approaches cannot be used because they are not safe. In extreme weather, an airport may need to close. Weather can block certain airways from use, requiring rerouting of aircraft. Rerouting increases demand on nearby airways, which may or may not have the required additional capacity, so the rerouting cascades on both sides of the weather.

Why Is Air Traffic Management Modernization So Hard?

Air traffic management modernization is difficult for financial and technical reasons. The air traffic management system operates around the clock. It cannot be taken down for a significant period of

time without a major effect on commerce and the economy.

Financing is a significant challenge for air traffic management modernization. Governments worldwide are facing budgetary challenges and improvements to air travel are one of many competing financial interests. Local airport authorities have similar challenges in raising money for airport improvements. Airlines have competitive limitations on how much ticket prices can rise and therefore need to see a payback on investment in aircraft upgrades that can be as short as 2 years.

Another financial challenge is that the entity that needs to pay for the majority of an improvement may not be the entity that gets the majority of the benefit, at least near term. One example of this is the installation of ADS-B transmitters on aircraft. Buying and installing an ADS-B transmitter costs the aircraft owner money. It benefits the ANSPs, who can receive the transmissions and have them augment or replace expensive radar surveillance, but only if a large number of aircraft are equipped. Eventually the ANSP benefit will be seen by the aircraft operator through lower operating costs but it takes time. This is one reason that ADS-B transmitter equipage was mandated in the USA, Europe, and other parts of the world rather than letting market forces drive equipage.

All entities, whether governmental or private, need some sort of business case to justify investment, where it can be shown that the benefit of the improvement outweighs the cost. The same system complexity that makes congestion and delays in one region propagate throughout the system makes it a challenge to accurately estimate benefits. It is complicated to understand if an improvement in one part of the system will really help or just shift where the congestion points are. Decisions need to be made on what improvements are the best to invest in. For government entities, societal benefits can be as important as financial payback, and someone needs to decide whose interests are more important. For example, the people living around an airport might want longer arrival paths at night to minimize noise while air travelers and the airline want the airline to fly the most direct route into an airport. A combination

of subject matter expertise and simulation can provide a starting point to estimate benefit, but often only operational deployment will provide realistic estimates.

It is a long process to develop new technologies and operational procedures even when the benefit is clear and financing is available. The typical development steps include describing the operational concept; developing new controllers procedures, pilot procedures, or phraseology if needed; performing a safety and performance analysis to determine high level requirements; performing simulations that at some point may include controllers or pilots; designing and building equipment that can include software, hardware, or both; and field testing or flight testing the new equipment. Typically, new ground tools are field tested in a shadow mode, where controllers can use the tool in a mock situation driven by real data before the tool is made fully operational. Flight testing is performed on aircraft that are flying with experimental certificates so that equipment can be tested and demonstrated prior to formal certification.

Avionics need to be certified before operational use to meet the rules established to ensure that a high safety standard is applied to air travel. To support certification, standards are developed. Frequently the standards are developed through international cooperation and through consensus decision-making that includes many different organizations such as ANSPs, airlines, aircraft manufacturers, avionics suppliers, pilot associations, controller associations, and more. This is a slow process but an important one, since it reduces development risk for avionics suppliers and helps ensure that equipment can be used worldwide.

Once new avionics or ground tools are available, it takes time for them to be deployed. For example, aircraft fleets are upgraded as aircraft come in for major maintenance rather than pulling them out of scheduled service. Flight crews need to be trained on new equipment before it can be used, and training takes time. Ground tools are typically deployed site by site, and the controllers also require training on new equipment and new procedures.

Promise for the Future

Despite the challenges and complexity of air traffic management, there is a path forward for significant improvement in both developed and developing air travel markets. Developing air travel markets in countries like China and India can improve air traffic management using procedures, tools, and technology that is already used in developed markets such as the USA and Europe. Emerging markets like China are willing to make significant investments in improving air traffic management by building new airports, expanding existing airports, changing controller procedures, and investing in controller tools. In developed markets, new procedures, tools, and technologies will need to be implemented. In some regions, mandates and financial incentives may play a part in enabling infrastructure and equipment changes that are not driven by the marketplace.

The USA and Europe are both supporting significant research, development, and implementation programs to support air traffic management modernization. In the USA, the FAA has a program known as NextGen, the Next Generation Air Transportation System. In Europe, the European Commission oversees a program known as SESAR, the Single European Sky Air Traffic Management Research, which is a joint effort between the European Union, EUROCONTROL, and industry partners. Both programs have substantial support and financing. Each program has organized its efforts differently but there are many similarities in the operational objectives and improvements being developed.

Airport capacity problems are being addressed in multiple ways. Controllers are being provided with advanced surface movement guidance and control systems that combine radar surveillance, ADS-B surveillance, and sensors installed at the airport with valued-added tools to assist with traffic control and alert controllers to potential collisions. Datalink communications between controllers and pilots will reduce radio-frequency congestion, reduce communication errors, and enable more complex communication. The USA and Europe have plans to develop a modernized

datalink communication infrastructure between controllers and pilots that would include information like departure clearances and the taxiway route clearance. Aircraft on arrival to an airport will be controlled more precisely by equipping aircraft with capabilities such as the ability to fly to a required time of arrival and the ability to space with respect to another aircraft.

Domestic airspace congestion is being addressed in Europe by moving towards a single European sky where the ANSPs for the individual nations coordinate activities and airspace is structured not as 27 national regions but operated as larger blocks. Similar efforts are undergoing in the USA to improve the cooperation and coordination between the individual airspace sectors. In some countries, large blocks of airspace are reserved for special use by the military. In those countries, efforts are in place to have dynamic special use airspace that is reserved on an as-needed basis but otherwise available for civil use.

Oceanic airspace congestion is being addressed by leveraging the improved navigation performance of aircraft. Some route structures are available only to aircraft that can flight to a required navigation performance. These route structures have less required lateral separation, and thus more routes can be flown in the same airspace. Pilot tools that leverage ADS-B are allowing aircraft to make flight level changes with reduced separation and in the future are expected to allow pilots to do additional maneuvering that is restricted today, such as passing slower aircraft.

Weather cannot be controlled but efforts are underway to do better prediction and provide more accurate and timely information to pilots, controllers, and aircraft dispatchers at airlines. On-board radars that pilots use to divert around weather are adding more sophisticated processing algorithms to better differentiate hazardous weather. Future flight management systems will have the capability to include additional weather information. Datalinks between the air and the ground or between aircraft may be updated to include information from the on-board radar systems, allowing aircraft to act as local weather

sensors. Improved weather information for pilots, controllers, and dispatchers improves flight planning and minimizes the necessary size of deviations around hazardous weather while retaining safety.

Weather is also addressed by providing aircraft and airports with equipment to improve airport access in reduced visibility. Ground-based augmentation systems installed at airports provide aircraft with the capability to do precision-based navigation for approaches to airports with low weather ceilings. Other technologies like enhanced vision and synthetic vision, which can be part of a combined vision system, provide the capability to land in poor visibility.

Summary

Air traffic management is a complex and interesting problem. The expected increase in air travel worldwide is driving a need for improvements to the existing system so that more passengers can be handled while at the same time reducing congestion and delays. Significant research and development efforts are underway worldwide to develop safe and effective solutions that include controller tools, pilot tools, aircraft avionics, infrastructure improvements, and new procedures. Despite the technical and financial challenges, many promising technologies and new procedures will be implemented in the near, mid-, and far term to support air traffic management modernization worldwide.

Cross-References

- [Aircraft Flight Control](#)
- [Pilot-Vehicle System Modeling](#)

Bibliography

- Collinson R (2011) Introduction to avionics systems, 3rd edn. Springer, Dordrecht
<http://www.faa.gov/nextgen/>
<http://www.sesarju.eu/>
- Nolan M (2011) Fundamentals of air traffic control, 5th edn. Cengage Learning, Clifton Park

Aircraft Flight Control

Dale Enns

Honeywell International Inc., Minneapolis, MN, USA

Abstract

Aircraft flight control is concerned with using the control surfaces to change aerodynamic moments, to change attitude angles of the aircraft relative to the air flow, and ultimately change the aerodynamic forces to allow the aircraft to achieve the desired maneuver or steady condition. Control laws create the commanded control surface positions based on pilot and sensor inputs. Traditional control laws employ proportional and integral compensation with scheduled gains, limiting elements, and cross feeds between coupled feedback loops. Dynamic inversion is an approach to develop control laws that systematically addresses the equivalent of gain schedules and the multivariable cross feeds, can incorporate constrained optimization for the limiting elements, and maintains the use of proportional and integral compensation to achieve the benefits of feedback.

Keywords

Control allocation · Control surfaces · Dynamic inversion · Proportional and integral control · Rigid body equations of motion · Zero dynamics

Introduction

Flying is made possible by flight control and this applies to birds and the Wright Flyer, as well as modern flight vehicles. In addition to balancing lift and weight forces, successful flight also requires a balance of moments or torques about the mass center. Control is a means to adjust these moments to stay in equilibrium and to perform maneuvers. While birds use their feathers and

the Wright Flyer warped its wings, modern flight vehicles utilize hinged control surfaces to adjust the moments. The control action can be open or closed loop, where closed loop refers to a feedback loop consisting of sensors, computer, and actuation. A direct connection between the cockpit pilot controls and the control surfaces without a feedback loop is open loop control. The computer in the feedback loop implements a control law (computer program). The development of the control law is discussed in this entry.

Although the following discussion is applicable to a wide range of flight vehicles including gliders, unmanned aerial vehicles, lifting bodies, missiles, rockets, helicopters, and satellites, the focus of this entry will be on fixed wing commercial and military aircraft with human pilots.

Flight

Aircraft are maneuvered by changing the forces acting on the mass center, e.g., a steady level turn requires a steady force towards the direction of turn. The force is the aerodynamic lift force (L) and it is banked or rotated into the direction of the turn. The direction can be adjusted with the bank angle (μ) and for a given airspeed (V) and air density (ρ), the magnitude of the force

can be adjusted with the angle-of-attack (α). This is called bank-to-turn. Aircraft, e.g., missiles, can also skid-to-turn where the aerodynamic side force (Y) is adjusted with the sideslip angle (β) but this entry will focus on bank-to-turn.

Equations of motion (Stevens and Lewis 1992; Enns et al. 1996) can be used to relate the time rates of change of μ , α , and β to roll (p), pitch (q), and yaw (r) rate. See Fig. 4. Approximate relations (for near steady level flight with no wind) are

$$\dot{\mu} = p$$

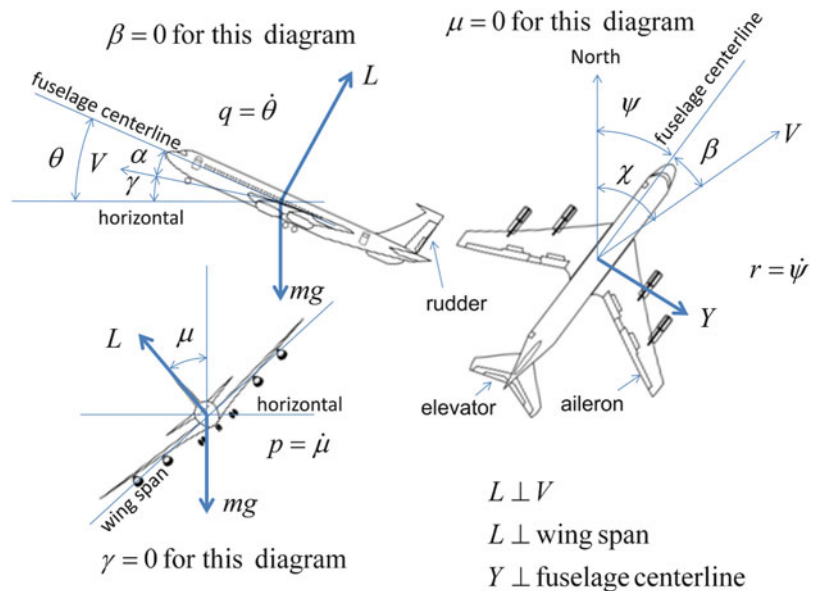
$$\dot{\alpha} = q + \frac{L - mg}{mV}$$

$$\dot{\beta} = -r + \frac{Y}{mV}$$

where m is the aircraft mass, and g is the gravitational acceleration. In straight and level flight conditions $L = mg$ and $Y = 0$ so we think of these equations as kinematic equations where the rates of change of the angles μ , α , and β are the angular velocities p , q , and r .

Three moments called roll, pitch, and yaw for angular motion to move the right wing up or down, nose up or down, and nose right or left, respectively create the angular accelerations to

Aircraft Flight Control,
Fig. 1 Flight control
variables



change p , q , and r , respectively. The equations are Newton's 2nd law for rotational motion. The moments (about the mass center) are dominated by aerodynamic contributions and depend on ρ , V , α , β , p , q , r , and the control surfaces. The control surfaces are aileron (δ_a), elevator (δ_e), and rudder (δ_r) and are arranged to contribute primarily roll, pitch, and yaw moments respectively.

The control surfaces (δ_a , δ_e , δ_r) contribute to angular accelerations which are integrated to obtain the angular rates (p , q , r). The integral of angular rates contributes to the attitude angles (μ , α , β). The direction and magnitude of aerodynamic forces can be adjusted with the attitude angles. The forces create the maneuvers or steady conditions for operation of the aircraft.

Pure Roll Axis Example

Consider just the roll motion. The differential equation (Newton's 2nd law for the roll degree-of-freedom) for this dynamical system is

$$\dot{p} = L_p p + L_{\delta_a} \delta_a$$

where L_p is the stability derivative and L_{δ_a} is the control derivative both of which can be regarded as constants for a given airspeed and air density.

Pitch Axis or Short Period Example

Consider just the pitch and heave motion. The differential equations (Newton's 2nd law for the pitch and heave degrees-of-freedom) for this dynamical system are

$$\dot{q} = M_\alpha \alpha + M_q q + M_{\delta_e} \delta_e$$

$$\dot{\alpha} = Z_\alpha \alpha + q + Z_{\delta_e} \delta_e$$

where M_α , M_q , Z_α are stability derivatives, and M_{δ_e} is the control derivative, all of which can be regarded as constants for a given airspeed and air density.

Although $Z_\alpha < 0$ and $M_q < 0$ are stabilizing, $M_\alpha > 0$ makes the short period motion inherently unstable. In fact, the short period motion of the Wright Flyer was unstable. Some modern aircraft are also unstable.

Lateral-Directional Axes Example

Consider just the roll, yaw, and side motion with four state variables (μ , p , r , β) and two inputs (δ_a , δ_r). We will use the standard state space equations with matrices A , B , C for this example.

The short period equations apply for yaw and side motion (or dutch roll motion) with appropriate replacements, e.g., q with r , α with $-\beta$, M with N . We add the term $V^{-1}g\mu$ to the $\dot{\beta}$ equation. We include the kinematic equation $\dot{\mu} = p$ and add the term $L_\beta \beta$ to the $\dot{p}$ equation. The dutch roll, like the short period, can be unstable if $N_\beta < 0$, e.g., airplanes without a vertical tail.

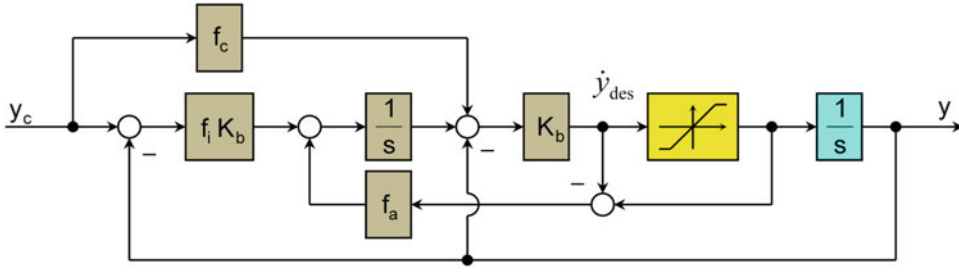
There is coupling between the motions associated with stability derivatives L_r , L_β , N_p and control derivatives L_{δ_r} and N_{δ_a} . This is a fourth order multivariable coupled system where δ_a , δ_r are the inputs and we can consider (p , r) or (μ , β) as the outputs.

Control

The control objectives are to provide stability, disturbance rejection, desensitization, and satisfactory steady state and transient response to commands. Specifications and guidelines for these objectives are assessed quantitatively with frequency, time, and covariance analyses and simulations.

Integrator with P + I Control

The system to be controlled is the integrator for y in Fig. 2 and the output of the integrator (y) is the controlled variable. The proportional gain ($K_b > 0$) is a frequency and sets the bandwidth or crossover frequency of the feedback loop. The value of K_b will be between 1 and 10 rad/s in most aircraft applications. Integral action can be included with the gain, $f_i > 0$ with a value between 0 and 1.5 in most applications. The value of the command gain, $f_c > 0$, is set to achieve a desired closed loop response from the command y_c to the output y . Values of $f_i = 0.25$ and $f_c = 0.5$ are typical. In realistic applications, there is a limit that applies at the input to the integrator. In these cases, we are obligated to include an anti-integral windup gain, $f_a > 0$ (typical value of 2)



Aircraft Flight Control, Fig. 2 Closed loop feedback system and desired dynamics

to prevent continued integration beyond the limit. The input to the limiter ($\dot{y}_{\text{des}}$) is called the desired rate of change of the controlled variable (Enns et al. 1996).

The closed loop transfer function is

$$\frac{y}{y_c} = \frac{K_b(f_c s + f_i K_b)}{s^2 + K_b s + f_i K_b^2}$$

and the pilot produces the commands, (y_c) with cockpit inceptors, e.g., sticks, pedals.

The control system robustness can be adjusted with the choices made for y , K_b , f_i , and f_c .

These desired dynamics are utilized in all of the examples to follow. In the following, we use dynamic inversion (Enns et al. 1996; Wacker et al. 2001) to algebraically manipulate the equations of motion into the equivalent of the integrator for y in Fig. 2.

Pure Roll Motion Example

With algebraic manipulations called dynamic inversion we can use the pure integrator results in the previous section for the pure roll motion example. For the controlled variable $y = p$, given a measurement of the state $x = p$ and values for L_p and $L_{\delta a}$, we simply solve for the input ($u = \delta_a$) that gives the desired rate of change of the output $\dot{y}_{\text{des}} = \dot{p}_{\text{des}}$. The solution is

$$\delta_a = L_{\delta a}^{-1}(\dot{p}_{\text{des}} - L_p p)$$

Since $L_{\delta a}$ and L_p vary with air density and airspeed, we are motivated to schedule these portions of the control law accordingly.

Short Period Example

Similar algebraic manipulations use the general state space notation

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

We want to solve for u to achieve a desired rate of change of y , so we start with

$$\dot{y} = CAx + CBu$$

If we can invert CB , i.e., it is not zero, for the short period case, we solve for u with

$$u = (CB)^{-1}(\dot{y}_{\text{des}} - CAx)$$

Implementation requires a measurement of the state, x and models for the matrices CA and CB .

The closed loop poles include the open loop zeros of the transfer function $\frac{y(s)}{u(s)}$ (zero dynamics) in addition to the roots of the desired dynamics characteristic equation. Closed loop stability requires stable zero dynamics. The zero dynamics have an impact on control system robustness and can influence the precise choice of y .

When $y = q$, the control law includes the following dynamic inversion equation

$$\delta_e = M_{\delta_e}^{-1}(\dot{q}_{\text{des}} - M_q q - M_\alpha \alpha)$$

and the open loop zero is $Z_\alpha - Z_{\delta_e} M_{\delta_e}^{-1} M_\alpha$, which in almost every case of interest is a negative number.

Note that there are no restrictions on the open loop poles. This control law is effective and

practical in stabilization of an aircraft with an open loop unstable short period mode.

Since M_{δ_e} , M_q and M_α vary with air density and airspeed we are motivated to schedule these portions of the control law accordingly.

When $y = \alpha$, the zero dynamics are not suitable as closed loop poles. In this case, the pitch rate controller described above is the inner loop and we apply dynamic inversion a second time as an outer loop (Enns and Keviczky 2006) where we approximate the angle-of-attack dynamics with the simplification that pitch rate has reached steady state, i.e., $\dot{q} = 0$ and regard pitch rate as the input ($u = q$) and angle-of-attack as the controlled variable ($y = \alpha$). The approximate equation of motion is

$$\begin{aligned}\dot{\alpha} &= Z_\alpha \alpha + q - Z_{\delta_e} M_{\delta_e}^{-1} (M_\alpha \alpha + M_q q) \\ &= (Z_\alpha - Z_{\delta_e} M_{\delta_e}^{-1} M_\alpha) \alpha \\ &\quad + (1 - Z_{\delta_e} M_{\delta_e}^{-1} M_q) q\end{aligned}$$

This equation is inverted to give

$$\begin{aligned}q_c &= (1 - Z_{\delta_e} M_{\delta_e}^{-1} M_q)^{-1} \\ &\quad \left[\dot{\alpha}_{\text{des}} - (Z_\alpha - Z_{\delta_e} M_{\delta_e}^{-1} M_\alpha) \alpha \right]\end{aligned}$$

q_c obtained from this equation is passed to the inner loop as a command, i.e., y_c of the inner loop.

Lateral-Directional Example

If we choose the two angular rates as the controlled variables (p, r), then the zero dynamics are favorable. We use the same proportional plus integral desired dynamics in Fig. 2 but there are two signals represented by each wire (one associated with p and the other r).

The same state space equations are used for the dynamic inversion step but now CA and CB are 2×4 and 2×2 matrices, respectively instead of scalars. The superscript in $u = (CB)^{-1} (\dot{y}_{\text{des}} - CAx)$ now means matrix inverse instead of reciprocal. The zero dynamics are assessed with the

transmission zeros of the matrix transfer function $(p, r)/(\delta_a, \delta_r)$.

In the practical case where the aileron and rudder are limited, it is possible to place a higher priority on solving one equation vs. another if the equations are coupled, by proper allocation of the commands to the control surfaces which is called control allocation (Enns 1998). In these cases, we use a constrained optimization approach

$$\min_{u_{\min} \leq u \leq u_{\max}} ||CBu - (\dot{y}_{\text{des}} - CAx)||$$

instead of the matrix inverse followed by a limiter. In cases where there are redundant controls, i.e., the matrix CB has more columns than rows, we introduce a preferred solution, u_p and solve a different constrained optimization problem

$$\min_{CBu + CAx = \dot{y}_{\text{des}}} ||u - u_p||$$

to find the solution that solves the equations that is closest to the preferred solution. We utilize weighted norms to accomplish the desired priority.

An outer loop to control the attitude angles (μ, β) can be obtained with an approach analogous to the one used in the previous section.

Nonlinear Example

Dynamic inversion can be used directly with the nonlinear equations of motion (Enns et al. 1996; Wacker et al. 2001). General equations of motion, e.g., 6 degree-of-freedom rigid body can be expressed with $\dot{x} = f(x, u)$ and the controlled variable is given by $y = h(x)$. With the chain rule of calculus we obtain

$$\dot{y} = \frac{\partial h}{\partial x}(x) f(x, u)$$

and for a given $\dot{y} = \dot{y}_{\text{des}}$ and (measured) x we can solve this equation for u either directly or approximately. In practice, the first order Taylor Series approximation is effective

$$\dot{y} \cong a(x, u_0) + b(x, u_0)(u - u_0)$$

where u_0 is typically the past value of u , in a discrete implementation. As in the previous example, Fig. 2 can be used to obtain $\dot{y}_{des}$. The terms $a(x, u_0) - b(x, u_0)u_0$ and $b(x, u_0)$ are analogous to the terms CAx and the matrix CB , respectively. Control allocation can be utilized in the same way as discussed above. The zero dynamics are evaluated with transmission zeros at the intended operating points. Outer loops can be employed in the same manner as discussed in the previous section.

The control law with this approach utilizes the equations of motion which can include table lookup for aerodynamics, propulsion, mass properties, and reference geometry as appropriate. The raw aircraft data or an approximation to the data takes the place of gain schedules with this approach.

Summary and Future Directions

Flight control is concerned with tracking commands for angular rates. The commands may come directly from the pilot or indirectly from the pilot through an outer loop, where the pilot directly commands the outer loop. Feedback control enables stabilization of aircraft that are inherently unstable and provides disturbance rejection and insensitive closed-loop response in the face of uncertain or varying vehicle dynamics. Proportional and integral control provide these benefits of feedback. The aircraft dynamics are significantly different for low altitude and high speed compared to high altitude and low speed and so portions of the control law are scheduled. Aircraft do exhibit coupling between axes and so multivariable feedback loop approaches are effective. Nonlinearities in the form of limits (noninvertible) and nonlinear expressions, e.g., trigonometric, polynomial, and table look-up (invertible) are present in flight control development. The dynamic inversion approach has been shown to include the traditional feedback control principles, systematically develops the equivalent of the gain schedules, applies to multivariable systems, applies to invertible nonlinearities, and can be

used to avoid issues with noninvertible nonlinearities to the extent it is physically possible.

Future developments will include adaptation, reconfiguration, estimation, and nonlinear analyses. Adaptive control concepts will continue to mature and become integrated with approaches such as dynamic inversion to deal with unstructured or nonparameterized uncertainty or variations in the aircraft dynamics. Parameterized uncertainty will be incorporated with near real time reconfiguration of the aircraft model used as part of the control law, e.g., reallocation of control surfaces after an actuation failure. State variables used as measurements in the control law will be estimated as well as directly measured in nominal and sensor failure cases. Advances in nonlinear dynamical systems analyses will create improved intuition, understanding, and guidelines for control law development.

Cross-References

- [Feedback Linearization of Nonlinear Systems](#)
- [PID Control](#)
- [Satellite Control](#)
- [Tactical Missile Autopilots](#)

Bibliography

- Enns DF (1998) Control allocation approaches. In: Proceedings of the AIAA guidance, navigation, and control conference, Boston
- Enns DF, Keviczky T (2006) Dynamic inversion based flight control for autonomous RMAX helicopter. In: Proceedings of the 2006 American control conference, Minneapolis, 14–16 June 2006
- Enns DF et al (1996) Application of multivariable control theory to aircraft flight control laws, final report: multivariable control design guidelines. Technical report WL-TR-96-3099, Flight Dynamics Directorate, Wright-Patterson Air Force Base, OH 45433-7562, USA
- Stevens BL, Lewis FL (1992) Aircraft control and simulation. Wiley, New York
- Wacker R, Enns DF, Bugajski DJ, Munday S, Merkle S (2001) X-38 application of dynamic inversion flight control. In: Proceedings of the 24th annual AAS guidance and control conference, Breckenridge, 31 Jan–4 Feb 2001

Application of Systems and Control Theory to Quantum Engineering

Naoki Yamamoto

Department of Applied Physics and
Physico-Informatics, Keio University,
Yokohama, Japan

Abstract

This entry is devoted to show that some quantum engineering problems, state transfer, state protection, non-demolition measurement, and back-action-evading measurement, can be formulated and solved within the framework of linear systems and control theory.

Keywords

Quantum information · Linear systems ·
Controllability and observability

Introduction

Systems and control theory has established vast analytic and computational methods for analyzing/synthesizing a system of the form

$$\dot{x} = Ax + Bu, \quad y = Cx + Du. \quad (1)$$

Surprisingly, many important quantum systems, such as optical, superconducting, and atomic systems, can be modeled with linear dynamical equations. In those cases, x denotes the vector of variables (called “observables” in quantum mechanics) such as the position q and the momentum p of a particle; note that these variables are operators, not scalar-valued quantities, which thus do not commute with each other, e.g., $qp - pq = i$. Also, if one aims to control the system or extract information from it, typically an electromagnetic field called the probe field is injected into the system; u denotes the vector of probe variables such as amplitude and phase of the field. y denotes the variables of (reflected or transmitted) output field or the

signal obtained as a result of measurement on the output field. The matrices (A, B, C, D) are determined from the system. The goal of this entry is to show that some quantum engineering problems can be well formulated and solved within the framework of systems and control theory. See Nurdin and Yamamoto (2017) for other applications to quantum engineering.

State Transfer

To realize quantum information technology such as quantum computation and quantum communication, it is very important to devise a scalable hybrid system composed of optical and solid-state systems, to compensate for their advantages and disadvantages; the main advantage of solid-state systems such as superconducting devices is that it is relatively easy to generate a highly non-linear coupling and produce a genuine quantum state, but the disadvantage is that sending those states to other places is hard; on the other hand, optical systems such as an optical fiber network is suited to efficiently sending a quantum state to several sites, but in general, the quantumness of optical states is weak, due to the weak non-linearity of optical systems. A key technique to connect these different systems is state transfer from optics to solid state and vice versa. In fact there have been a number of research studies on this topic, including the analysis of particular systems achieving perfect state transfer.

Yamamoto and James (2014) demonstrated a system theoretic approach for designing an input single-photon state that can be perfectly transferred to a general SISO passive linear quantum system of the form

$$\dot{x} = Ax - C^\dagger u, \quad y = Cx + u,$$

where $A = -i\Omega - C^\dagger C/2$ with Ω a Hermitian matrix and C a complex row vector describing the system-probe coupling. The idea is simple; the perfect input $u(t)$ (more precisely $u(t)$ is the pulse shape of the input single-photon state) should be the one such that $y(t) = 0$ always holds for the duration of the transfer, meaning that the input field is completely absorbed into the

solid-state system with variable $x(t)$, and accordingly, the output field must be a vacuum. Then $y(t) = 0$, ($t \leq t_1$) readily leads to $u^{\text{opt}}(t) = -x(t_1)^\top e^{-A^*(t-t_1)} C^\top$ ($\bullet^\dagger$, $\bullet^\top$, and $\bullet^*$ represent the Hermitian conjugate, transpose, and element-wise complex conjugate, respectively). This special type of system, whose output is always zero, is said to have the *zero dynamics* in the systems and control theory, which are completely characterized by the zeros of the transfer function in the linear case; Yamamoto et al. (2016) used this fact to extend the above result to the multi-input case. Moreover, Nakao and Yamamoto (2017) considered an optimal control problem where an additional control signal is introduced through the A matrix (more precisely through the system Hamiltonian) for reducing the complexity of the desired pulse shape $u^{\text{opt}}(t)$.

State Protection

Once a desirable quantum state is generated in a solid-state system by, e.g., the state transfer method described above, the next task is to manipulate and store that state. This process must be carried out in a subsystem isolated from the probe field as well as the surrounding environment. This subsystem is called a *decoherence-free subsystem (DFS)*.

A contribution of systems and control theory is that it can offer a convenient and general characterization of DFS, particularly in the linear case. The point is that a DFS is a system that is not affected by the probe and environment fields, and also it cannot be monitored from outside; in the language of systems and control, a DFS is a u -uncontrollable and y -unobservable subsystem for the system (1). Based on this fact, Yamamoto (2014) gave a simple necessary and sufficient condition for the general linear quantum system to have a DFS. An example is an atomic ensemble described by the following linear dynamical equation:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -\kappa & ig & 0 \\ ig & -i\delta & i\omega \\ 0 & i\omega^* & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} \sqrt{2\kappa} \\ 0 \\ 0 \end{bmatrix} u, \quad (2)$$

$$y = \sqrt{2\kappa}x_1 + u,$$

where (x_1, x_2, x_3) are the system variables (spin operators) and $(\kappa, g, \delta, \omega)$ are the system parameters. Clearly, x_3 is decoupled from both the input and output fields when $\omega = 0$; thus, it functions as a DFS. This system can be used as a quantum memory as follows. By setting $\omega \neq 0$ and using the technique described in the previous subsection, one can transfer an input photon state to x_3 ; then by setting $\omega = 0$, that state is preserved. If one wants to use the stored state later, then the DFS again couples with the probe field and releases the state. Also, Yamamoto (2014) considered a linear system having a two-dimensional DFS and showed a simple manipulation of the DFS state.

An interesting related problem is how to engineer a system that contains a DFS. *Coherent feedback (CF)* control, the method connecting some quantum subsystems through electromagnetic fields in a feedback way without introducing any measurement process, gives a solution for this purpose. In fact it was demonstrated in Yamamoto (2014) that, actually, the CF method is applied to engineer a linear system having a DFS.

Quantum Non-demolition Measurement

There is a need to precisely determine and control a particular element of x , say x_s , such as the motional variable of a mechanical oscillator and the number of photons in an optical cavity. For this purpose, rich information on x_s should be available via some measurement on the output probe field, while x_s should not be perturbed by the corresponding input probe field; if there exists such a variable x_s , then it is called a *quantum non-demolition (QND)* variable, and the corresponding measurement schematic is called the QND measurement. The system theoretic formulation of a QND variable is that x_s is u -uncontrollable and y -observable. Thanks to this general characterization, it is possible to take several new approaches for analyzing and synthesizing QND variables. For instance, using a special type of CF control, one can create a QND variable which does not exist in the original uncontrolled system (Yamamoto 2014).

Back-Action-Evading Measurement

To detect a very small (i.e., quantum level) signal such as a gravitational wave and a tiny magnetic field, it is important to devise a special type of detector that fully takes into account quantum mechanical property; the *back-action-evading* (BAE) measurement is one such established method. Here we describe the problem for the system (1). This system functions as a sensor for a small signal, in such a way that the signal drives some y -observable elements of x . In the case where the probe field is single mode, the input is represented as $u = (Q, P)$, where Q and P are the amplitude and phase variables of the input field; these are the so-called conjugate variables satisfying the Heisenberg uncertainty relation $\langle \Delta Q^2 \rangle \langle \Delta P^2 \rangle \geq 1$. Also now the corresponding measurement output is given by $y = Cx + Q$. Hence it seems that S/N increases by reducing the magnitude of the noise Q , but this induces an increase of P from the abovementioned Heisenberg uncertainty relation; for this reason, P is called the back-action noise. Consequently, in the usual setup of the detector, the noise level of y is lower bounded by the so-called *standard quantum limit* (SQL).

There have been a number of proposals for the detector configuration that can beat the SQL and achieve better sensitivity of the signal, that is, the aim is to devise a BAE detector such that y is not affected by P . In the language of systems and control theory, this condition is that the transfer function from P to y , $\Xi(s)$, is zero for all s . This is equivalent to the geometric condition that the y -observable subspace is contained in the P -uncontrollable subspace. Using this characterization, (Yamamoto 2014; Yokotera and Yamamoto 2016) gave a new BAE force detector, based on the CF control. Moreover, in Yokotera and Yamamoto (2016), the BAE problem is formulated as the optimization problem $\|\Xi(s)\| \rightarrow \min.$ for synthesizing a detector that beats the SQL.

Cross-References

- [Control of Quantum Systems](#)
- [Quantum Networks](#)

- [Quantum Stochastic Processes and the Modelling of Quantum Noise](#)

References

- Nurdin HI, Yamamoto N (2017) Linear dynamical quantum systems: analysis, synthesis, and control. Springer, Cham
- Yamamoto N, James MR (2014) Zero dynamics principle for perfect quantum memory in linear networks. *New J Phys* 16:073032
- Yamamoto N, Nurdin HI, James MR (2016) Quantum state transfer for multi-input linear quantum systems. In: *Proceedings of 55th IEEE CDC*
- Nakao H, Yamamoto N (2017) Optimal control for perfect state transfer in linear quantum memory. *J Phys B At Mol Opt Phys* 50:065501
- Yamamoto N (2014) Decoherence-free linear quantum subsystems. *IEEE Trans Autom Control* 59-7:1845/1857
- Yamamoto N (2014) Coherent versus measurement feedback: linear systems theory for quantum information. *Phys Rev X* 4:041029
- Yokotera Y, Yamamoto N (2016) Geometric control theory for quantum back-action evasion. *EPJ Quantum Technol* 3:15

Applications of Discrete Event Systems

Spyros Reveliotis

School of Industrial and Systems Engineering,
Georgia Institute of Technology, Atlanta, GA,
USA

Abstract

This entry provides an overview of the problems addressed by DES theory, with an emphasis on their connection to various application contexts. The primary intentions are to reveal the caliber and the strengths of this theory and to direct the interested reader, through the listed citations, to the corresponding literature. The concluding part of the article also identifies some remaining challenges and further opportunities for the area.

Keywords

Discrete event systems · Applications

Introduction

Discrete event system (DES) theory ► [Models for Discrete Event Systems: An Overview](#) Christos Cassandras, (Cassandras and Lafor-tune 2008; Seatzu et al. 2013) emerged in the late 1970s/early 1980s from the effort of the controls community to address the control needs of applications concerning some complex production and service operations, like those taking place in manufacturing and other workflow systems, telecommunication and data processing systems, and transportation systems. These operations were seeking the ability to support higher levels of efficiency and productivity and more demanding notions of quality of product and service. At the same time, the thriving computing technologies of the era, and in particular the emergence of the microprocessor, were cultivating, and to a significant extent supporting, visions of ever-increasing automation and autonomy for the aforementioned operations. The DES community set out to provide a systematic and rigorous understanding of the dynamics that drive the aforementioned operations and their complexity and to develop a control paradigm that would define and enforce the target behaviors for those environments in an effective and robust manner.

In order to address the aforementioned objectives, the controls community had to extend its methodological base, borrowing concepts, models, and tools from other disciplines. Among these disciplines, the following two played a particularly central role in the development of the DES theory: (i) theoretical computer science (TCS) and (ii) operations research (OR). As a new research area, DES thrived on the analytical strength and the synergies that resulted from the rigorous integration of the modeling frameworks that were borrowed from TCS and OR. Furthermore, the DES community substantially extended those borrowed frameworks, bringing in them

many of its control-theoretic perspectives and concepts.

In general, DES-based approaches are characterized by (i) their emphasis on a rigorous and formal representation of the investigated systems and the underlying dynamics; (ii) a double focus on time-related aspects and metrics that define traditional/standard notions of performance for the considered systems but also on a more behaviorally oriented analysis that is necessary for ensuring fundamental notions of “correctness,” “stability,” and “safety” of the system operation, especially in the context of the aspired levels of autonomy; (iii) the interplay between the two lines of analysis mentioned in item (ii) above and the further connection of this analysis to structural attributes of the underlying system; and (iv) an effort to complement the analytical characterizations and developments with design procedures and tools that will provide solutions provably consistent with the posed specifications and effectively implementable within the time and other resource constraints imposed by the “real-time” nature of the target applications.

The rest of this entry overviews the current achievements of DES theory with respect to (w.r.t.) the different classes of problems that have been addressed by it and highlights the potential that is defined by these achievements for a range of motivating applications. On the other hand, the constricted nature of this entry does not allow an expansive treatment of the aforementioned themes. Hence, the provided coverage is further supported and supplemented by an extensive list of references that will connect the interested reader to the relevant literature.

A Tour of DES Problems and Applications

DES-Based Behavioral Modeling, Analysis, and Control

The basic characterization of behavior in the DES theoretic framework is through the various event sequences that can be generated by the underlying system. Collectively, these sequences are known as the (formal) language generated by

the plant system, and the primary intention is to restrict the plant behavior within a subset of the generated event strings. The investigation of this problem is further facilitated by the introduction of certain mechanisms that act as formal representations of the studied systems, in the sense that they generate the same strings of events (i.e., the same formal language). Since these models are concerned with the representation of the event sequences that are generated by DES, and not by the exact timing of these events, they are frequently characterized as *untimed* DES models. In the practical applications of DES theory, the most popular such models are the finite state automaton (FSA) (Hopcroft and Ullman 1979; Cassandras and Lafortune 2008), ► [Supervisory Control of Discrete-Event Systems](#) W. Murray Wonham, ► [Diagnosis of Discrete Event Systems](#) Stéphane Lafortune and the Petri net (PN) (Murata 1989; Cassandras and Lafortune 2008), ► [Modeling, Analysis, and Control with Petri Nets](#) Manuel Silva.

In the context of DES applications, these modeling frameworks have been used to provide succinct characterizations of the underlying event-driven dynamics and to design controllers, in the form of supervisors, that will restrict these dynamics so that they abide to safety, consistency, fairness, and other similar considerations ► [Supervisory Control of Discrete-Event Systems](#) W. Murray Wonham. As a more concrete example, in the context of contemporary manufacturing, DES-based behavioral control – frequently referred to as supervisory control (SC) – has been promoted as a systematic methodology for the synthesis and verification of the control logic that is necessary for the support of the so-called SCADA (supervisory control and data acquisition) function. This control function is typically implemented through the programmable logic controllers (PLCs) that have been employed in contemporary manufacturing shop floors, and DES SC theory can support it (i) by providing more rigor and specificity to the models that are employed for the underlying plant behavior and the imposed specifications and (ii) by offering the ability to synthesize control policies that are provably correct by construction. Some example

works that have pursued the application of DES SC along these lines can be found in Balemi et al. (1993), Brandin (1996), Park et al. (1999), Chandra et al. (2003), Endsley et al. (2006), and Andersson et al. (2010).

On the other hand, the aforementioned activity has also defined a further need for pertinent interfaces that will translate (a) the plant structure and the target behavior to the necessary DES-theoretic models and (b) the obtained policies to PLC executables. This need has led to a line of research, in terms of representational models and computational tools, which is complementary to the core DES developments described in the previous paragraphs. Indicatively we mention the development of GRAFCET (David and Alla 1992) and of the sequential function charts (SFCs) (Lewis 1998) from the earlier times, while some more recent endeavor along these lines is reported in Wightkin et al. (2011) and Alenljung et al. (2012) and the references cited therein.

Besides its employment in the manufacturing domain, DES SC theory has also been considered for the coordination of the communicating processes that take place in various embedded systems (Feng et al. 2007); the systematic validation of the embedded software that is employed in various control applications, ranging from power systems and nuclear plants to aircraft and automotive electronics (Li and Kumar 2012); the synthesis of the control logic in the electronic switches that are utilized in telecom and data networks; and for the modeling, analysis, and control of the operations that take place in health-care systems (Sampath et al. 2008). Wassying et al. (2011) gives a very interesting account of the gains but also the extensive challenges, experienced by a team of researchers who have tried to apply formal methods, similar to those that have been promoted by the behavioral DES theory, to the development and certification of the software that manages some safety critical operations for Canadian nuclear plants.

Apart from control, untimed DES models have also been employed for the diagnosis of critical events, like certain failures, that cannot be observed explicitly, but their occurrence can be inferred from some resultant behavioral

patterns (Sampath et al. 1996) ► [Diagnosis of Discrete Event Systems](#) Stephane Lafortune. More recently, the relevant methodology has been extended with prognostic capability (Kumar and Takai 2010), while an interesting variation of it addresses the “dual” problem that concerns the design of systems where certain events or behavioral patterns must remain undetectable by an external observer who has only partial observation of the system behavior; this last requirement has been formally characterized by the notion of “opacity” in the relevant literature, and it finds application in the design and operation of secure systems (Dubreil et al. 2010; Saboori and Hadjicostis 2012, 2014; Wu and Lafortune 2013) ► [Opacity of Discrete Event Systems](#) Christoforos Hadjicostis.

Dealing with the Underlying Computational Complexity

As revealed from the discussion of the previous paragraphs, many of the applications of DES SC theory concern the integration and coordination of behavior that is generated by a number of interacting components. In these cases, the formal models that are necessary for the description of the underlying plant behavior may grow their size very fast, and the algorithms that are involved in the behavioral analysis and control synthesis may become practically intractable. Nevertheless, the rigorous methodological base that underlies DES theory provides also a framework for addressing these computational challenges in an effective and structured manner.

More specifically, DES SC theory provides conditions under which the control specifications can be decomposable to the constituent plant components while maintaining the integrity and correctness of the overall plant behavior ► [Supervisory Control of Discrete-Event Systems](#) W. Murray Wonham, (2006). The aforementioned works of Brandin (1996) and Endsley et al. (2006) provide some concrete examples for the application of modular control synthesis. But there are also fundamental problems addressed by SC theory and practice that require a holistic view of the underlying plant and its operation, and thus, they are not

amenable to the aforementioned decomposing solutions. DES SC theory can provide effective and tractable solutions for many of these cases as well, by, e.g., (i) helping identify special plant structure, of practical relevance, for which the target supervisors are implementable in a computationally efficient manner, or (ii) developing customized structured approaches that can systematically trade-off the original specifications for computational tractability. Additional substantial leverage in such endeavors is provided by the availability of more than one formal framework for tackling these problems, with complementary modeling and analytical capabilities. In particular, Petri nets can be an especially useful tool in the context of these endeavors, since they (a) provide very compact representations of the underlying plant dynamics, (b) capture effectively the connection of these dynamics to the structural properties of the plant, and also (c) admit analytical techniques of a more algebraic nature that do not require an explicit enumeration of the underlying state space (Murata 1989; Holloway et al. 1997; Cabasino et al. 2013) ► [Modeling, Analysis, and Control with Petri Nets](#) Manuel Silva.

A particular application that has benefited from, and, at the same time, has significantly promoted the capabilities of DES SC theory to deal in an effective and structured manner with the high inherent complexity of the targeted behaviors, is that concerning the deadlock-free operation of many systems where a set of processes that execute concurrently and in a staged manner, are competing, at each of their processing stages, for the allocation of a finite set of reusable resources. In DES theory, this problem is known as the liveness-enforcing supervision of sequential resource allocation systems (RAS) (Reveliotis 2005; Zhou and Fanti 2004), and it underlies the operation of many contemporary applications: from the resource allocation taking place in contemporary manufacturing shop floors (Ezpeleta et al. 1995; Reveliotis and Ferreira 1996; Jeng et al. 2002) to the traveling and/or work-space negotiation in robotic systems (Reveliotis and Roszkowska 2011), automated railway (Giua et al. 2006),

and other guidelpath-based traffic systems (Reveliotis 2000), to Internet-based workflow management systems like those envisioned for e-commerce and certain banking and insurance claim processing applications (Van der Aalst 1997), and to the allocation of the semaphores that control the accessibility of shared resources by concurrently executing threads in parallel computer programs (Liao et al. 2013). A comprehensive and systematic introduction to the DES-based modeling of RAS and the problem of their liveness-enforcing supervision can be found in Reveliotis (2017).

Closing the above discussion on the ability of DES theory to address effectively the complexity that underlies the DES SC problem, we should point out that the same merits of the theory have also enabled the effective management of the complexity that underlies problems related to the performance modeling and control of the various DES applications. We shall return to this capability in the next subsection that discusses the achievements of DES theory in this domain.

DES Performance Control and the Interplay Among Structure, Behavior, and Performance

DES theory is also interested in the performance modeling, analysis, and control of its target applications w.r.t. time-related aspects like throughput, resource utilization, experienced latencies, and congestion patterns. To support this type of analysis, the untimed DES behavioral models are extended to their *timed* versions. This extension takes place by endowing the original untimed models with additional attributes that characterize the experienced delays between the activation of an event and its execution (provided that it is not preempted by some other conflicting event). Timed models are further classified by the extent and the nature of the randomness that is captured by them. A basic such categorization is between deterministic models, where the aforementioned delays take fixed values for every event, and stochastic models which admit more general distributions. From an application standpoint, timed DES models connect DES theory to the multitude of applications that have been

addressed by dynamic programming, stochastic control, and scheduling theory (Bertsekas 1995; Meyn 2008; Pinedo 2002), Control and Optimization of Stochastic Discrete Event Systems Xiren Cao. Also, in their most general definition, stochastic DES models provide the theoretical foundation of discrete event simulation (Banks et al. 2009).

Similar to the case of behavioral DES theory, a practical concern that challenges the application of timed DES models for performance modeling, analysis, and control is the very large size of these models, even for fairly small systems. DES theory has tried to circumvent these computational challenges through the development of methodology that enables the assessment of the system performance, over a set of possible configurations, from the observation of its behavior and the resultant performance at a single configuration. The required observations can be obtained through simulation, and in many cases, they can be collected from a single realization – or sample path – of the observed behavior; but then, the considered methods can also be applied on the actual system, and thus, they become a tool for real-time optimization, adaptation, and learning.

Collectively, the aforementioned methods define a “sensitivity”-based approach to DES performance modeling, analysis, and control (Cassandras and Lafortune 2008), ▶ [Perturbation Analysis of Discrete Event Systems](#) Yorai Wardi. Historically, DES sensitivity analysis originated in the early 1980s in an effort to address the performance analysis and optimization of queueing systems w.r.t. certain structural parameters like the arrival and processing rates (Ho and Cao 1991). But the current theory addresses more general stochastic DES models that bring it closer to broader endeavors to support incremental optimization, approximation, and learning in the context of stochastic optimal control (Cao 2007; Wardi et al. 2018). Some particular applications of DES sensitivity analysis for the performance optimization of production, telecom, and computing systems can be found in Cassandras and Strickland (1988), Cassandras (1994), Panayiotou and Cassandras (1999), Homem-de Mello et al. (1999), Fu and

Xie (2002), Santoso et al. (2005), and Li and Reveliotis (2015).

Another interesting development in time-based DES theory is the theory of $(\max,+)$ algebra (Baccelli et al. 1992; Hardouin et al. 2018). In its practical applications, this theory addresses the timed dynamics of systems that involve the synchronization of a number of concurrently executing processes with no conflicts among them, and it provides important structural results on the factors that determine the behavior of these systems in terms of the occurrence rates of various critical events and the experienced latencies among them. Motivational applications of $(\max,+)$ algebra can be traced in the design and control of telecommunication and data networks, manufacturing, and railway systems, and more recently, the theory has found considerable practical application in the computation of repetitive/cyclical schedules that seek to optimize the throughput rate of automated robotic cells and of the cluster tools that are used in semiconductor manufacturing (Park et al. 1999; Lee 2008; Kim and Lee 2012).

Both, sensitivity-based methods and the theory of $(\max,+)$ algebra, that were discussed in the previous paragraphs, are enabled by the explicit, formal modeling of the DES structure and behavior in the pursued performance analysis and control. This integrative modeling capability that is supported by DES theory also enables a profound analysis of the impact of the imposed behavioral-control policies upon the system performance and, thus, the pursuance of a more integrative approach to the synthesis of the behavioral and the performance-oriented control policies that are necessary for any particular DES instantiation. This is a rather novel topic in the relevant DES literature, and some recent works in this direction can be found in Cao (2005), Markovski and Su (2013), David-Henriet et al. (2013), Li and Reveliotis (2015), and Li and Reveliotis (2016).

The Roles of Abstraction and Fluidification

The notions of “abstraction” and “fluidification” play a significant role in mastering the complexity that arises in many DES applications. Furthermore, both of these concepts have an important

role in defining the essence and the boundaries of DES-based modeling.

In general systems theory, abstraction can be broadly defined as the effort to develop simplified models for the considered dynamics that retain, however, adequate information to resolve the posed questions in an effective manner. In DES theory, abstraction has been pursued w.r.t. the modeling of, both, the timed and untimed behavior, giving rise to hierarchical structures and models. A theory for hierarchical SC is presented in Wonham (2006), while some applications of hierarchical SC in the manufacturing domain are presented in Hill et al. (2010) and Schmidt (2012). In general, hierarchical SC relies on a “spatial” decomposition that tries to localize/encapsulate the plant behavior into a number of modules that interact through the communication structure that is defined by the hierarchy. On the other hand, when it comes to timed DES behavior and models, a popular approach seeks to define a hierarchical structure for the underlying decision-making process by taking advantage of the different time scales that correspond to the occurrence of the various event types. Some particular works that formalize and systematize this idea in the application context of production systems can be found in Gershwin (1994) and Sethi and Zhang (1994) and the references cited therein.

In fact, the DES models that have been employed in many application areas can be perceived themselves as abstractions of dynamics of a more continuous, time-driven nature, where the underlying plant undergoes some fundamental (possibly structural) transition upon the occurrence of certain events that are defined either endogenously or exogenously w.r.t. these dynamics. The combined consideration of the discrete event dynamics that are generated in the manner described above, with the continuous, time-driven dynamics that characterize the modalities of the underlying plant, has led to the extension of the original DES theory to the so-called hybrid systems theory. Hybrid systems theory is itself very rich, and it is covered in another section of this encyclopedia (see also ► [Discrete Event Systems and Hybrid Systems](#),

Connections Between Alessandro Giua). From an applications standpoint, it increases substantially the relevance of the DES modeling framework and brings this framework to some new and exciting applications. Some of the most prominent such applications concern the coordination of autonomous vehicles and robotic systems, and a nice anthology of works concerning the application of hybrid systems theory in this particular application area can be found in the IEEE Robotics and Automation magazine of September 2011. These works also reveal the strong affinity that exists between hybrid systems theory and the DES modeling paradigm. Along similar lines, hybrid systems theory underlies also the endeavors for the development of the automated highway systems that have been explored for the support of the future urban traffic needs (Horowitz and Varaiya 2000; Fleck et al. 2016). Finally, hybrid systems theory and its DES component have been explored more recently as potential tools for the formal modeling and analysis of the molecular dynamics that are studied by systems biology (Curry 2012).

Fluidification, on the other hand, is the effort to represent as continuous flows, dynamics that are essentially of discrete event type, in order to alleviate the computational challenges that typically result from discreteness and its combinatorial nature. The resulting models serve as approximations of the original dynamics; frequently, they have the formal structure of hybrid systems, and they define a basis for developing “relaxations” for the originally addressed problems. Usually, their justification is of an ad hoc nature, and the quality of the established approximations is empirically assessed on the basis of the delivered results (by comparing these results to some “baseline” performance). There are, however, a number of cases where the relaxed fluid model has been shown to retain important behavioral attributes of its original counterpart (Dai 1995). Furthermore, some recent works have investigated more analytically the impact of the approximation that is introduced by these models on the quality of the delivered results (Wardi and Cassandras 2013), while an additional important extension of the fluid modeling framework for

DES is through the notion of “stochastic flow models (SFM),” which allow the flow rates themselves to be random processes. Some works introducing the SFM framework and providing a first set of results for it can be found in Cassandras et al. (2002) and Sun et al. (2004), while a brief but more tutorial exposition of this framework is provided in Wardi et al. (2018). On the other hand, some works exemplifying the application of fluidification in the DES-theoretic modeling frameworks, and the potential advantages that this approach brings in various application contexts, can be found in Srikant (2004), Meyn (2008), David and Alla (2005), Cassandras and Yao (2013), and Ibrahim and Reveliotis (2019). Finally, the work of Vázquez et al. (2013) provides a nice introduction on the pursuance of the “fluidification” concept in the Petri net modeling framework, while the recent work of Ibrahim and Reveliotis (2018) demonstrates very vividly how the corresponding results enable a synergistic employment of all the different representations of DES behavior that have been discussed in this document, in a totally integrative and seamless manner.

Summary and Future Directions

The discussion of the previous section has revealed the extensive application range and potential of DES theory and its ability to provide structured and rigorous solutions to complex and sometimes ill-defined problems. On the other hand, the same discussion has revealed the challenges that underlie many of the DES applications. The complexity that arises from the intricate and integrating nature of most DES models is perhaps the most prominent of these challenges. This complexity manifests itself in the involved computations but also in the need for further infrastructure, in terms of modeling interfaces and computational tools, that will render DES theory more accessible to the practitioner.

The DES community is aware of this need, and the last few years have seen the development of a number of computational platforms that seek to implement and leverage the existing theory

by connecting it to various application settings; indicatively, we mention DESUMA (Ricker et al. 2006), SUPREMICA (Akeson et al. 2006), and TCT (Feng and Wonham 2006), that support DES behavioral modeling, analysis, and control along the lines of DES SC theory, while the website entitled “The Petri Nets World” has an extensive database of tools that support modeling and analysis through untimed and timed variations of the Petri net model. Model checking tools, like SMV and NuSPIN, which are used for verification purposes, are also important enablers for the practical application of DES theory, and, of course, there are a number of programming languages and platforms, like Arena, AutoMod, Simio, and SimEvents that support discrete event simulation (SimEvents also supports simulation of hybrid systems). However, with the exception of the discrete event simulation software, which is a pretty mature industry, the rest of the aforementioned endeavors currently evolve primarily within the academic and the broader research community. Hence, a remaining challenge for the DES community is the strengthening and expansion of the aforementioned computational platforms to robust and user-friendly computational tools. The availability of such industrial strength computational tools, combined with the development of a body of control engineers well-trained in DES theory, will be catalytic for bringing all the developments that were described in the earlier parts of this document even closer to the industrial practice.

Cross-References

- [Diagnosis of Discrete Event Systems](#)
- [Discrete Event Systems and Hybrid Systems, Connections Between](#)
- [Modeling, Analysis, and Control with Petri Nets](#)
- [Models for Discrete Event Systems: An Overview](#)
- [Opacity of Discrete Event Systems](#)
- [Perturbation Analysis of Discrete Event Systems](#)
- [Supervisory Control of Discrete-Event Systems](#)

Bibliography

- Akesson K, Fabian M, Flordal H, Malik R (2006) SUPREMICA—an integrated environment for verification, synthesis and simulation of discrete event systems. In: Proceedings of the 8th international workshop on discrete event systems. IEEE, pp 384–385
- Alenljung T, Lennartson B, Hosseini MN (2012) Sensor graphs for discrete event modeling applied to formal verification of PLCs. *IEEE Trans Control Syst Technol* 20:1506–1521
- Andersson K, Richardsson J, Lennartson B, Fabian M (2010) Coordination of operations by relation extraction for manufacturing cell controllers. *IEEE Trans Control Syst Technol* 18:414–429
- Baccelli F, Cohen G, Olsder GJ, Quadrat JP (1992) Synchronization and linearity: an algebra for discrete event systems. Wiley, New York
- Balemi S, Hoffmann GJ, Wong-Toi PG, Franklin GJ (1993) Supervisory control of a rapid thermal multiprocessor. *IEEE Trans Autom Control* 38:1040–1059
- Banks J, Carson II JS, Nelson BL, Nicol DM (2009) Discrete-event system simulation, 5th edn. Prentice Hall, Upper Saddle
- Bertsekas DP (1995) Dynamic programming and optimal control, vols 1,2. Athena Scientific, Belmont
- Brandin B (1996) The real-time supervisory control of an experimental manufacturing cell. *IEEE Trans Robot Autom* 12:1–14
- Cabasino MP, Giua A, Seatzu C (2013) Structural analysis of Petri nets. In: Seatzu C, Silva M, van Schuppen JH (eds) Control of discrete-event systems: automata and petri net perspectives. Springer, London, pp 213–233
- Cao X-R (2005) Basic ideas for event-based optimization of Markov systems. *Discrete Event Syst Theory Appl* 15:169–197
- Cao X-R (2007) Stochastic learning and optimization: a sensitivity approach. Springer Science, New York
- Cassandras CG (1994) Perturbation analysis and “rapid learning” in the control of manufacturing systems. In: Leonides CT (ed) Dynamics of discrete event systems, vol 51, pp 243–284. Academic Press
- Cassandras CG, Lafortune S (2008) Introduction to discrete event systems, 2nd ed. Springer, New York
- Cassandras CG, Strickland SG (1988) Perturbation analytic methodologies for design and optimization of communication networks. *IEEE J Sel Areas Commun* 6:158–171
- Cassandras CG, Yao C (2013) Hybrid models for the control and optimization of manufacturing systems. In: Campos J, Seatzu C, Xie X (eds) Formal methods in manufacturing. CRC Press/Taylor and Francis, Boca Raton
- Cassandras CG, Wardi Y, Melamed B, Sun G, Panayiotou CG (2002) Perturbation analysis for on-line control and optimization of stochastic fluid models. *IEEE Trans Autom Control* 47:1234–1248
- Chandra V, Huang Z, Kumar R (2003) Automated control synthesis for an assembly line using discrete event

- system theory. *IEEE Trans Syst Man Cybern Part C* 33:284–289
- Curry JER (2012) Some perspectives and challenges in the (discrete) control of cellular systems. In: *Proc WODES 2012*. IFAC, pp 1–3
- Dai JG (1995) On positive Harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *Ann Appl Probab* 5:49–77
- David R, Alla H (1992) *Petri nets and grafacet: tools for modelling discrete event systems*. Prentice-Hall, Upper Saddle
- David R, Alla H (2005) *Discrete, continuous and hybrid petri nets*. Springer, Berlin
- David-Henriet X, Hardouin L, Raisch J, Cottenceau B (2013) Optimal control for timed event graphs under partial synchronization. In: *52nd IEEE conference on decision and control*. IEEE
- Dubreil J, Darondeau P, Marchand H (2010) Supervisory control for opacity. *IEEE Trans Autom Control* 55:1089–1100
- Endsley EW, Almeida EE, Tilbury DM (2006) Modular finite state machines: development and application to reconfigurable manufacturing cell controller generation. *Control Eng Pract* 14:1127–1142
- Ezpeleta J, Colom JM, Martinez J (1995) A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Trans Robot Autom* 11:173–184
- Feng L, Wonham WM (2006) TCT: a computation tool for supervisory control synthesis. In: *Proceedings of the 8th international workshop on discrete event systems*. IEEE, pp 388–389
- Feng L, Wonham WM, Thiagarajan PS (2007) Designing communicating transaction processes by supervisory control theory. *Formal Meth Syst Des* 30:117–141
- Fleck JL, Cassandras CG, Geng Y (2016) Adaptive quasi-dynamic traffic light control. *IEEE Trans Control Syst Technol* 24:830–842
- Fu M, Xie X (2002) Derivative estimation for buffer capacity of continuous transfer lines subject to operation-dependent failures. *Discrete Event Syst Theory Appl* 12:447–469
- Gershwin SB (1994) *Manufacturing systems engineering*. PTR Prentice Hall, Englewood Cliffs
- Giua A, Fanti MP, Seatzu C (2006) Monitor design for colored Petri nets: an application to deadlock prevention in railway networks. *Control Eng Pract* 10:1231–1247
- Hardouin L, Cottenceau B, Shang Y, Raisch J (2018) Control and state estimation for max-plus linear systems. *NOW Ser Found Trends Syst Control* 6:1–116
- Hill RC, Cury JER, de Queiroz MH, Tilbury DM, Lafor-tune S (2010) Multi-level hierarchical interface-based supervisory control. *Automatica* 46:1152–1164
- Ho YC, Cao X-R (1991) *Perturbation analysis of discrete event systems*. Kluwer Academic Publishers, Boston
- Holloway LE, Krogh BH, Giua A (1997) A survey of Petri net methods for controlled discrete event systems. *JDEDS* 7:151–190
- Homem-de Mello T, Shapiro A, Spearman ML (1999) Finding optimal material release times using simulation-based optimization. *Manag Sci* 45:86–102
- Hopcroft JE, Ullman JD (1979) *Introduction to automata theory, languages and computation*. Addison-Wesley, Reading
- Horowitz R, Varaiya P (2000) Control design of automated highway system. *Proc IEEE* 88:913–925
- Ibrahim M, Reveliotis S (2018) Throughput maximization of complex resource allocation systems through timed-continuous Petri-net modeling. Technical report, School of Industrial & Systems Eng., Georgia Institute of Technology (submitted for publication)
- Ibrahim M, Reveliotis S (2019) Throughput maximization of capacitated re-entrant lines through fluid relaxation. *IEEE Trans Autom Sci Eng* 16:792–810
- Jeng M, Xie X, Peng MY (2002) Process nets with resources for manufacturing modeling and their analysis. *IEEE Trans Robot Autom* 18:875–889
- Kim J-H, Lee T-E (2012) Feedback control design for cluster tools with wafer residency time constraints. In: *IEEE conference on systems, man and cybernetics*. IEEE, pp 3063–3068
- Kumar R, Takai S (2010) Decentralized prognosis of failures in discrete event systems. *IEEE Trans Autom Control* 55:48–59
- Lee T-E (2008) A review of cluster tool scheduling and control for semiconductor manufacturing. In: *Proceedings of 2008 winter simulation conference*. INFORMS, pp 1–6
- Lewis RW (1998) *Programming industrial control systems using IEC 1131-3*. Technical report, The Institution of Electrical Engineers
- Li M, Kumar R (2012) Model-based automatic test generation for simulink/stateflow using extended finite automaton. In: *Proceedings of CASE 2012*. IEEE
- Li R, Reveliotis S (2015) Performance optimization for a class of generalized stochastic Petri nets. *Discrete Event Dyn Syst Theory Appl* 25:387–417
- Li R, Reveliotis S (2016) Designing parsimonious scheduling policies for complex resource allocation systems through concurrency theory. *Discrete Event Dyn Syst Theory Appl* 26:511–537
- Liao H, Wang Y, Cho HK, Stanley J, Kelly T, Lafor-tune S, Mahlke S, Reveliotis S (2013) Concurrency bugs in multithreaded software: modeling and analysis using Petri nets. *Discrete Event Dyn Syst Theory Appl* 23:157–195
- Markovski J, Su R (2013) Towards optimal supervisory controller synthesis of stochastic nondeterministic discrete event systems. In: *52nd IEEE conference on decision and control*. IEEE
- Meyn S (2008) *Control techniques for complex networks*. Cambridge University Press, Cambridge
- Murata T (1989) Petri nets: properties, analysis and applications. *Proc IEEE* 77:541–580
- Panayiotou CG, Cassandras CG (1999) Optimization of kanban-based manufacturing systems. *Automatica* 35:1521–1533
- Park E, Tilbury DM, Khargonekar PP (1999) Modular logic controllers for machining systems: formal representations and performance analysis using Petri nets. *IEEE Trans Robot Autom* 15:1046–1061

- Pinedo M (2002) Scheduling. Prentice Hall, Upper Saddle River
- Reveliotis SA (2000) Conflict resolution in AGV systems. *IEEE Trans* 32(7):647–659
- Reveliotis SA (2005) Real-time management of resource allocation systems: a discrete event systems approach. Springer, New York
- Reveliotis S (2017) Logical control of complex resource allocation systems. *NOW Ser Found Trends Syst Control* 4:1–224
- Reveliotis SA, Ferreira PM (1996) Deadlock avoidance policies for automated manufacturing cells. *IEEE Trans Robot Autom* 12:845–857
- Reveliotis S, Roszkowska E (2011) Conflict resolution in free-ranging multi-vehicle systems: a resource allocation paradigm. *IEEE Trans Robot* 27:283–296
- Ricker L, Lafortune S, Gene S (2006) DESUMA: a tool integrating Giddes and Umdes. In: Proceedings of the 8th international workshop on discrete event systems. IEEE, pp 392–393
- Saboori A, Hadjicostis CN (2012) Opacity-enforcing supervisory strategies via state estimator constructions. *IEEE Trans Autom Control* 57:1155–1165
- Saboori A, Hadjicostis CN (2014) Current-state opacity formulations in probabilistic finite automata. *IEEE Trans Autom Control* 59:120–133
- Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D (1996) Failure diagnosis using Discrete Event models. *IEEE Trans Control Syst Technol* 4:105–124
- Sampath R, Darabi H, Buy U, Liu J (2008) Control reconfiguration of discrete event systems with dynamic control specifications. *IEEE Trans Autom Sci Eng* 5:84–100
- Santoso T, Ahmed S, Goetschalckx M, Shapiro A (2005) A stochastic programming approach for supply chain network design under uncertainty. *Eur J Oper Res* 167:96–115
- Schmidt K (2012) Computation of supervisors for reconfigurable machine tools. In: Proceedings of WODES 2012. IFAC, pp 227–232
- Seatzu C, Silva M, van Schuppen JH (eds) (2013) Control of discrete-event systems: automata and petri net perspectives. Springer, London
- Sethi SP, Zhang Q (1994) Hierarchical decision making in stochastic manufacturing systems. Birkhäuser, Boston
- Srikant R (2004) The mathematics of internet congestion control. Birkhäuser, Boston
- Sun G, Cassandras CG, Panayiotou CG (2004) Perturbation analysis and optimization of stochastic flow networks. *IEEE Trans Autom Control* 49:2113–2128
- Van der Aalst W (1997) Verification of workflow nets. In: Azema P, Balbo G (eds), Lecture notes in computer science, vol 1248, pp 407–426. Springer, Heidelberg
- Vázquez CR, Mahulea C, Júlvez J, Silva M (2013) Introduction to fluid Petri net models. In: Seatzu C, Silva M, van Schuppen JH (eds), Control of discrete-event systems: automata and petri net perspectives. Springer, London, pp 365–386
- Wardi Y, Cassandras CG (2013) Approximate IPA: trading unbiasedness for simplicity. In: 52nd IEEE Conference on decision and control. IEEE
- Wardi Y, Cassandras CG, Cao XR (2018) Perturbation analysis: a framework for data-driven control and optimization of discrete event and hybrid systems. *Annu Rev Control* 45:267–280
- Wassing A, Lawford M, Maibaum T (2011) Software certification experience in the Canadian nuclear industry: lessons for the future. In: EMSOFT' 11
- Wightkin N, Guy U, Darabi H (2011) Formal modeling of sequential function charts with time Petri Nets. *IEEE Trans Control Syst Technol* 19:455–464
- Wonham WM (2006) Supervisory control of discrete event systems. Technical Report ECE 1636F/1637S 2006-07, Electrical & Computer Engineering, University of Toronto
- Wu Y-C, Lafortune S (2013) Comparative analysis of related notions of opacity in centralized and coordinated architectures. *Discrete Event Syst Theory Appl* 23:307–339
- Zhou M, Fanti MP (eds) (2004) Deadlock resolution in computer-integrated systems. Marcel Dekker, Inc., Singapore

Approximate Dynamic Programming (ADP)

Paul J. Werbos

National Science Foundation, Arlington, VA, USA

Abstract

Approximate dynamic programming (ADP or RLADP) includes a wide variety of general methods to solve for optimal decision and control in the face of complexity, nonlinearity, stochasticity, and/or partial observability. This entry first reviews methods and a few key applications across decision and control engineering (e.g., vehicle and logistics control), computer science (e.g., AlphaGo), operations research, and connections to economics, neuropsychology, and animal behavior. Then it summarizes a sixfold mathematical taxonomy of methods in use today, with pointers to the future.

Paul J. Werbos has retired.

Keywords

Bellman · Pontryagin · Adaptive critic ·
HJB · Nonlinear optimal control · HDP ·
DHP · Neurocontrol · TD(λ) · Adaptive
dynamic programming · RLADP ·
Reinforcement learning · Neurodynamic
programming · Missile interception ·
Nonlinear robust control · Deep learning

Introduction**What Is ADP: A General Definition of ADP and Notation**

Logically, ADP is that branch of applied mathematics which develops and studies general purpose methods for optimal decision or control over dynamical systems which may or may not be subject to stochastic disturbance, partial observability, nonlinearity, and complexity, for situations where exact dynamic programming (DP) is too expensive to perform directly. Many ADP methods assume the case of discrete time t , but others address the case of continuous time t . ADP tries to learn or converge to the exact DP solution as accurately and as reliably as possible, but some optimization problems are more difficult than others. In some applications, it is useful enough to start from the best available controller based on more classical methods and then use ADP iteration to improve that controller. In other applications, users of ADP have obtained results so accurate that they can claim to have designed fully robust nonlinear controllers, by solving (accurately enough) the Hamilton-Jacobi-Bellman (HJB) equation for the control task under study.

Let us write the state of a dynamical system at time t as the vector $\underline{x}(t)$, and the vector of observations as $\underline{y}(t)$. (Full observability is just the special case where $\underline{y}(t) = \underline{x}(t)$.) Let us define a *policy* $\underline{\pi}$ as a control rule, function, or procedure used to calculate the vector $\underline{u}(t)$ of controls (physical actions, decisions) by:

$$\underline{u}(t) = \underline{\pi}(\{y(\tau), \tau \leq t\}) \quad (1)$$

The earliest useful ADP methods (White and Sofge 1992) asked the user to supply a control rule $\underline{\pi}(\underline{y}(t), \underline{\alpha})$ or $\underline{\pi}(\underline{R}(t), \underline{\alpha})$, where $\underline{\alpha}$ is the set of parameters or weights in the control rule, and where $\underline{R}(t)$ may either be an estimate of the state $\underline{x}(t)$, or, more generally, a representation of the “belief state” which is defined as the probability density function $\text{Pr}(\underline{x}(t) | \{y(\tau), \tau \leq t\})$. These types of methods may logically be called “adaptive policy methods.” Of course, the user is free to try several candidate control rules. Just as he or she might try different stochastic models in trying to identify an unknown plant and compare to see which policy works best in simulation. Long before the term “neurodynamic programming” was coined, the earliest ADP designs included the use of universal function approximators such as neural networks to try to approximate the best possible function $\underline{\pi}$.

Later work on ADP developed applications in areas such as missile interception (like the work of Balakrishnan included in Lewis and Liu (2013)) and in complex logistics (like the paper by L. Werbos, Kozma et al on Metamodeling) where it works better simply to calculate the optimal action $u(t)$ at each time t , exploiting other aspects of the standard DP tools. In the future, hybrid methods may be developed which modify a global control law on the fly, similar to the moving intercept trick used by McAvoy with great success in model predictive control (MPC) (White and Sofge 1992). These may be called hybrid policy methods.

The earliest ADP methods (White and Sofge 1992) focused entirely on the case of discrete time. In that case, the task of ADP is to find the policy π which maximizes (or minimizes) the expected value of:

$$J_0(x(t)) = \sum_{\tau=t}^T \frac{U(x(\tau), u(\tau))}{(1+r)^{\tau-t}}, \quad (2)$$

where U is a utility function (or cost function), where T is a termination time (which may be infinity), where r is a discount factor (interest rate), and where future values of $\underline{x}$ and $\underline{u}$ are stochastic functions of the policy π .

As with any optimization problem or control design, it is essential for the user to think hard about what he or she really wants and what the ultimate metric of success or performance really is. The interest rate r is just as important as the utility function, in expressing what the ultimate goal of the exercise is. It often makes sense to define U as a measure of economic value added, plus a penalty function to keep the system within a broad range of controllability. With certain plants, like the dynamically unstable SR-71 aircraft, optimal performance will sometimes take advantage of giving the system “permission” to deviate from the nominal control trajectory to some degree. To solve a very tricky nonlinear control problem, it often helps to solve it first with a larger interest rate, in order to find a good starting point, and then “tighten the screws” or “extending the foresight horizon” by dialing r down to zero. Many ADP computer programs use a kind of “discount factor” $\gamma = 1/(1 + r)$, to simply programming, but there is a huge literature on decision analysis and economics on how to understand the interest rate r .

Continuous time versions of ADP have been pioneered by Frank Lewis (Lewis and Liu 2013), with extensions to differential games (such as two-player “adversarial” ADP offering strategies for worst case survival).

Varieties of ADP from Reinforcement Learning to Operations Research

In practice, there probably does not exist an integrated mathematical textbook on ADP which gives both full explanation and access to the full range of practical tools for the field as a whole. The reason for this is that nonlinear dynamic optimization problems abound in a wide variety of fields, from economics and psychology to control engineering and even to physics. The underlying mathematics and tools are universal, but even today it is important to reach across disciplines and across different approaches and terminologies to know what is available.

Prior to the late 1980s, it was well understood that “exact” DP and the obvious approximations

to DP suffer from a severe “curse of dimensionality.” Finding an exact solution to a DP task required calculations which grew exponentially, even for small tasks in the general case. In the special case of a linear plant and a quadratic utility function, the calculations are much easier; that gave rise to the vast field of linear-quadratic optimal control, discussed in other articles of this encyclopedia. Ron Howard developed iterative dynamic programming for systems with a discrete number of states; this was rigorous and universal enough that it was disseminated from operations research to mathematical economics, but it did not help with curse of dimensionality as such. Research in (mainly discrete) Markov decision processes (MDP) and partially observed MDP (POMDP) has also provided useful mathematical foundations for further work.

All of this work was well grounded in the concept of cardinal utility function, defined in the classic book *Theory of Games and Economic Behavior* by von Neumann and Morgenstern. It started from the classic work of Richard Bellman, most notably the Bellman equation, to be presented in the next section. von Neumann’s concept had deep roots in utilitarian economics and in Aristotle’s concept of “telos”; in fact, there are striking parallels between modern discussion of how to formulate a utility function and Aristotle’s discussion of “happiness” in his *Nicomachean Ethics*.

In the same early period, there was substantial independent interest in a special case of ADP called “reinforcement learning” in psychology and in artificial intelligence in computer science. Unfortunately, the same term, “reinforcement learning,” was applied to many other things as well. For example, it was sometimes used as an umbrella term for all types of optimization method, including static optimization, performed in a heuristic manner.

The original concept of “reinforcement learning” came from B.F. Skinner and his many followers, who worked to develop mathematical models to describe animal learning. Instead of a utility function, “ U ,” they developed models of animals as systems which learn to maximize a reinforcement signal, just “ r ” or “ $r(t)$,” represent-

ing the reward or punishment, pain, or pleasure, experienced by the animal. Even to this day, followers of that tradition sometimes use that notation. The work of Klopff in the tradition of Skinner was a major driver of the classic 1983 paper by Barto, Sutton, and Anderson in the IEEE Transactions on Systems, Man, and Cybernetics (SMC). This work implicitly assumed that the utility function U is not known as a function and that we only have access to its value at any time.

Many years ago, a chemical engineer said in a workshop: “I really don’t want to study all these complex mathematical details. What I really want is a black box, which I can hook up to my sensors and my actuators, and to a real time measure of performance which I can provide. I really want a black box which could do all the work of figuring out the relations between sensors and actuators, and how to maximize performance over time.” The task of reinforcement learning was seen as the task of how to design that kind of black box.

The early pioneers of artificial intelligence (AI) were well aware of reinforcement learning. For example, Marvin Minsky (in the book *Computers and Thought*) argued that new methods for reinforcement learning would be the path to true brain-like general artificial intelligence. But no such methods were available. Minsky recognized that all of the early approaches to reinforcement learning suffered from a severe curse of dimensionality and could not begin to handle the kind of complexity which brains learn to handle. AI and computer science mostly gave up on this approach at that time, just as it gave up on neural networks.

In a way, the paralysis ended in 1987, when Sutton read my paper in SMC describing how the curse of dimensionality could be overcome by treating the reinforcement learning problem as a special case of ADP and by using a combination of neural networks, backpropagation, and new approximation methods. That led to many discussions and to an NSF workshop in 1988, which for the first time brought together many of the relevant disciplines and tried to provide an integration of tools and applications. The book from that workshop, *Neural Networks for Control*, edited by Miller, Sutton, and Werbos is still

of current use. However, later workshops went further and deeper; references White and Sofge (1992) and Lewis and Liu (2013), led by the control community, provide the best comprehensive starting point for what is available from the various disciplines.

Within the relevant disciplines, there has been some important fundamental extension since the publication of Lewis and Liu (2013). The most famous extension by far was the triumph of the AlphaGo systems which learned to defeat the best human champion in Go. At the heart of that success was the use of an ADP method in the heuristic dynamic programming (HDP) group of methods, to be discussed in the next section, as tuned and applied by Sutton. However, success in a game of that complexity also depended on combining HDP with complementary techniques, which are actually being assimilated in engineering applications such as self-driving cars, pioneered at the Chinese Academy of Sciences Institute of Automation (CASIA), which is the center of China’s major new push in “the new AI.”

There has also been remarkable progress in ADP within the control field proper, especially in applications and in stability theory. Building on Lewis and Liu (2013), CASIA and Derong Liu have spawned a wide range of new applications. The new book by Liu is probably a good source, but I am not aware of a single good review of all the more recent stability work across the world. Some commentators on “the new AI” have said that China is so far ahead in engineering kinds of applications relevant to the Internet of Things that Western reviews often fail to appreciate just how much they are missing of new technologies. Off-the-shelf software for ADP tends to reflect the limited set of methods best known in computer science, simply because they tend to have more interest than engineers do in open-source software development (and in dissemination which may benefit competitors).

Within the field of operations research, Warren Powell has developed a comprehensive framework and textbook well-informed by discussions across the disciplines and by a wide range of applications from logistics to battery management.

Back in psychology, the original home of reinforcement learning, progress in developing mathematical models powerful enough to be useful in engineering has been limited because of cultural problems between disciplines, similar to what held back reinforcement learning in general before 1987. However, a new path has opened up to connect neuropsychology with ADP which brings us closer to being able to truly understand the wiring and dynamics of the learning systems of the brain (Werbos and Davis 2016).

Mathematical Taxonomy of ADP Methods in General

There are many ways to classify the vast range of ADP tools and applications developed since 1990. Nevertheless, there are few if any ADP systems in use which do not fall into the sixfold classification given in White and Sofge (1992). Here, I will summarize these six types of ADP method and then mention extensions.

Heuristic Dynamic Programming (HDP)

HDP is the most direct method for approximating the Bellman equation, the foundation of exact dynamic programming. There are many legitimate ways to write that equation; one is:

$$J(\underline{x}(t)) = \max_{\underline{u}(t)} (U(\underline{x}(t), \underline{u}(t)) + \langle J(\underline{x}(t+1) / (1+r) \rangle) \quad (3)$$

where angle brackets denote the expectation value. When $\underline{x}$ is governed by a fully observable stochastic process (MDP), and when $r > 0$ or T is finite, it is well-known that a function J which solves this equation will exist. (See White and Sofge (1992) for the simple additional term needed to guarantee the existence of J for $r = 0$ and $T = \infty$, the case which some of us try to do justice to in our own decision-making.) To calculate the optimal policy of action, one tries to find an exact solution for this function J and then choose actions according to the (static)

optimization problem over $\underline{u}$ shown in Eq. 3. Because J may be any nonlinear function, in principle, early users would often try to develop a lookup table approximation to J , the size of which would grow exponentially with the number of possible states in the table.

The key idea in HDP is to model the function J in a parameterized way, just as one models stochastic processes when identifying their dynamics. That idea was developed in some detail in my 1987 paper in SMC, but was also described in many papers before that. The treatment in White and Sofge (1992) describes in general terms how to train or estimate the parameters or weights $\underline{W}$ in any model of J , $\hat{J}(\underline{R}(t), \underline{W})$, supplied by a user. A user who actually understands the dynamics of the plant well enough may choose to try different models of J and see how they perform. But the vast majority of applications today model J by use of a universal approximation function, such as a simple feedforward neural network. Andrew Barron has proven that the required complexity of such an approximation grows far more slowly, as the number of variables grows, than it would when more traditional linear approximation schemes such as lookup tables or Taylor series are used. Ilin, Kozma, and Werbos have shown that approximation is still better when an upgraded type of recurrent neural network is used instead; in the example of generalized maze navigation, a feedforward approximator such as a convolutional neural network simply does not work. In general, unless the model of J is a simple lookup table or linear model, the training requires the use of generalized backpropagation, which provides the required derivatives in exact closed form at minimum computational cost (White and Sofge 1992; Werbos 2005).

In his famous paper on $TD(\lambda)$ in 1987, Sutton proposed a generalization of HDP to include a kind of approximation quality factor λ which could be used to get convergence albeit to a less exact solution. I am not aware of cases where that particular approximation was useful or important, but with very tricky nonlinear decision problems,

it can be helpful to have a graded series of problems to allow convergence to an optimal solution of the problem of interest; see the discussion of “shaping” in White and Sofge (1992). There is a growing literature on “transfer learning” which provides ever more rigorous methods for transferring solutions from simpler problems to more difficult ones, but even a naive reuse of $\hat{J}(\mathbf{R}(t), \mathbf{W})$ and of π from one plant model to another is often a very effective strategy, as Jay Farrell showed long ago in aerospace control examples.

In getting full value from ADP, it helps to have a deep understanding of what the function $\hat{J}(\mathbf{R}(t), \mathbf{W})$ represents, across many disciplines (White and Sofge 1992; Lewis and Liu 2013). More and more, the world has come to call this function the “value function.” (This is something of a misnomer, however, as you will see in the discussion of Dual Heuristic Programming (DHP) below.) It is often denoted as “V” in computer science. It is often called a “Critic” in the engineering literature, following an early seminal paper by Widrow.

In control theory, Eq. 3 is often called the “Hamilton-Jacobi-Bellman” equation, because it is the stochastic generalization of the older Hamilton-Jacobi equation of enduring importance in fundamental physics. When control assumes no stochastic disturbance, it really is the older equation that one is solving. However, even in the deterministic cases, theorems like those of Barras require solution of the full Bellman equation (or at least the limit of it as noise goes to zero) in order to design a truly rigorous robust controller for the general nonlinear case. From that viewpoint, ADP is simply a useful toolbox to perform the calculations needed to implement nonlinear robust control in the general case.

Action-Dependent HDP (ADHDP), aka “Q Learning”

This family of ADP methods results from modeling a new value function J' (or “Q”) defined by:

$$J'(\underline{x}(t), \underline{u}(t)) = U(\underline{x}(t), \underline{u}(t)) + \gamma J(\underline{x}(t+1)) \quad (4)$$

One may develop a recurrence relation similar to (3) simply by substituting this into (3); the general method for estimating $\hat{J}'(\mathbf{R}(t), \mathbf{u}(t), \mathbf{W})$ is given in White and Sofge (1992), along with concrete practical applications from McDonnell Douglas in reconfigurable flight control and in low-cost mass production of carbon-carbon composite parts (a breakthrough important to the later development of the Dreamliner airplane after Boeing bought McDonnell Douglas and assimilated the technology). This type of value function has more complexity than the HDP value function, but its simplicity has value in some applications. One might argue that a truly general system like a brain would have an optimal hybrid of these two methods and more.

Dual Heuristic Programming (DHP, Dual HDP)

DHP uses yet another type of value function $\underline{\lambda}$ defined by:

$$\underline{\lambda}(\underline{x}(t)) = \nabla_{\underline{x}} J(\underline{x}(t)) \quad (5)$$

A convergent, consistent method for training this kind of value function is given in White and Sofge (1992), but many readers would find it useful to study the details and explanation by Balakrishnan and by Wunsch et al., reviewed in Lewis and Liu (2013). $\underline{\lambda}$ is a generalization of the dual variables well-known in economics, in operations research, and in control. The method given in White and Sofge (1992) is essentially a stochastic generalization of the Pontryagin equation. Because DHP entails a rich and detailed flow of feedback of values (plural) from one iteration to the next, it performs better in the face of more and more complexity, as discussed in theoretical terms in White and Sofge (1992) and shown concretely in simulation studies by Wunsch and Prokhorov reviewed in Lewis and Liu (2013). DHP value learning was the basis for Balakrishnan’s breakthrough in performance in hit-to-kill missile interception (Lewis and Liu

2013), which is possibly better known and understood in China than in the USA for political reasons.

Action-Dependent DHP and Globalized HDP

For reasons of length, I will not describe these three other groups discussed in White and Sofge (1992) in detail. The action-dependent version of DHP may have uses in training time-lagged recurrent networks in real time, and GDHP offers the highest performance and generality at the cost of some complexity. (In applications so far, DHP has performed as well. In the absence of noise, neural model predictive control (White and Sofge 1992) has also performed well and is sometimes considered as a (direct) type of reinforcement learning.)

Future Directions

Extensions to Spatial Complexity, Temporal Complexity, and Creativity

In the late 1990s, major extensions of ADP were designed and even patented to cope with much greater degrees of complexity, by adaptively making use of multiple time intervals and spatial structure (“chunking”) and by addressing the inescapable problem of local minima more effectively. See Werbos (2014) for a review of research steps needed to create systems which can cope with complexity as well as a mouse brain, for example. Realistically, however, the main challenge to ADP at present is to prove optimality under rigorous prior assumptions about “vector” plants (as in “vector intelligence”), considering systems which learn plants dynamics and optimal policy together in real time. It is possible that the more advanced systems may actually be dangerous if widely deployed before they are fully understood. It is amusing how movies like “Terminator 2” and “Terminator 3” may be understood as a (realistic) presentation of possible modes of instability in more complex nonlinear adaptive systems. In July 2014, the National Science Foundation of China and the relevant Dean at Tsinghua University asked me to propose

a major new joint research initiative based on Werbos (2014), with self-driving cars as a major testbed; however, the NSF of the USA responded at that time by terminating that activity.

Cross-References

- [Nonlinear Adaptive Control](#)
- [Numerical Methods for Nonlinear Optimal Control Problems](#)
- [Optimal Control and Pontryagin’s Maximum Principle](#)
- [Optimal Control and the Dynamic Programming Principle](#)
- [Reinforcement Learning for Approximate Optimal Control](#)

Bibliography

- Lewis FL, Liu D (eds), (2013) Reinforcement learning and approximate dynamic programming for feedback control, vol 17. Wiley (IEEE Series), New York
- Werbos PJ (2005) Backwards differentiation in AD and neural nets: past links and new opportunities. In: Bucker M, Corliss G, Hovland P, Naumann, Norris B (eds) Automatic differentiation: applications, theory and implementations. Springer, New York
- Werbos PJ (2014) Werbos, from ADP to the brain: foundations, roadmap, challenges and research priorities. In: Proceedings of the international joint conference on neural networks 2014. IEEE, New Jersey. <https://arxiv.org/abs/1404.0554>
- Werbos PJ, Davis JJ, (2016) Regular cycles of forward and backward signal propagation in prefrontal cortex and in consciousness. Front Syst Neurosci 10:97. <https://doi.org/10.3389/fnsys.2016.00097>
- White DA, Sofge DA (eds) (1992) Handbook of intelligent control: neural, fuzzy, and adaptive approaches. Van Nostrand Reinhold, New York

ATM Modernization

- [Air Traffic Management Modernization: Promise and Challenges](#)

Auctions

Bruce Hajek

University of Illinois, Urbana, IL, USA

Abstract

Auctions are procedures for selling one or more items to one or more bidders. Auctions induce games among the bidders, so notions of equilibrium from game theory can be applied to auctions. Auction theory aims to characterize and compare the equilibrium outcomes for different types of auctions. Combinatorial auctions arise when multiple-related items are sold simultaneously.

Keywords

Auction · Combinatorial auction · Game theory

Introduction

Three commonly used types of auctions for the sale of a single item are the following:

- *First price auction:* Each bidder submits a bid one of the bidders submitting the maximum bid wins, and the payment for the item is the maximum bid. (In this context “wins” means receives the item, no matter what the payment.)
- *Second price auction* or *Vickrey auction:* Each bidder submits a bid, one of the bidders submitting the maximum bid wins, and the payment for the item is the second highest bid.
- *English auction:* The price for the item increases continuously or in some small increments, and bidders drop out at some points in time. Once all but one of the bidders has dropped out, the remaining bidder wins and the payment is the price at which the last of the other bidders dropped out.

A key goal of the theory of auctions is to predict how the bidders will bid, and predict the resulting outcomes of the auction: which bidder is the winner and what is the payment. For example, a seller may be interested in the expected payment (seller revenue). A seller may have the option to choose one auction format over another and be interested in revenue comparisons. Another item of interest is efficiency or social welfare. For sale of a single item, the outcome is efficient if the item is sold to the bidder with the highest value for the item. The book of V. Krishna (2002) provides an excellent introduction to the theory of auctions.

Auctions Versus Seller Mechanisms

An important class of mechanisms within the theory of mechanism design are seller mechanisms, which implement the sale of one or more items to one or more bidders. Some authors would consider all such mechanisms to be auctions, but the definition of auctions is often more narrowly interpreted, with auctions being the subclass of seller mechanisms which do not depend on the fine details of the set of bidders. The rules of the three types of auction mentioned above do not depend on fine details of the bidders, such as the number of bidders or statistical information about how valuable the item is to particular bidders. In contrast, designing a procedure to sell an item to a known set of bidders under specific statistical assumptions about the bidders’ preferences in order to maximize the expected revenue (as in Myerson (1981)) would be considered a problem of mechanism design, which is outside the more narrowly defined scope of auctions. The narrower definition of auctions was championed by R. Wilson (1987). An article on ► [Mechanism Design](#) appears in this encyclopedia.

Equilibrium Strategies in Auctions

An auction induces a noncooperative game among the bidders, and a commonly used predictor of the outcome of the auction is an

equilibrium of the game, such as a Nash or Bayes-Nash equilibrium. For a risk neutral bidder i with value x_i for the item, if the bidder wins and the payment is M_i , the payoff of the bidder is $x_i - M_i$. If the bidder does not win, the payoff of the bidder is zero. If, instead, the bidder is risk averse with risk aversion measured by an increasing utility function u_i , the payoff of the bidder would be $u_i(x_i - M_i)$ if the bidder wins and $u_i(0)$ if the bidder does not win.

The second price auction format is characterized by simplicity of the bidding strategies. If bidder i knows the value x_i of the item to himself, then for the second price auction format, a weakly dominant strategy for the bidder is to truthfully report x_i as his bid for the item. Indeed, if y_i is the highest bid of the other bidders, the payoff of bidder i is $u_i(x_i - y_i)$ if he wins and $u_i(0)$ if he does not win. Thus, bidder i would prefer to win whenever $u_i(x_i - y_i) > u_i(0)$ and not win whenever $u_i(x_i - y_i) < u_i(0)$. That is precisely what happens if bidder i bids x_i , no matter what the bids of the other bidders are. That is, bidding x_i is a weakly dominant strategy for bidder i .

Nash equilibrium can be found for the other types of auctions under a model with incomplete information, in which the type of each bidder i is equal to the value of the object to the bidder and is modeled as a random variable X_i with a density function f_i supported by some interval $[a_i, b_i]$. A simple case is that the bidders are all risk neutral, the densities are all equal to some fixed density f , and the X_i 's are mutually independent. The English auction in this context is equivalent to the second price auction: in an English auction, dropping out when the price reaches his true value is a weakly dominant strategy for a bidder, and for the weakly dominant strategy equilibrium, the outcome of the auction is the same as for the second price auction. For the first price auction in this symmetric case, there exists a symmetric Bayesian equilibrium. It corresponds to all bidders using the bidding function β (so the bid of bidder i is $\beta(X_i)$), where β is given by $\beta(x) = E[Y_1 | Y_1 \leq x]$. The expected revenue to the seller in this case is $E[Y_1 | Y_1 < X_1]$, which is the same as the expected revenue for the second price auction and English auction.

Equilibrium for Auctions with Interdependent Valuations

Seminal work of Milgrom and Weber (1982) addresses the performance of the above three auction formats in case the bidders do not know the value of the item, but each bidder i has a private signal X_i about the value V_i of the item to bidder i . The values and signals $(X_1, \dots, X_n, V_1, \dots, V_n)$ can be interdependent. Under the assumption of invariance of the joint distribution of $(X_1, \dots, X_n, V_1, \dots, V_n)$ under permutation of the bidders and a strong form of positive correlation of the random variables $(X_1, \dots, X_n, V_1, \dots, V_n)$ (see Milgrom and Weber 1982 or Krishna 2002 for details), a symmetric Bayes-Nash equilibrium is identified for each of the three auction formats mentioned above, and the expected revenues for the three auction formats are shown to satisfy the ordering $R^{(\text{first price})} \leq R^{(\text{second price})} \leq R^{(\text{English})}$. A significant extension of the theory of Milgrom and Weber due to DeMarzo et al. (2005) is the theory of security-bid auctions in which bidders compete to buy an asset and the final payment is determined by a contract involving the value of the asset as revealed after the auction.

Combinatorial Auctions

Combinatorial auctions implement the simultaneous sale of multiple items. A simple version is the simultaneous ascending price auction with activity constraints (Cramton 2006; Milgrom 2004). Such an auction procedure was originally proposed by Preston, McAfee, Paul Milgrom, and Robert Wilson for the US FCC wireless spectrum auction in 1994 and was used for the vast majority of spectrum auctions worldwide since then Cramton (2013). The auction proceeds in rounds. In each round a minimum price is set for each item, with the minimum prices for the initial round being reserve prices set by the seller. A given bidder may place a bid on an item in a given round such that the bid is greater than or equal to the minimum price for the item. If one or more bidders bid on an item in a round, a provi-

sional winner of the item is selected from among the bidders with the highest bid for the item in the round, with the new provisional price being the highest bid. The minimum price for the item is increased 10 % (or some other small percentage) above the new provisional price. Once there is a round with no bids, the set of provisional winners is identified. Often constraints are placed on the bidders in the form of *activity rules*. An activity rule requires a bidder to maintain a history of bidding in order to continue bidding, so as to prevent bidders from not bidding in early rounds and bidding aggressively in later rounds. The motivation for activity rules is to promote *price discovery* to help bidders select the packages (or bundles) of items most suitable for them to buy. A key is that complementarities may exist among the items for a given bidder. Complementarity means that a bidder may place a significantly higher value on a bundle of items than the sum of values the bidder would place on the items individually. Complementarities lead to the *exposure problem*, which occurs when a bidder wins only a subset of items of a desired bundle at a price which is significantly higher than the value of the subset to the bidder. For example, a customer might place a high value on a particular pair of shoes, but little value on a single shoe alone.

A variation of simultaneous ascending price auctions for combinatorial auctions is auctions with package bidding (see, e.g., Ausubel and Milgrom 2002; Cramton 2013). A bidder will either win a package of items he bid for or no items, thereby eliminating the exposure problem. For example, in simultaneous clock auctions with package bidding, the price for each item increases according to a fixed schedule (the clock), and bidders report the packages of items they would prefer to purchase for the given prices. The price for a given item stops increasing when the number of bidders for that item drops to zero or one, and the clock phase of the auction is complete when the number of bidders for every item is zero or one. Following the clock phase, bidders can submit additional bids for packages of items. With the inputs from bidders acquired during the clock phase and supplemental bid phase, the auctioneer then runs a winner determination algorithm to

select a set of bids for non-overlapping packages that maximizes the sum of the bids. This winner determination problem is NP hard, but is computationally feasible using integer programming or dynamic programming methods for moderate numbers of items (perhaps up to 30). In addition, the vector of payments charged to the winners is determined by a two-step process. First, the (generalized) Vickrey price for each bidder is determined, which is defined to be the minimum the bidder would have had to bid in order to be a winner. Secondly, the vector of Vickrey prices is projected onto the core of the reported prices. The second step insures that no coalition consisting of a set of bidders and the seller can achieve a higher sum of payoffs (calculated using the bids received) for some different selection of winners than the coalition received under the outcome of the auction. While this is a promising family of auctions, the projection to the core introduces some incentive for bidders to deviate from truthful reporting, and much remains to be understood about such auctions.

Summary and Future Directions

Auction theory provides a good understanding of the outcomes of the standard auctions for the sale of a single item. Recently emerging auctions, such as for the generation and consumption of electrical power, and for selection of online advertisements, are challenging to analyze and comprise a direction for future research. Much remains to be understood in the theory of combinatorial auctions, such as the degree of incentive compatibility offered by core projecting auctions.

Cross-References

- [Game Theory: A General Introduction and a Historical Overview](#)
- [Option Games: The Interface Between Optimal Stopping and Game Theory](#)

Bibliography

- Ausubel LM, Milgrom PR (2002) Ascending auctions with package bidding. *BE J Theor Econ* 1(1): Article 1
- Cramton P (2006) Simultaneous ascending auctions. In: Cramton P, Shoham Y, Steinberg R (eds) *Combinatorial auctions*, chapter 4. MIT, Cambridge, pp 99–114
- Cramton P (2013) Spectrum auction design. *Rev Ind Organ* 42(4):161–190
- DeMarzo PM, Kremer I, Skrzypacz A (2005) Bidding with securities: auctions and security bidding with securities: auctions and security design. *Am Econ Rev* 95(4):936–959
- Krishna V (2002) *Auction theory*. Academic, San Diego
- Milgrom PR (2004) *Putting auction theory to work*. Cambridge University Press, Cambridge/ New York
- Milgrom PR, Weber RJ (1982) A theory of auctions and competitive bidding. *Econometrica* 50(5):1089–1122
- Myerson R (1981) Optimal auction design. *Math Oper Res* 6(1):58–73
- Wilson R (1987) Game theoretic analysis of trading processes. In: Bewley T (ed) *Advances in economic theory*. Cambridge University Press, Cambridge/New York

anesthetic drug mechanisms, the large inter-patient variability, and the difficulty in sensing the responses to anesthetic drugs. Following a brief introduction to general anesthesia, those challenges will be reviewed from a control engineering perspective. Recent advances in the understanding of the neurobiology of anesthesia and developments in sensing and monitoring have resulted in renewed interest in automatic control of anesthesia. These developments will be discussed, followed by a review of recent research in control of depth of anesthesia, as well as analgesia. The appropriateness of various control methodologies for this problem will also be discussed.

Keywords

Anesthesia · Automated drug delivery · Depth of hypnosis · Nociception · Robust control · Patient safety

Automated Anesthesia Systems

Guy A. Dumont^{1,2} and J. Mark Ansermino²

¹Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada

²BC Children's Hospital Research Institute, Vancouver, BC, Canada

Abstract

In many fields of human endeavor, ranging from the mundane such as DVD players to the technologically advanced such as space flight, control systems have become ubiquitous, to the point that control technology has been termed by Karl Astrom the “hidden technology.” However, in the field of anesthesia, despite efforts going back to 1950, closed-loop control is rarely used to automate the delivery of drugs to safely achieve and maintain a desired clinical anesthetic effect. This might be because of the complexity of physiological systems, the poor understanding of

Introduction

Anesthesia drug delivery involves the continuous administration of a combination of potentially dangerous drugs with frequent adjustments in order to induce a temporary loss of consciousness and memory while maintaining normal vital cardiorespiratory function during surgical stimulation. Over the last two decades, advances in central and autonomous nervous system monitoring technology have yielded a new set of real-time monitors and sensors to capture the effect of these drugs on the patient. As a result, automated feedback control of anesthetic drug delivery to a pre-defined state can be a means of providing the patient with a titration specifically adjusted to his or her needs. Although the idea of automated anesthetic drug delivery has been investigated for over 50 years, and despite rapid progress over the last 10 years, no real clinical breakthrough has yet been achieved and it has yet to become standard of care. In this entry, drawing from Dumont (2012), we attempt to give an overview of the current state of the art in the field.

Overview of Anesthesia

The goals of anesthesia are to allow the surgeon to operate in optimal conditions while (i) protecting the patient from the detrimental effects of the surgical procedure and (ii) maintaining homeostasis and hemodynamic stability as much as it is possible. For this, the anesthesiologist administers a number of drugs to the patients: hypnotics that act primarily on the brain to induce unconsciousness (also known as hypnotic state) in the patient so as to prevent intraoperative awareness and memorization; analgesics to suppress nociceptive reactions in presence of painful stimulus; and in some surgeries neuromuscular blocking agents to induce paralysis in order to suppress reflex muscle activity. The role of the anesthesiologist is to carefully dose the amount of drugs to avoid underdosing which can lead to intraoperative awareness and possible postoperative post-traumatic stress as well as overdosing which, due to the toxicity of the drugs involved, may lead to serious or even fatal intra- or postoperative consequences for the patient.

There are two broad classes of anesthetic agents: inhaled agents and intravenous agents. Modern inhaled anesthetics have a combined hypnotic and minor analgesic effect (and limited neuromuscular blocking action) and limited hypotensive action. An advantage of inhaled anesthetics is that measuring the difference between inhaled and exhaled concentrations allows an accurate estimation of plasma or brain drug uptake. Modern total intravenous anesthesia (TIVA) usually involves propofol as hypnotic agent and remifentanyl as analgesic agent. Propofol is characterized by fast redistribution and metabolism, provides rapid emergence, and has good anti-emetic properties. Remifentanyl is characterized by a very rapid onset and brevity of action, thus minimizing undesirable opioid-induced side effects. Together with the high specificity of both agents, this makes them ideal for control of anesthesia. Hence, the vast majority of studies of closed-loop control of anesthesia have been performed using TIVA based on propofol and remifentanyl.

We can divide the anesthesia procedure into three distinct stages: induction, maintenance, and emergence.

Induction Induction is the phase during which the patient goes from consciousness to unconsciousness and although quite short is critical to the overall safety of the procedure. As soon as the patient loses consciousness, he or she will usually stop breathing and need to be rapidly intubated to allow for artificial ventilation. To facilitate insertion of the endotracheal tube, the bolus of propofol is usually preceded by a bolus of opioid such as remifentanyl. Furthermore, in some situations, as soon as the patient loses consciousness, a NMB is administered, to optimize intubation conditions and blunt any reflex during intubation. Overdosing of the patient at induction may lead to severe hypotension which will need to be corrected with vasopressors and may place elderly or fragile patients into too deep an hypnotic state which may lead to prolonged periods of electrocortical silence, thought to have harmful long-term effects. Minimizing the amount of overshoot at induction is thus critical.

Maintenance After induction, it is necessary to maintain an adequate level of unconsciousness and to blunt nociceptive reactions. When using inhaled anesthetics, the measurement of the end-tidal vapor concentration provides the anesthesiologist with a reliable feedback quantity. The situation is more complex with TIVA, as there is no reliable method to provide arterial concentration of propofol or remifentanyl. In the absence of brain monitoring, the anesthesiologist will typically use hemodynamic parameters such as heart rate and blood pressure for guidance, or watch out for patient movement. The development of TIVA has been made easier through the development of pharmacokinetic model-driven infusion devices. These devices reach a desired plasma (or effect site) theoretical concentration by using a computer-controlled infusion pump driven by the pharmacokinetic parameters of the drug. The resulting target-controlled infusion (TCI) (Absalom and Struys 2007) anesthesia is used exten-

sively in most of the developed world except in the USA where it has not been approved by the FDA.

Emergence The emergence from anesthesia is simply achieved by turning off delivery of the hypnotic and analgesic agents used during the surgery. This is usually done during skin closure so that the patient wakes up faster at the end of the surgery. An additional bolus of a long-acting opioid may be given for postoperative pain management. Extubation takes place as soon as the patient shows clinical signs of wakefulness.

Sensing for Anesthesia

In this section, we will focus on sensing for the three major components of anesthesia, hypnosis, nociception, and muscular relaxation. This is above and beyond the monitoring of standard physiological parameters such as heart rate, respiratory rate, minute ventilation, airway pressure, end-tidal CO_2 through capnography, blood pressure (either noninvasive or through an arterial line), and oxygen saturation via pulse oximetry.

Sensing for Hypnosis

The effects of anesthetic drugs on the electroencephalogram (EEG) have been known since the early 1940s when neurophysiologists observed that the EEG of anesthetized patients contained slower waves with higher amplitudes. However, raw EEG is difficult to detect and interpret in real time, and thus a number of techniques have been used to extract univariate features from the EEG to quantify the hypnotic component of anesthesia. Two such features of historical interest are the median frequency (MEF) and the spectral edge frequency (SEF), i.e., the frequency up to which 95% of the EEG power is present. However, this is not until the advent of the BISTM monitor that EEG has become more commonplace. The BIS monitor is based on the observation that with increasing anesthetic depth, EEG frequencies tend to synchronize. This leads to the use of the bispectrum to characterize phase coupling of different frequencies, Sigl and Chamoun (1994).

The BISTM monitor combines a number of bispectra, bicoherence indices, and power spectral values to derive a [0–100] index known as depth of hypnosis (DoH). An index of 100 represents the awake state, while it decreases with increasing concentration of anesthetics. General anesthesia is obtained for an index between 60 and 40. Lower values represent deep hypnotic states and usually are not desirable. Introduced in the mid-1990s, the BIS monitor largely dominates the market for DoH monitors.

Another DoH monitor is the M-EntropyTM monitor, introduced in 2003, (Viertio-Oja et al. 2004) which provides two indices, the State Entropy (SE), a measure of the irregularity of frontal electroencephalogram activity within the frequency range of 0.8–32 Hz, and the Response Entropy (RE), a measure of the irregularity of frontal electroencephalogram activity within the frequency range of 0.8–47 Hz. While SE is a surrogate of the BIS, the difference between RE and SE is thought of as an indication of nociception because it may contain some facial EMG.

Although it provides anesthesiologists with a reliable index of hypnosis, the BIS monitor introduces a large and variable delay; is inherently nonlinear, tending to evolve in a stepwise manner during transient phases of anesthesia; and is essentially a black box hard to characterize for control design purposes. On the other hand, the M-Entropy responds much faster, is a simpler algorithm, but tends to provide a very noisy index.

The more recent NeuroSenseTM monitor that addresses these concerns was developed specifically for use in closed-loop control of anesthesia. It derives a bilateral index based on wavelet decomposition of a frontal EEG, with emphasis on the γ -band activity, Zikov et al. (2006). It has been shown to relate well to the BIS in steady state, but possesses much faster, delay-free, and constant dynamics over its entire range, Bibian et al. (2011).

Sensing for Nociception

Clinical monitoring of nociception or antinociception is based on capturing the response

of the autonomic nervous system (ANS) to a noxious stimulus. For example, heart rate and blood pressure both tend to increase sharply in case of a sympathetic response to a non-properly blunted noxious stimulus. This is, for instance, the basis for the AnalgoscoreTM, a pain score derived from heart rate (HR) and mean arterial pressure (MAP) that has been used in closed-loop control, Hemmerling et al. (2009). The GE's Surgical Pleth Index (SPITM), first introduced as the Surgical Stress Index (SSI), is computed from finger photoplethysmographic (PPG) waveform amplitudes and pulse-to-pulse intervals, Wennervirta et al. (2008). The ANITM monitor analyzes the tachogram with wavelets and tracks the time-varying power in the high-frequency band. It has been shown to respond to the administration of anesthetic drugs and to nociceptive stimuli (Jeanne et al. 2009). A related technique is based on wavelet-based cardiorespiratory coherence, resulting in a normalized index that has been shown to respond to both nociceptive and anti-nociceptive events, Brouse et al. (2010). The NOLTM index by Medasense uses machine learning to derive an index from features of the PPG, heart rate variability (HRV), skin conductance, movement, and temperature; see Ledowski (2019) for more on nociception monitoring.

An interesting method for assessing nociceptive reactions is based on the observation that a sudden electrocortical activation, e.g., demonstrated by an increase in exDoH values, is a reflection of an inadequate analgesic state. This principle has been used in the most clinically tested closed-loop controller for anesthesia, Liu et al. (2011).

Modelling for Anesthesia

Modelling of the distribution and effect of anesthetic drugs has traditionally been done using pharmacokinetic (PK) models for the former and pharmacodynamic (PD) models for the latter. Typically, pharmacokinetic models are based on compartmental models, while PD models consist of a simple compartment followed by a sigmoidal

nonlinearity. For propofol, a three-compartment is used, yielding the transfer function between the infusion rate I_p and the plasma concentration C_p :

$$C_p = \frac{1}{V_1} \frac{(s + z_1)(s + z_2)}{(s + p_1)(s + p_2)(s + p_3)} I_p(s)$$

The PD model is usually described by a transfer function between the plasma concentration C_p and the effect site concentration C_e

$$C_e(s) = \frac{k_{e0}}{s + k_{e0}} C_p(s)$$

followed by a Hill equation relating C_e to the effect:

$$E(C_e) = \frac{C_e^\gamma}{EC_{50}^\gamma + C_e^\gamma}$$

where EC_{50} denotes the effect site concentration corresponding to a 50% clinical effect and γ is the cooperativity coefficient. For remifentanyl, most PK models involve only two compartments, resulting in a simpler transfer function (Fig. 1).

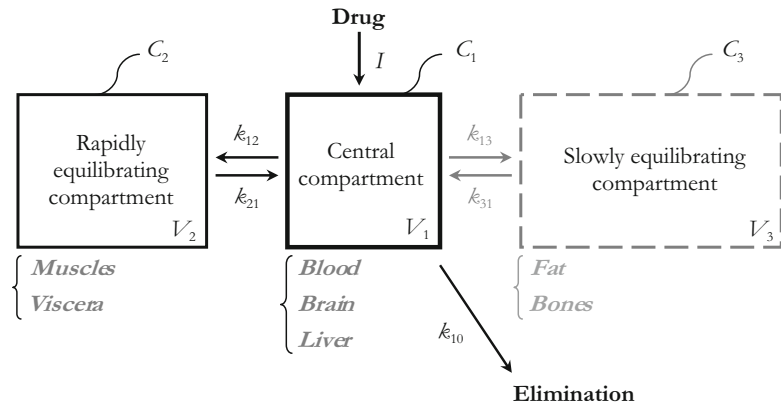
Propofol and remifentanyl are known to interact with each other in a synergistic fashion in their hypnotic/analgesic effect (Kern et al. 2004). This observation constitutes the basic assumption on which the concept of balanced anesthesia is based. Static pharmacodynamic interactions between these two drugs have been studied using the concept of isoboles reflecting, e.g., the probability of response to painful stimuli through the use of response surfaces, as:

$$E(v_p, v_r) = \frac{(v_p + v + r + \alpha v_p v_r)^\gamma}{(v_p + v + r + \alpha v_p v_r)^\gamma + 1}$$

where v_p and v_r are, respectively, the effect site concentrations of propofol and remifentanyl normalized by their EC_{50} , and $\alpha > 0$ characterizes the synergy between the two drugs. Note that the interaction is equivalent to the use of a new fictitious drug $v = v_p + v_r + \alpha v_p v_r$.

While pharmacokinetic and dynamicists strive to improve the accuracy of PKPD models with the introduction of a number of covariates in an attempt to reduce the uncertainty in TCI systems, a number of studies have shown that in

Automated Anesthesia Systems, Fig. 1 A three-compartment PK model



order to develop a clinically satisfactory closed-loop control systems, simpler models such as first-order plus delay may have the same level of predictive power.

Control of Anesthesia

Control Paradigm

After ensuring a fast and safe induction, the anesthesiologist needs to maintain the patient in an adequate state of hypnosis and analgesia by adjusting the anesthetic and opioid titration rates in order to avoid both under- and overdosing of the patient. An automated system that would regulate drug dosing to maintain the adequacy of the anesthetic regimen could reduce the anesthesiologist's workload. Closed-loop anesthesia is not meant to replace the anesthesiologist, but to allow them to concentrate on higher-level tasks focusing on the well-being of the patient.

A closed-loop controller for anesthesia should induce the patient rapidly, but with minimal overshoot, and then maintain the patient in an adequate state of anesthesia and analgesia at least as well as an expert anesthesiologist. Translating this in control specifications is difficult, but for a DoH index, it could be translated into a rise time at induction of 3–4 min, with overshoot less than 10–15% and a good damping ratio of at least 0.7. During maintenance the DoH index should stay about 85% of the time within 10 points of the target. Control of analgesia should be such that in case of arousal (which in control engineering

terms can be thought of as an output disturbance), the patient response is rapidly suppressed, say within 2 min and without inducing oscillations. The clinical outcome should be improved hemodynamic stability, faster emergence, and possibly reduced drug consumption. The main challenge is the inherent variability, both interpatient and intraoperative; thus, robust stability and performance are paramount.

Historical Period

The first efforts to automate anesthesia go back to the work of Mayo and Bickford in the early 1950s with their attempts to develop EEG-based automatic delivery of volatile agents, see, e.g., Bickford (1950). What follows is by no means an exhaustive review of the published work on closed-loop control of anesthesia. Closer to us, in the 1980s a significant amount of work was performed on end-tidal concentration control for inhaled anesthetics such as halothane (see, e.g., Westenskow et al. 1986); closed-loop control of neuromuscular blockade (see, e.g., Brown et al. 1980); or mean arterial pressure control (Monk et al. 1989). In 1989, Schwilden (Schwilden et al. 1989) published the first work on closed-loop delivery of propofol guided by the EEG median frequency. during a study on 11 healthy volunteers. For a review of the progress from 1949 to 1980, see Chilcoat (1980).

The Modern Period

The advent of the BIS DoH monitor in the mid-1990s resulted in a significant increase in the

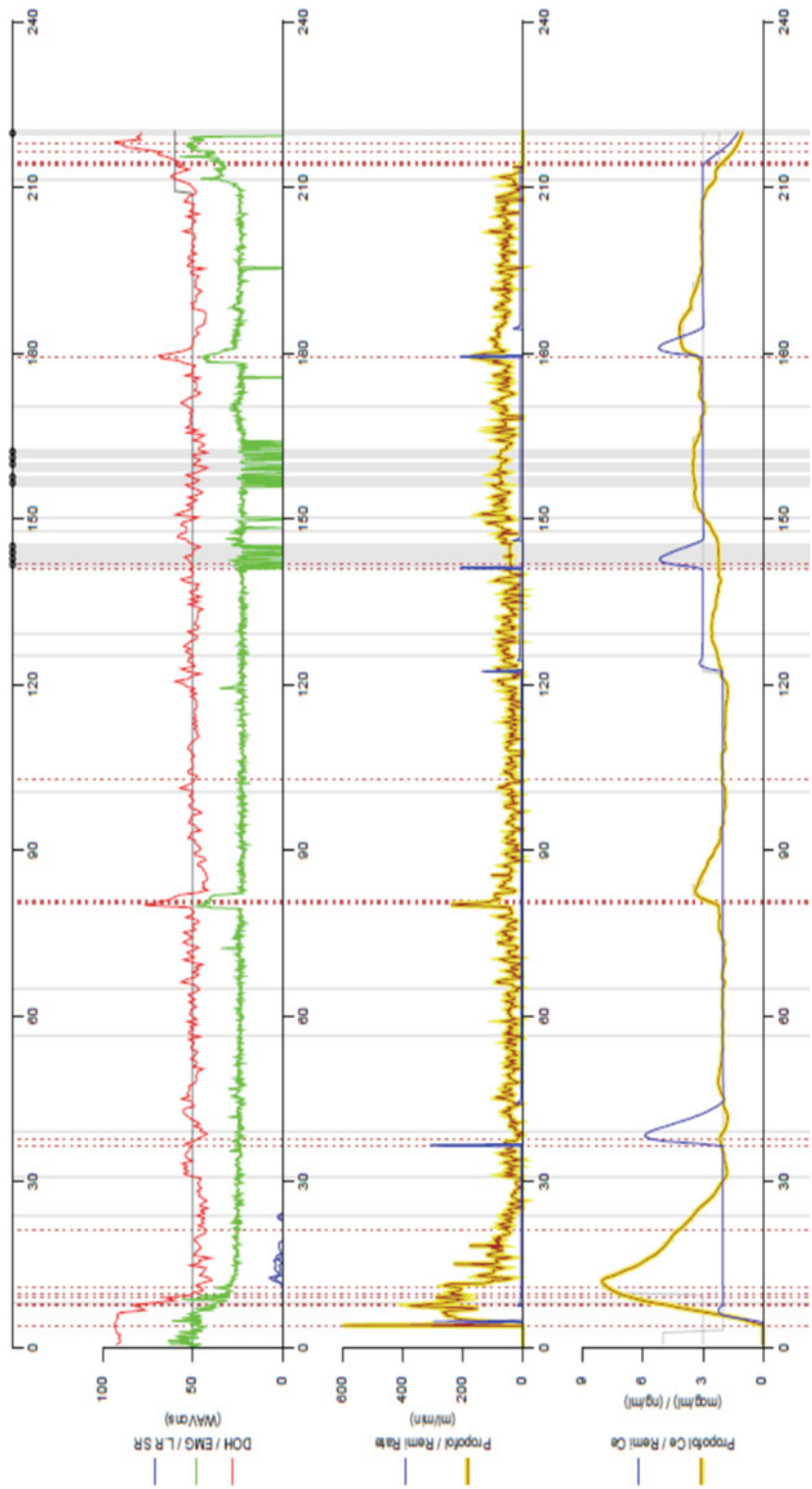
number of both simulated and clinical studies on closed-loop control of anesthesia, focusing on DoH.

An early effort by an engineering group was the development of DoH control during isoflurane anesthesia and its clinical testing on human volunteers during minor surgical procedures, Gentilini et al. (2001b). The system consisted of two cascaded loops, the slave one being in charge of isoflurane end-tidal concentration at a setpoint given by the master loop in charge of maintaining the BIS DoH index between 40 and 50. Both controllers were model-based internal model controllers (IMC). The system behaved satisfactorily during clinical tests. The patients, however, were induced manually, the controllers taking over only for the maintenance phase. That group also considered control of mean arterial pressure by closed-loop control of alfentanil using a PKPD model-based explicit predictive controller with constraints, Gentilini et al. (2002). These two systems were then combined for joint control of hypnosis and analgesia during successful clinical tests, Gentilini et al. (2001a).

Early efforts at BIS-guided closed-loop control of propofol infusion were performed by Absalom et al. (2002) and Absalom and Kenny (2003) who used a PID controller tuned in a very ad hoc manner to adjust the propofol effect site concentration target of a TCI system. In clinical tests on 20 patients, the performance varied significantly from patient to patient, the system displaying instability for some of the patients. All patients were induced under TCI mode, with an effect site concentration chosen by the clinician who then switched to closed-loop control for the maintenance phase. A system with a similar structure is described in Liu et al. (2006), using a rule-based controller that is very similar to a PD controller. After significant tuning, this system was tested against manual control in a randomized controlled trial involving 164 patients (83 in closed-loop). That system was shown to outperform manual control in terms of BIS variability and resulted in similar hemodynamic stability. Puri et al. (2007) describes a heuristically tuned “adaptive” PID

controller tested against manual control in a clinical trial involving 40 subjects. A similar system is described in Hemmerling et al. (2010) which tests a heuristic set of rules emulating a PD controller against manual control in a clinical trial involving 40 subjects. Both studies report similar results but lack a detailed description of the control algorithms. In Liu et al. (2011), a system that manipulates both the propofol and remifentanyl infusion rates based on the BIS alone is presented. The basic idea is that sudden increases in the BIS index are due to a nociceptive reaction and reflect an inadequate analgesic state. The controller that manipulates the remifentanyl is a combination of a proportional action and a number of heuristic rules. Randomized clinical trials involving 167 patients on a wide variety of procedures showed that the system provides better control of the BIS than manual control and similar hemodynamic stability accompanied with increased remifentanyl consumption. This system, like its predecessor, induces the patient in TCI mode, with manually set targets for both propofol and remifentanyl effect site concentrations.

Because the above systems were designed heuristically, their theoretical properties are difficult, if not impossible to assess. A rigorous approach to robust PID tuning for anesthesia is described in Dumont et al. (2009) where a PID controller is robustly tuned for a population of 44 adult patients. Results of a feasibility study in adults showed that this simple controller provided adequate anesthesia (Dumont et al. 2011). This led to the development of a similar system for pediatric use (van Heusden et al. 2014; West et al. 2013), whose results in a clinical study indicate that a robust PID controller provides clinically satisfactory performance despite the significant interpatient uncertainty in that pediatric population. An extension of this system that manipulates both propofol and remifentanyl infusion rates based only on the DoH index was developed using a modified habituating control framework (van Heusden et al. 2018) and successfully tested on 127 patients undergoing surgical procedures (West et al. 2018). An important feature of that system is its safety system



Automated Anesthesia Systems, Fig. 3 Example of closed-loop control of anesthesia for a 220 min procedure for a 37-year-old patient. The top pane shows the evolution of the depth of hypnosis (red trace) as well as the EMG signal (green trace). At induction, the DoH drops to about 50, which is the setpoint aimed for during maintenance. It then goes back up at emergence, a few minutes after and of drug infusion. The middle pane shows the infusion for propofol (yellow) and remifentanyl (blue). The bottom pane shows the corresponding theoretical plasma concentrations for propofol and remifentanyl

loop control of anesthesia, the regulatory hurdles abound before it can be approved for routine clinical use (Manberg et al. 2008). The US FDA has made it clear that these systems will have to be based on sound and verifiable control theory and that it will require guarantees in terms of stability, robustness, and performance. For the technology to mature and become clinical reality, close cooperation between control engineers, clinicians, medical device manufacturers, and regulatory bodies will be required.

It is also important to realize that the systems discussed here only consider a subset of what an anesthesiologist needs to do to ensure patient safety. Other systems, e.g., for hemodynamics and fluid management, are under development, and some are even undergoing clinical trials. Some clinical trials combining DoH closed-loop control together with hemodynamic (blood pressure) control and automated fluid management have recently taken place.

For those systems to be adopted clinically, their benefits in terms of improved patient outcomes will have to be clearly demonstrated, and this will necessitate large and costly multicenter clinical studies involving tens of thousands of patients.

Cross-References

- [PID Control](#)
- [Robust Control in Gap Metric](#)

Bibliography

- Absalom AR, Kenny GNC (2003) Closed loop control of propofol anaesthesia using bispectral index: performance assessment in patients receiving computer-controlled propofol and manually controlled remifentanyl infusions for minor surgery. *Br J Anaesth* 90(6):737–741
- Absalom A, Struys MMRF (2007) An overview of TCI & TIVA, 2nd edn. Academia Press, Gent
- Absalom A, Sutcliffe N, Kenny G (2002) Closed-loop control of anesthesia using bispectral index. *Anesthesiology* 96(1):67–73
- Bibian S, Dumont G, Zikov T (2011) Dynamic behavior of BIS, m-entropy and NeuroSENSE brain function monitors. *J Clin Monit Comput* 25(1):81–87
- Bickford R (1950) Automatic electroencephalographic control of general anesthesia. *Electroencephalog Clin Neurophysiol* 2:93
- Brouse C, Dumont G, Myers D, Cooke E, Lim J, Ansermino J (2010) Wavelet transform cardiorespiratory coherence for monitoring nociception. In: *Computing in cardiology 2010*, Belfast
- Brown B, Asbury J, Linkens D, Perks R, Anthony M (1980) Closed-loop control of muscle relaxation during surgery. *Clin Phys Physiol Meas* 1:203–210
- Chilcoat R (1980) A review of the control of depth of anaesthesia. *Trans Inst Meas Control* 2:38–45
- de Smet T, Struys M, Neckebroek M, van den Hauwe K, Bonte S, Mortier E (2008) The accuracy and clinical feasibility of a new bayesian-based closed-loop control system for propofol administration using the bispectral index as a controlled variable. *Anesth Analg* 107:1200–1210
- Dumont G (2012) Closed-loop control of anesthesia – a review. In: *8th IFAC symposium on biological and medical systems*, Budapest
- Dumont G, Martínez A, Ansermino J (2009) Robust control of depth of anesthesia. *Int J Adapt Control Signal Process* 23(5):435–454
- Dumont G, Liu N, Petersen C, Chazot T, Fischler M, Ansermino J (2011) Closed-loop administration of propofol guided by the neurosense: clinical evaluation using robust proportional- integral-derivative design. *Anesthesiology*, p A1170
- Gentilini A, Frei C, Glattfedler A, Morari M, Sieber T, Wymann R, Schnider T, Zbinden A (2001a) Multitasked closed-loop control in anesthesia. *IEEE Eng Med Biol* 20:39–53
- Gentilini A, Rossoni-Gerosa M, Frei CW, Wymann R, Morari M, Zbinden AM, Schnider TW (2001b) Modeling and closed-loop control of hypnosis by means of bispectral index (BIS) with isoflurane. *IEEE Trans Biomed Eng* 48(8):874–889
- Gentilini A, Schaniel C, Bieniok C, Morari M, Wymann R, Schnider T (2002) A new paradigm for the closed-loop intraoperative administration of analgesics in humans. *IEEE Trans Biomed Eng* 49:289–299
- Haddad WM, Hayakawa T, Bailey JM (2006) Adaptive control for nonlinear compartmental dynamical systems with applications to clinical pharmacology. *Syst Control Lett* 55(1):62–70
- Hemmerling T, Charabati S, Salhab E, Bracco D, Mathieu P (2009) The analgoscoreTM: a novel score to monitor intraoperative nociception and its use for closed-loop application of remifentanyl. *J Comput* 4(4):311–318
- Hemmerling T, Charabati S, Zauouter C, Minardi C, Mathieu P (2010) A randomized controlled trial demonstrates that a novel closed-loop propofol system performs better hypnosis control than manual administration. *Can J Anaesth* 57:725–735
- Ionescu CM, Keyser RD, Torricco BC, Smet TD, Struys MMRF, Normey-Rico JE (2008) Robust predictive control strategy applied for propofol dosing using bis as a controlled variable during anesthesia. *IEEE Trans Biomed Eng* 55(9):2161–2170

- Janda M, Simanski O, Bajorat J, Pohl B, Noeldge-Schomburg GFE, Hofmockel R (2011) Clinical evaluation of a simultaneous closed-loop anaesthesia control system for depth of anaesthesia and neuromuscular blockade. *Anaesthesia* 66:1112–1120
- Jeanne M, Logier R, Jonckheere JD, Tavernier B (2009) Heart rate variability during total intravenous anaesthesia: effects of nociception and analgesia. *Auton Neurosci Basic Clin* 147:91
- Kern SE, Xie G, White JL, Egan TD (2004) Opioid-hypnotic synergy: a response surface analysis of propofol-remifentanyl pharmacodynamic interaction in volunteers. *Anesthesiology* 100(6):1374–1381
- Ledowski T (2019) Objective monitoring of nociception: a review of current commercial solutions. *Br J Anaesth* 123(2):e312–e321
- Liu N, Chazot T, Genty A, Landais A, Restoux A, McGee K, Laloë PA, Trillat B, Barvais L, Fischler M (2006) Titration of propofol for anesthetic induction and maintenance guided by the bispectral index: closed-loop versus manual control. *Anesthesiology* 104(4):686–695
- Liu N, Chazot T, Hamada S, Landais A, Boichut N, Dussaussoy C, Trillat B, Beydon L, Samain E, Sessler DI, Fischler M (2011) Closed-loop coadministration of propofol and remifentanyl guided by bispectral index: a randomized multicenter study. *Anesth Analg* 112(3):546–557
- Mahfouf M, Nunes CS, Linkens DA, Peacock JE (2005) Modeling and multivariable control in anaesthesia using neural-fuzzy paradigms part II. Closed-loop control of simultaneous administration of propofol and remifentanyl. *Artif Intell Med* 35(3):207–213
- Manberg P, Vozella C, Kelley S (2008) Regulatory challenges facing closed-loop anesthetic drug infusion devices. *Clin Pharmacol Ther* 84:166–169
- Monk C, Millard R, Hutton P, Prys-Roberts C (1989) Automatic arterial pressure regulation using isoflurane: comparison with manual control. *Br J Anaesth* 63:22–30
- Puri G, Kumar B, Aveek J (2007) Closed-loop anaesthesia delivery system (clads) using bispectral index: A performance assessment study. *Anaesth Intensive Care* 35:357–362
- Ralph M, Beck CL, Bloom M (2011) I_1 -adaptive methods for control of patient response to anaesthesia. In: *Proceedings of 2011 American control conference*, San Francisco
- Sawaguchi Y, Furutani E, Shirakami G, Araki M, Fukuta K (2008) A model-predictive hypnosis control system under total intravenous anaesthesia. *IEEE Trans Biomed Eng* 55(3):874–887
- Schwilden H, Stoeckel H, Schuttler J (1989) Closed-loop feedback control of propofol anaesthesia by quantitative eeg analysis in humans. *Br J Anaesth* 62:290–296
- Sigl J, Chamoun N (1994) An introduction to bispectral analysis for the electroencephalogram. *J Clin Monitor Comput* 10(6):309–404
- Simanski O, Janda M, Schubert A, Bajorat J, Hofmockel R, Lampe B (2009) Progress of automatic drug delivery in anaesthesia: the rostock assistant system for anaesthesia control (RAN)*. *Int J Adapt Control Signal Process* 23:504–521
- van Heusden K, Dumont G, Soltesz K, Petersen C, Umedaly A, West N, Ansermino J (2014) Design and clinical evaluation of robust pid control of propofol anaesthesia in children. *IEEE Trans Control Syst Technol* 22(2):491–501
- van Heusden K, Ansermino J, Dumont G (2018) Robust miso control of propofol-remifentanyl anaesthesia guided by the neurosense monitor. *IEEE Trans Control Syst Technol* 26(5):1758–1770
- Viertio-Oja H, Maja V, Sarkela M, Talja P, Tenkanen N, Tolvanen-Laakso H et al (2004) Description of the entropy algorithm as applied in the datex-ohmeda s/s entropy module. *Acta Anaesthesiol Scand* 48:154–161
- Wennervirta J, Hynynen M, Koivusalo AM, Uutela K, Huiku M, Vakkuri A (2008) Surgical stress index as a measure of nociception/antinociception balance during general anaesthesia. *Acta Anaesthesiol Scand* 52(8):1038–1045
- West N, Dumont G, vanHeusden K, Petersen C, Khosravi S, Soltesz K, Umedaly A, Reimer E, Ansermino J (2013) Robust closed-loop control of induction and maintenance of propofol anaesthesia in children. *Pediatr Anesth* 23(8):712–719
- West N, vanHeusden K, Görges M, Brodie S, Petersen C, Dumont G, Ansermino J, Merchant R (2018) Design and evaluation of a closed-loop anaesthesia system with robust control and safety system. *Anesth Analg* 12(4):883–894
- Westenskow D, Zbinden A, Thomson D, Kohler B (1986) Control of end-tidal halothane concentration – part A: anaesthesia breathing system and feedback control of gas delivery. *Br J Anaesth* 58:555–562
- Yousefi M, vanHeusden K, West N, Mitchell I, Ansermino J, Dumont G (2019) A formalized safety system for closed-loop anaesthesia with pharmacokinetic and pharmacodynamic constraints. *Control Eng Pract* 84:23–31
- Zikov T, Bibian S, Dumont G, Huzmezan M, Ries C (2006) Quantifying cortical activity during general anaesthesia using wavelet analysis. *IEEE Trans Biomed Eng* 53(4):617–632

Automated Insulin Dosing for Type 1 Diabetes

B. Wayne Bequette

Chemical and Biological Engineering,

Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract

The development of automated insulin delivery (also known as a closed-loop artificial

pancreas) systems has been an active research area since the 1960s, with an intense focus since 2005. In the United States in 2019, there is currently one commercial device available, with others under development. There is also a strong do-it-yourself community of individuals developing and using closed-loop technology. In this chapter we provide an overview of the challenges in developing automated insulin delivery systems and the algorithms that are commonly used to regulate blood glucose.

Keywords

Biomedical control · Drug infusion · Fault detection · Clinical trials

Introduction

The beta and alpha cells of the pancreas secrete insulin and glucagon, respectively, to regulate blood glucose (BG) concentrations; insulin decreases BG, while glucagon increases BG. The pancreas of an individual with type 1 diabetes no longer produces these hormones, so they must take insulin (by injection, or using continuous insulin infusion pumps) to survive. This is in contrast to people with type 2 diabetes, who have a pancreas that produces insulin but not enough to adequately regulate their BG. People with type 2 diabetes (90–95% of the diabetes population) often regulate BG through diet, exercise, and oral medications. The focus of this chapter is on automated insulin dosing (AID) systems for people with type 1 diabetes.

Before the discovery by Banting and Best that insulin was the hormone that regulated blood glucose in 1921, anyone diagnosed with diabetes was doomed to a short life; for an overview of the history of the development and use of insulin, see Hirsch (2004). For many decades someone with type 1 diabetes survived with multiple daily injections of insulin, but without real feedback. By the 1940s urine could be tested for glucose concentration, but easy-to-use urine strips were not available until the 1960s. Blood glucose

test meters were not widely available for home use by an individual until the 1980s. Finally, blood glucose control was not recognized to be important until results of the Diabetes Control and Complications Trial (DCCT) were published in 1993. For many people today the state of the art remains multiple daily injections of insulin (one long-acting shot in the morning, followed by injections of rapid-acting insulin at mealtime, or to correct for high blood glucose) and multiple finger-stick measurements of blood glucose using self-monitoring blood glucose (SMBG) meters – ideally before bedtime, when awakening, at each meal, and before potentially dangerous activities such as driving.

The use of insulin pumps became more common in the late 1990s; these pumps are operated “open-loop” with specified rates of insulin delivery, using rapid-acting insulin. Continuous glucose monitors (CGM), which provide a measurement of the interstitial fluid (just underneath the skin) glucose at frequent intervals, typically every 5 min, have been widely available for less than 15 years. These insulin pumps and CGMs are absolutely critical components of a closed-loop automated insulin dosing (artificial pancreas) system. Note that the CGM signal is related to blood glucose (BG) but will often lag the actual BG, may be biased, will have some uncertainty, and could suffer from signal attenuation or dropout. Devices for a closed-loop system are shown in Fig. 1.

To begin our discussion, it is helpful to understand common units and orders of magnitude. In the United States, blood glucose is measured in mg/dl. A healthy individual (without diabetes) will typically have a fasting BG level of 80–90 mg/dl, with brief excursions to 125–150 mg/dl due to meals. An individual with type 1 diabetes can drift to over 400 mg/dl (hyperglycemia) if insufficient insulin is given and yet be in danger of going below 70 mg/dl (hypoglycemia) if too much insulin is given. The common quantity of insulin is international units, which we will refer to as units (U) of insulin throughout this chapter. An individual with type 1 diabetes may have a basal (steady-state) insulin infusion rate of 1 U/h, or 24 U/day, but inject another 24 U of insulin

Automated Insulin Dosing for Type 1 Diabetes, Fig. 1

Components of a closed-loop automated insulin delivery (IAD) system: CGM (sensor), pump and infusion set, smartphone or other receiver with a control interface and containing the control algorithm and other logic. The number of devices can be reduced by putting the control algorithm and interface directly on the insulin pump



throughout the day to compensate for meals. A population range might be 0.5–3 U/h basal (12–72 U/day), with another 12–72 U/day for meals (while a 50-50 split of basal-meal insulin is a rule of thumb, this can vary significantly from person to person). An individual will typically require one U of insulin for each 10–15 g carbohydrates consumed in a meal. Individuals will also make use of a correction or insulin sensitivity factor to decide how much insulin to give to reduce their blood glucose levels. Typically, 1 U of insulin will reduce BG by 40–80 mg/dL; while time is not explicitly mentioned in these factors, it will typically take 2 h to reduce the BG. Also notice that, since a bolus (injection) of insulin is a pulse input, the fact that a new pseudo steady-state in BG occurs is indicative of an integrating process, at least for the short time scale. On a longer time scale, however, there is a one-to-one relationship between the insulin basal rate (U/h) and BG (mg/dl).

The focus of this discussion has been on insulin, because there is no current glucagon formulation that is stable for long periods of time at body temperature. If someone needs to raise their BG, they will consume carbohydrates; currently glucagon is used to rescue someone who passed out or is in a coma and cannot eat – a glucagon solution is quickly mixed and injected when needed. Stable forms of glucagon are under

development, and glucagon has been used in some clinical studies as noted below.

Challenges

There are multiple challenges to individuals regulating BG. Insulin delivered subcutaneously takes a long time to act to decrease the BG. While the pancreas of a healthy individual has a peak in insulin action less than 5 min after a “glucose challenge,” insulin delivered subcutaneously has a peak action of roughly 75 min and continues to act for 4–8 h. Thus, an individual must realize that insulin given 2 h ago may have 50% of the insulin effect remaining. Current insulin pumps allow individuals to keep track of “insulin on board” (IOB), an estimation of the amount of insulin remaining to act. Bequette (2009) reviews the research protocols used to develop pharmacodynamics models and to calculate IOB.

Meals and exercise represent the greatest “disturbances” to regulating BG. The dynamic impact of a meal varies tremendously with meal content – a glass of orange juice can have a rapid effect on BG (and is often taken if an individual is becoming hypoglycemic), while a high-fat meal has an effect that extends over several hours. A meal also requires a significant amount of insulin to compensate for the carbohydrates in the meal.

An individual with a basal insulin requirement of 1 U/h and a carb-to-insulin ratio of 10 g/U might need 7.5 U of insulin to cover a 75 g carbohydrate meal. Thus, the amount of insulin given for the meal is equivalent to 7.5 h worth of basal insulin. It would be rare for any manufacturing process to have this type of rangeability in an actuator!

Aerobic exercise can cause a relatively rapid decrease in BG and lead to reduced insulin needs for several hours, while anaerobic activity may result in a short-term increase in BG. An individual with type 1 diabetes walks a tightrope between under-delivery of insulin, which can lead to high blood glucose (hyperglycemia) and over-delivery of insulin, which can lead to low blood glucose (hypoglycemia). The risks of hyperglycemia are generally long-term in that higher mean glucose levels are correlated with micro- and macrovascular diseases and retinopathy. The risks of hypoglycemia, on the other hand, are largely short-term, such as drowsiness and, in a worst-case scenario, a diabetic coma leading possibly to death.

Overnight hyperglycemia is perhaps the greatest fear of a parent of a child with type 1 diabetes; it is common for a parent to check blood glucose at around 1 am or so. This fear often causes people to under-deliver insulin overnight, which can lead to hyperglycemia and long-term consequences. CGMs that provided alarms to awake individuals, or their parents, were not as effective as expected, partially due to a relatively high false positive rate for the devices at that time (the early 2000s); these problems were noted by Buckingham et al. (2005). These challenges motivated the development of initial closed-loop systems, in the form of low-glucose suspend algorithms, which shut off an insulin pump to avoid hypoglycemia, with a focus on overnight applications.

A number of faults can occur with sensors and insulin infusion sets. CGM signals can attenuate due to a person putting pressure on the sensor. Also, Bluetooth or other signals between the devices can drop out. Infusion sets are normally worn for 3 days before replacement, but these sets can fail in less time, sometimes immediately after insertion. Bequette (2014) provides an overview of these problems and possible fault detection techniques.

Control Algorithms and Objectives

Four types of algorithms have typically been used to regulate glucose by manipulating insulin infusion: (i) on-off, (ii) fuzzy logic/expert systems, (iii) proportional-integral-derivative (PID), and (iv) model predictive control (MPC). The control objectives are often either to control to a specific set point or to control to a desired range of glucose; the set point or range can vary with time of day. An overview of the different algorithms is provided by Bequette (2012).

Early closed-loop research focused on overnight control of blood glucose using a simple on-off algorithm, that is, if the glucose reached a hyperglycemic threshold, then the pump was shut off (low-glucose suspend) for a period of time. The next advance was to develop a predictive low-glucose suspend (PLGS) algorithm to shut off the pump if the CGM was predicted to go low during a future prediction horizon (often 30 min) (Buckingham et al. 2010; Cameron et al. 2012a).

A fuzzy logic-based strategy, based on the CGM, its rate of change, and acceleration, is used by Mauseth et al. (2010). The MD-Logic system by Atlas et al. (2010) uses a combination of set point and control-to-range concepts.

The proportional-integral-derivative (PID) controller is ubiquitous in chemical process applications. Steil et al. (2011) proposed a PID controller with a model-based insulin feedback term, making it similar to a cascade control strategy; an analysis of this approach is provided by Palerm (2011). The model-based insulin feedback plays a similar role to IOB since high model-predicted insulin predictions correspond to a high IOB.

Model predictive control (MPC) uses a model to forecast the effect of proposed insulin infusion rates on glucose over a prediction horizon. The majority of proposed AID algorithms are based on MPC. Objectives include tracking a BG set point (which could be changing with time), keeping BG within a desired range (Zone-MPC or control to range) (Kovatchev et al. 2009b; Grosman et al. 2010), or minimizing risk. Cameron et al. (2012b) develop a multiple model probabilistic predictive control (MMPPC)

strategy that accounts for uncertainty in predictions and manipulates insulin to minimize hypoglycemic risk.

In a simulation-based study, Cameron et al. (2011) compare basal-bolus, PID, MPC, and an enhanced MPC algorithm based on risk management. The merits of PID and MPC were analyzed by Steil (2013) and Bequette (2013), respectively. An overview of the different algorithms, delivery methods, and other engineering decisions is provided by Doyle et al. (2014). Much of the effort has involved the use of a single hormone, but a number of two-hormone (insulin and glucagon) studies have been performed.

Most algorithms require that a meal be “announced” to provide feedforward control through the associated meal insulin bolus; this requires the individual to estimate the carbohydrates in their meals and provide this information to the “hybrid” controller (the term commonly used for the combination of feedforward and feedback control). The MMPPC approach of Cameron et al. (2012b, 2014) also anticipates and detects meals, reducing the burden on individuals to provide a meal announcement and insulin bolus at mealtime.

Exercise can rapidly decrease BG and increase insulin sensitivity (reducing the insulin required) for many hours; thus it is desirable to provide exercise information as part of an AID strategy. Stenerson et al. (2014) incorporate heart rate and an accelerometer into a PLGS system but find little additional benefit due to the use of heart rate. Turksoy et al. (2015) use energy expenditure and galvanic skin resistance as additional sensor inputs to improve BG control during exercise. Breton et al. (2014) add heart rate to a control-to-range MPC strategy to improve BG regulation during exercise.

Fault Detection

Possible component-related faults include loss of the sensor signal (due to Bluetooth dropouts), pressure-induced sensor attenuation (PISA, due to pressure on the sensor), partial or complete insulin infusion set failure, and loss of controller

(smart phone battery depletion). Baysal et al. (2014) present an approach to detect PISAs based on signal characteristics such as rate of change. Howsmon et al. (2017, 2018) present a statistical process monitoring type of approach to detect insulin infusion rate failures. Most systems default to a pre-programmed basal insulin delivery rate upon loss of communication of 20 min or more.

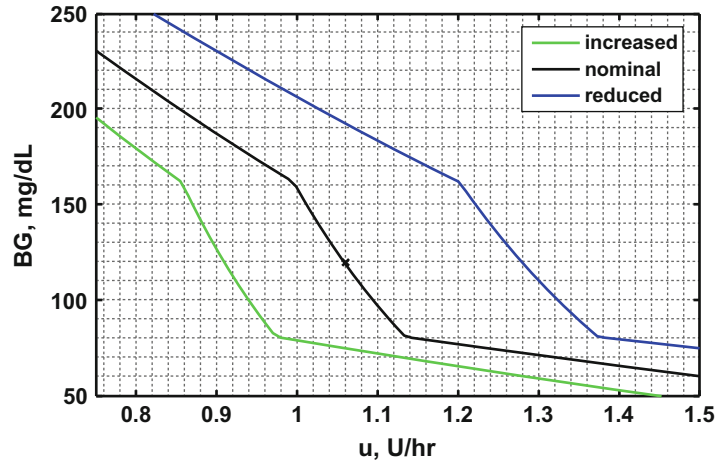
Simulation Models

There are two commonly used models to simulate the response of blood glucose to subcutaneous insulin infusion and meal carbohydrates. Kovatchev et al. (2009a) discuss the use of a simulator that has been accepted by the FDA to demonstrate preliminary results that can be used in investigational device exemption (IDE) applications for clinical trial studies. Wilinska et al. (2010) present a simulation framework based on the model developed by Hovorka et al. (2004).

The Hovorka simulation model is used to illustrate the effects of insulin and carbohydrates on BG. Figure 2 compares the effect of basal insulin rate on the steady-state BG for three different insulin sensitivities (nominal $\pm$ 20%). Notice that even with a fixed sensitivity a relatively small change in insulin basal rate has a large effect on the steady-state BG. Also, insulin sensitivity varies with time of day and with exercise and illness. In this example, the nominal sensitivity curve with a basal rate of 1.058 U/h yields a steady-state BG of 120 mg/dL. If the actual sensitivity is 20% less, the BG is 190 mg/dL, while if the sensitivity is 20% more, the BG is 75 mg/dL. An individual that struggles with hypoglycemia is likely to err on the side of a lower basal insulin rate. Figure 3 is an open-loop simulation for a 50 g carbohydrate meal, with and without an insulin bolus provided at mealtime, with the desired range of 70–180 mg/dL also plotted; while it is clearly important to provide an insulin bolus at mealtime, it is known that many adolescents fail to do this two or more times a week, leading to higher mean BG levels.

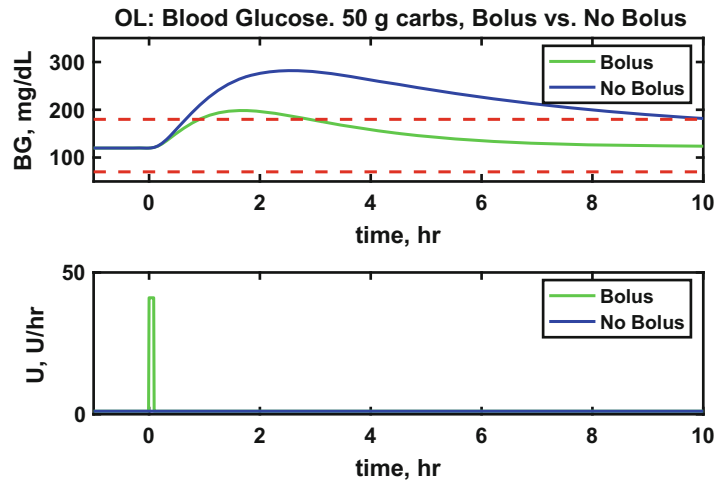
Automated Insulin Dosing for Type 1 Diabetes, Fig. 2

Steady-state relationship between insulin delivery and BG, for three different insulin sensitivities. The nominal condition used in the dynamic simulations that follow is shown as the “x” (1.058 U/h, 120 mg/dl)



Automated Insulin Dosing for Type 1 Diabetes, Fig. 3

Nominal sensitivity and open-loop simulations for a 50 g carbohydrate meal. Desired range of 70–180 mg/dl is shown. Illustrates the importance of providing an insulin bolus at mealtime



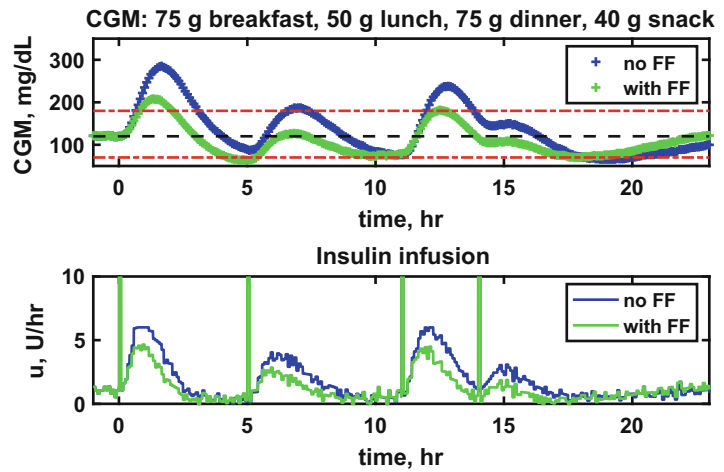
The closed-loop (using PID) simulation study shown in Fig. 4 is for breakfast, lunch, dinner, and a snack of 75, 50, 75, and 40 g, respectively; a scenario with feedforward/feedback control (insulin bolus at mealtime) is compared with feedback only. Clearly much better results can be achieved when a “meal announcement” is given so that an insulin bolus can be delivered; the artificial pancreas literature refers to this feedforward/feedback strategy as “hybrid control.” Forlenza et al. (2018) have shown clinically the BG control improvement when an insulin bolus is given 20 min ahead of the meal vs. using only feedback control action.

Clinical Trial Results

Much effort is required to conduct clinical trials, including (in the United States) approvals from an Institute Review Board (IRB), an FDA Investigational Device Exemption (IDE), review by a data safety management (DSMB) team, and clinical trials registration; see Bequette et al. (2016, 2018) for overviews of the regulatory process. Initially, simulation-based studies are conducted to validate expected algorithm performance (Patek et al. 2009). Typically, system safety is then verified in a hospital or clinical research center. These are often followed, after DSMB

Automated Insulin Dosing for Type 1 Diabetes, Fig. 4

Closed-loop simulations of PID-based feedback with and without feedforward control. Nominal sensitivity with three meals (75, 50, 75 g) and a snack (40 g); desired range of 70–180 mg/dl is shown. Illustrates the importance of providing an insulin bolus at mealtime (feedforward or meal announcement)



A

review and perhaps another IDE, by supervised outpatient studies in a hotel or diabetes camp and then by short- and long-term outpatient studies at home. Clinical studies have been inconsistent with performance metrics, so Maahs et al. (2016) proposed clinical trial metrics that include mean CGM as well as %CGM time in range for a large number of ranges. A recent review of clinical trial results is presented by Boughton and Hovorka (2019), for both single (insulin) and multi-hormone (insulin + glucagon) studies.

Buckingham and co-workers, in a series of articles, studied 160 subjects over 5722 subject nights using a predictive low-glucose suspend (PLGS) algorithm and showed hypoglycemia reduction in all age groups. A modification to the algorithm also bolused insulin if the glucose was predicted to be high, using a predictive hypo-/hyper-glycemic mitigation (PHHM) algorithm; 58 subjects over 2436 subject nights were studied in this phase. The PLGS and PHHM studies are summarized in Bequette et al. (2018); PHHM can be viewed as a control-to-range algorithm focused on overnight care.

Pinsker et al. (2016) report clinical trial results of a head-to-head comparison of MPC and PID in a study involving 30 participants and concluded that MPC had better performance with more time in range and lower mean glucose. A Zone-MPC study by Forlenza et al. (2017) was designed for prolonged infusion set wear, specifically to increase the probability of infusion set failures,

in a 2-week study involving 19 subjects. While most studies use “meal announcement” (feed-forward control), Cameron et al. (2017) present results on a multiple model probabilistic predictive control (MMPPC) system that anticipates and detects meals and does not require meal announcement. An MMPPC-based study by Forlenza et al. (2018) demonstrates the improvement and control that can be achieved if meal insulin boluses are given 20 min in advance of the meal. Dassau et al. (2017) report a comprehensive study involving 29 subjects over 12 weeks. In the longest study to date, Musolino et al. (2019) propose a protocol for a 6-month study involving 130 participants; the nonlinear MPC algorithm developed by Hovorka et al. (2004) will be used in this study.

A number of studies use both insulin and glucagon. El-Khatib et al. (2010) report in-patient clinical studies using a PD controller that is active under certain glucose concentrations to manipulate glucagon. Insulin is administered based on an adaptive, model predictive control strategy with a very short prediction horizon, making it similar to a PID controller. Russell et al. (2014) present outpatient results for a 5-day study with 20 adults and 32 adolescents. Blauw et al. (2016) stress the advantages of using a single integrated device (rather than separate smartphones and pumps) to manipulate both insulin and glucagon in a 4-day study involving 10 subjects. El-Khatib et al. (2017) study 39 subjects in a dual-arm at-home

study of 11 days in closed-loop and 11 days in conventional therapy.

Commercial Devices

The first threshold-based low-glucose suspend system approved by the US FDA (in 2013) was the Medtronic 530G, while the Medtronic 640G uses a predictive low-glucose suspend algorithm. The first closed-loop system approved by the US FDA (in 2017) was the Medtronic 670G, which is commonly called a “hybrid closed-loop” system because meal announcement (an estimated of the amount of carbohydrates in the meal, with the associated insulin bolus) is required. Garg et al. (2017) report results on adolescents and adults in a 3-month at-home study using the 670G.

As of July 2019, a number of potential commercial devices have gone through various stages of clinical trials. Buckingham et al. (2018) report results for an OmniPod system based on the OmniPod patch pump, a Dexcom CGM, and a MPC algorithm developed by Doyle and co-workers at UCSB and Harvard. Brown et al. (2018) report a study involving five subjects and the use of a T:slim Pump, Dexcom CGM, and a “ControlIQ” algorithm developed at the University of Virginia by Kovatchev and co-workers. The iLet system by Damiano at Boston University is a device that delivers both insulin and glucagon; they plan to first commercialize the insulin-only system. Bigfoot Biomedical also has a closed-loop system under development, based on a model predictive control algorithm.

Do-It-Yourself (DIY) Movement

Frustrated by the slow development in commercial devices for automated insulin delivery, a large do-it-yourself community has started an open-source movement. Initial commercial CGM manufacturers did not provide a way of sharing CGM data in real time, leading to a community called Nightscout that shared ways of “hacking” CGMs to share data in real time in 2013 (Lee et al. 2017). Soon thereafter (2014)

the open APS movement began, allowing DIY “Loopers” to implement their own closed-loop systems (Lewis 2018). The general approach that was developed is similar to a model predictive control algorithm, using models to predict the effect of meals and insulin on future CGM values (Diamond et al. 2019). The algorithm calculates the current insulin bolus (when spread over a 30-min interval) that will yield the desired CGM value at the end of the prediction horizon (the insulin action time). This involves an algebraic calculation rather than an optimization problem that is used in traditional MPC; the calculated insulin is not delivered if any CGM is predicted to be below a hypoglycemic threshold during the prediction horizon. Barnard et al. (2018) discuss challenges and potential benefits of collaborations between DIY individuals, device manufacturers, regulatory agencies, and caregivers.

Summary and Future Directions

Automated insulin delivery systems are far more than control algorithms to regulate BG by manipulating insulin (and perhaps glucagon). It is important to have easy-to-calibrate (or calibration-free) sensors (CGM), insulin infusion pumps, and infusion sets that have a low failure rate and easy-to-use interfaces to make it clear whether the system is under manual (open) or automatic (closed) loop. System faults should result in defaulting to a safe mode, which again should be clear to the user.

Future systems will better integrate knowledge about lifestyle (predictable exercise, eating and sleep times) and will include the use of calendar information (e.g., lunch meeting) and additional sensors (such as accelerometers and gyroscope from a smart watch). Indeed, Navarathna et al. (2018) report preliminary results for detecting meal motions using a smart watch, enabling an advisory feedforward action for meal announcement.

It is desirable to reduce the number of devices that must be placed on the body for current AID systems, which include a CGM (with transmitter), insulin infusion catheter attached to an

insulin pump, and a smart phone or other device used for the controller. There is some activity toward incorporating the CGM and infusion set into the same device. Indeed, if these were incorporated into a patch pump that also contained the interface and algorithm, only a single device would need to be placed on the body. It is likely, however, that a smart phone would still be used to communicate with the patch pump which may be worn under clothing.

Cross-References

- [Control of Drug Delivery for Type 1 Diabetes Mellitus](#)
- [Model Predictive Control for Power Networks](#)
- [Model Predictive Control in Practice](#)
- [PID Control](#)

Recommended Reading

Doyle et al. (2014) review the many different engineering decisions that must be made when developing AID systems. Castle et al. (2017) provide a comprehensive appraisal of the future of AID. Ramkisson et al. (2017) present a detailed assessment of safety hazards associated with AID. A special issue on Artificial Pancreas systems was published in the February 2018 issue of *IEEE Control Systems Magazine*. Cinar (2018) provides an overview of the papers in the issue. Huyett et al. (2018) presented the effect of glucose sensor (CGM) dynamics. Bondia et al. (2018) review the estimation of insulin pharmacokinetics and pharmacodynamics. El Fathi et al. (2018) focus on meal control, while Messori et al. (2018) perform a simulation-based study of an MPC strategy using individualized parameters. Turksoy et al. (2018) present an adaptive control based procedure to include activity monitor sensors in addition to CGM signals. Bequette et al. (2018) provide an overview of a 5000 subject night study to reduce hypoglycemic risk overnight.

Bibliography

- Atlas E, Nimri R, Miller S, Grunberg EA, Phillip M (2010) MD-logic artificial pancreas systems. *Diabetes Care* 33(5):1072–1076
- Barnard KD, Ziegler R, Klonoff DC, Braune K, Petersen B, Rendschmidt T, Finan D, Kowalski A, Heinemann L (2018). Open source closed-loop insulin delivery systems: a clash of cultures or merging of diverse approaches? *J Diabetes Sci Technol* 12(6):1223–1226
- Baysal N, Cameron F, Buckingham BA, Wilson DM, Chase HP, Maahs DM, Bequette BW (2014) A novel method to detect pressure-induced sensor attenuations (PISA) in an artificial pancreas. *J Diabetes Sci Technol* 8(6):1091–1096
- Bequette BW (2009) Glucose clamp algorithms and insulin time-action profiles. *J Diabetes Sci Technol* 3(5):1005–1013
- Bequette BW (2012) Challenges and progress in the development of a closed-loop artificial pancreas. *Annu Rev Control* 36:255–266
- Bequette BW (2013) Algorithms for a closed-loop artificial pancreas: the case for model predictive control (MPC). *J Diabetes Sci Technol* 7(6):1632–1643
- Bequette BW (2014) Fault detection and safety in closed-loop artificial pancreas systems. *J Diabetes Sci Technol* 8(6):1204–1214
- Bequette BW, Cameron F, Baysal N, Howsmon DP, Buckingham BA, Maahs DM, Levy CJ (2016) Algorithms for a single hormone closed-loop artificial pancreas: challenges pertinent to chemical process operations and control. *Processes* 4(4):39. <https://doi.org/10.3390/pr4040039>
- Bequette BW, Cameron F, Buckingham BA, Maahs DM, Lum J (2018) Overnight hypoglycemia and hyperglycemia mitigation for individuals with type 1 diabetes. How risks can be reduced. *IEEE Control Syst* 38(1):125–134. <https://doi.org/10.1109/MCS.2017.2767119>
- Blauw H, van Bon AC, Koops R, DeVries JH (2016) Performance and safety of an integrated artificial pancreas for fully automated glucose control at home. *Diabetes Obes Metab* 18:671–677
- Bondia J, Romero-Vivo S, Ricaret B, Diez JL (2018) Insulin estimation and prediction. *IEEE Control Syst Mag* 38(1):47–66
- Boughton CK, Hovorka R (2019) Advances in artificial pancreas systems. *Sci Transl Med* 11:484. eaaw4949. <https://doi.org/10.1126/scitranslmed.aaw4949>
- Breton MD, Brown SA, Karvetski CH, Kollar L, Topchyan KA, Anderson SM, Kovatchev BP (2014) Adding heart rate signal to a control-to-range artificial pancreas system improves the protection against hypoglycemia during exercise in type 1 diabetes. *Diabetes Technol Ther* 16(8):506–11. <https://doi.org/10.1089/dia.2013.0333>
- Brown S, Raghinaru D, Emory E, Kovatchev B (2018) First look at control-IQ: a new-generation automated insulin delivery system. *Diabetes Care* 41(12):2634–2636. <https://doi.org/10.2337/dc18-1249>

- Buckingham BA, Block J, Burdick J, Kalajian A, Kollman C, Choy M et al (2005) Response to nocturnal alarms using a real-time glucose sensor. *Diabetes Technol Ther* 7:440–447
- Buckingham B, Chase HP, Dassau E, Cobry E, Clinton P, Gage V, Caswell K, Wilkinson J, Cameron F, Lee H, Bequette BW, Doyle FJ III (2010) Prevention of nocturnal hypoglycemia using predictive alarm algorithms and insulin pump suspension. *Diabetes Care* 33(5):1013–1018
- Buckingham BA, Forlenza GP, Pinsker JE, Christiansen MP, Wadwa RP, Schneider J, Peyser TA, Dassau E, Lee JB, O'Connor J, Layne JE, Ly TT (2018) Safety and feasibility of the OmniPod hybrid closed-system in adult, adolescent, and pediatric patients with type 1 diabetes using a personalized model predictive control algorithm. *Diabetes Technol Ther* 20(4):257–262
- Cameron F, Bequette BW, Wilson DM, Buckingham BA, Lee H, Niemeyer G (2011) A closed-loop artificial pancreas based on risk management. *J Diabetes Sci Technol* 5(2):368–379
- Cameron F, Wilson DM, Buckingham BA, Arzumanyan H, Clinton P, Chase HP, Lum J, Maahs DM, Calhoun PM, Bequette BW (2012a) In-patient studies of a Kalman filter based predictive pump shut-off algorithm. *J Diabetes Sci Technol* 6(5):1142–1147
- Cameron F, Niemeyer G, Bequette BW (2012b) Extended multiple model prediction with application to blood glucose regulation. *J Process Control* 22(7):1422–1432
- Cameron F, Niemeyer G, Wilson DM, Bequette BW, Benassi KS, Clinton P, Buckingham BA (2014) Inpatient trial of an artificial pancreas based on multiple model probabilistic predictive control (MMPPC) with repeated large unannounced meals. *Diabetes Technol Ther* 16(11):728–734
- Cameron FM, Ly TT, Buckingham BA, Maahs DM, Forlenza GP, Levy CJ, Lam D, Clinton P, Messer LH, Westfall E, Levister C, Xie YY, Baysal N, Howsmon D, Patek SD, Bequette BW (2017) Closed-loop control without meal announcement in type 1 diabetes. *Diabetes Technol Ther* 19(9):527–532. <https://doi.org/10.1089/dia.2017.0078>
- Castle JR, DeVries JH, Kovatchev B (2017) Future of automated insulin delivery. *Diabetes Technol Ther* 19(S3):S-67–S-72
- Cinar A (2018) Artificial pancreas systems. *IEEE Control Syst Mag* 38(1):26–29
- Dassau E, Pinsker JE, Kudva YC, Brown SA, Gondhalekar R, Dalla Man C, Patek S, Schiavon M, Dadlani V, Dasanayake I, Church MM, Carter RE, Bevier WC, Huyett LM, Hughes J, Anderson S, Lv D, Schertz E, Emory E, McCrady-Spitzer SK, Jean T, Bradley PK, Hinshaw L, Sanz AJL, Basu A, Kovatchev B, Cobelli C, Doyle FJ III (2017) Twelve-week 24/7 ambulatory artificial pancreas with weekly adaptation of insulin delivery settings: effect on hemoglobin A1c in hypoglycemia. *Diabetes Care* 40:1719–1726
- Diabetes Control and Complications Trial (DCCT) Research Group (1993) The effect of intensive treatment of diabetes on the development and progression of long-term complications in insulin-dependent diabetes mellitus. *N Engl J Med* 329:977–986
- Diamond T, Bequette BW, Cameron F (2019) A control systems analysis of “DIY looping.” Presented at the 2019 Diabetes Technology Meeting, Bethesda
- Doyle FJ III, Huyett LM, Lee JB, Zisser HC, Dassau E (2014) Closed-loop artificial pancreas systems: engineering the algorithms. *Diabetes Care* 37:1191–1197
- El-Khatib FH, Russell SJ, Nathan DM, Sutherlin RG, Damiano ER (2010) A bihormonal closed-loop artificial pancreas for type 1 diabetes. *Sci Transl Med* 2:27ra27
- El-Khatib F, Balliro C, Hillar MA, Magyar KL, Ekhlaspour L, Sinha M, Mondesir D, Esmaili A, Hartigan C, Thompson MJ, Malkani S, Lock JP, Harlan DM, Clinton P, Frank E, Wilson DM, DeSalvo D, Norlander L, Ly T, Buckingham BA, Diner J, Dezube M, Young LA, Goley A, Kirkman MS, Buse JB, Zheng H, Selagamsetty RR, Damiano ER, Russell SJ (2017) Home use of bihormonal bionic pancreas versus insulin pump therapy in adults with type 1 diabetes: a multicentre randomised crossover trial. *Lancet* 389:369–380
- El Fathi A, Smaqui MR, Gingras V, Bolout B, Haidar A (2018) The artificial pancreas and meal control. *IEEE Control Syst Mag* 38(1):67–85
- Forlenza GP, Deshpande S, Ly TT, Howsmon DP, Cameron F, Baysal N, Mauritzen E, Marcal T, Towers L, Bequette BW, Huyett LM, Pinsker JE, Gondhalekar R, Doyle FJ III, Maahs DM, Buckingham BA, Dassau E (2017) Application of zone model predictive control artificial pancreas during extended use of infusion set and sensor: a randomized crossover-controlled home-use trial. *Diabetes Care* 40:1096–1102. <https://doi.org/10.2337/dc17-0500>
- Forlenza GP, Cameron FM, Ly TT, Lam D, Howsmon DP, Baysal N, Kulina G, Messer L, Clinton P, Levister C, Patek SD, Levy CJ, Wadwa RP, Maahs DM, Bequette BW, Buckingham BA (2018) Fully closed-loop multiple model predictive controller (MMPPC) artificial pancreas (AP) performance in adolescents and adults in a supervised hotel setting. *Diabetes Technol Ther* 20(5):335–343
- Garg SK, Weinzimer SA, Tamborlane WV, Buckingham BA, Bode BW, Bailey TS, Brazg RL, Ilany J, Slover RH, Anderson SM, Bergental RM, Grosman B, Roy A, Cordero TL, Shin J, Lee SW, Kaufman FR (2017) Glucose outcomes with the in-hoe use of a hybrid closed-loop insulin delivery system in adolescents and adults with type 1 diabetes. *Diabetes Technol Ther* 19(3):156–163
- Grosman B, Dassau E, Zisser HC, Jovanovic L, Doyle FJ III (2010) Zone model predictive control: a strategy to minimize hyper- and hypoglycemic events. *J Diabetes Sci Technol* 4:961–975
- Hirsch IB (2004) Treatment of patients with severe insulin deficiency: what we have learned over the past 2 years. *Am J Med* 116(3A):17S–22S
- Hovorka R, Canonico V, Chassin LJ, Haueter U, Massi-Benedetti M, Fedrici MO et al (2004) Nonlinear model

- predictive control of glucose concentration in subjects with type 1 diabetes. *Physiol Meas* 25(4):905–920
- Howson DP, Cameron F, Baysal N, Ly TT, Forlenza GP, Maahs DM, Buckingham BA, Hahn J, Bequette BW (2017) Continuous glucose monitoring enables the detection of losses in infusion set actuation (LISAs). *Sensors* 17:161. <https://doi.org/10.3390/s17010161>
- Howson DP, Baysal N, Buckingham BA, Forlenza GP, Ly TT, Maahs DM, Marcal T, Towers L, Mauritzen E, Deshpande S, Huyett LM, Pinsky JE, Gondhalekar R, Doyle FJ III, Dassau E, Hahn J, Bequette BW (2018) Real-time detection of infusion site failures in a closed-loop artificial pancreas. *J Diabetes Sci Technol* 12(3):599–607. <https://doi.org/10.1177/19322968187551>
- Huyett LM, Dassau E, Zisser HC, Doyle FJ III (2018) Glucose sensor dynamics and the artificial pancreas. *IEEE Control Syst Mag* 39(1):30–46
- Kovatchev BP, Breton M, Dalla Man C, Cobelli C (2009a) In silico preclinical trials: a proof of concept in closed-loop control of type 1 diabetes. *J Diabetes Sci Technol* 3(1):44–55
- Kovatchev B, Patek S, Dassau E, Doyle FJ III, Magni L, De Nicolao G et al (2009b) The juvenile diabetes research foundation artificial pancreas consortium. Control to range for diabetes: functionality and modular architecture. *J Diabetes Sci Technol* 3(5):1058–1065
- Lee JM, Newman MW, Gebremariam A, Choi P, Lewis D, Nordgren W, Costik J, Wedding J, West B, Gilby N, Benovich C, Pasek J, Garrity A, Hirschfeld E (2017) Real-world use and self-reported health outcomes of a patient-designed do-it-yourself mobile technology system for diabetes: lessons for mobile health. *Diabetes Technol Ther* 19(4):209219
- Lewis D (2018) History and perspective on DIY closed looping. *J Diabetes Sci Technol* 13(4):790–793
- Maahs DM, Buckingham BA, Castle JR, Cinar A, Damiano ER, Dassau E, DeVries JH, Doyle III FJ, Griffen SC, Haidar A et al (2016) Outcome measures for artificial pancreas clinical trials: a consensus report. *Diabetes Care* 39:1175–1179
- Mauseth R, Wang Y, Dassau E, Kircher R, Matheson D, Zisser H et al (2010) Proposed clinical application for tuning fuzzy logic controller of artificial pancreas utilizing a personalization factor. *J Diabetes Sci Technol* 4:913–922
- Messori M, Incremona CP, Cobelli C, Magni (2018) Individualized model predictive control for the artificial pancreas. *IEEE Control Syst Mag* 38(1):86–104
- Musolino G, Allen JM, Hartnell S, Wilinska ME, Tauschmann M, Boughton C, Campbell F, Denvir L, Trevelyan N, Wadwa P, DiMeglio L, Buckingham BA, Weinzierl S, Acerini CL, Hood K, Fox S, Kollman C, Sibayan J, Borgman S, Cheng P, Hovorka R (2019) Assessing the efficacy, safety and utility of 6-month day-and-night automated closed-loop insulin delivery under free-living conditions compared with insulin pump therapy in children and adolescents with type 1 diabetes: an open-label, multicenter, multinational, single-period, randomized, parallel group study protocol. *BMJ Open* 9:e027856. <https://doi.org/10.1136/bmjopen-2018-027856>
- Navarathna P, Bequette BW, Cameron F (2018) Device based activity recognition and prediction for improved feedforward control. 2018 American control conference, Milwaukee, pp 3571–3576. <https://doi.org/10.23919/ACC.2018.8430775>
- Palerm CC (2011) Physiologic insulin delivery with insulin feedback: a control systems perspective. *Comput Methods Prog Biomed* 102(2):130–137
- Patek SD, Bequette BW, Breton M, Buckingham BA, Dassau E, Doyle FJ III, Lum J, Magni L, Zisser H (2009) In silico preclinical trials: methodology and engineering guide to closed-loop control. *J Diabetes Sci Technol* 3(2):269–282
- Pinsky JE, Lee JB, Dassau E, Seborg DE, Bradley PK, Gondhalekar R, Bevier WC, Huyett L, Zisser HC, Doyle FJ 3rd (2016) Randomized crossover comparison of personalized MPC and PID control algorithms for the artificial pancreas. *Diabetes Care* 39(7):1135–1142. <https://doi.org/10.2337/dc15-2344>. PubMed PMID: 27289127; PMCID: PMC4915560
- Ramkissoon C, Aufderheide B, Bequette BW, Vehi J (2017) Safety and hazards associated with the artificial pancreas. *IEEE Rev Biomed Eng* 10:44–62. <https://doi.org/10.1109/RBME.2017.2749038>
- Russell SJ, El-Khatib F, Manasi S, Magyar KL, McKeon K, Goergen LG, ... Damiano ER (2014) Outpatient glycemic control with a bionic pancreas in type 1 diabetes. *N Engl J Med* 371(4):313–325. <https://doi.org/10.1056/NEJMoa1314474>
- Steil GM (2013) Algorithms for a closed-loop artificial pancreas: the case for proportional-integral-derivative control. *J Diabetes Sci Technol* 7(6):1621–1631
- Steil GM, Palerm CC, Kurtz N, Voskanyan G, Roy A, Paz S, Kandeel FR (2011) The effect of insulin feedback on closed loop glucose control. *J Clin Endocrinol Metab* 96(5):1402–1408
- Stenerson M, Cameron F, Wilson DM, Harris B, Payne S, Bequette BW, Buckingham BA (2014) The impact of accelerometer and heart rate data on hypoglycemia mitigation in type 1 diabetes. *J Diabetes Sci Technol* 8(1):64–69
- Turksoy K, Paulino TM, Zaharieva DP, Yavelberg L, Jamnik V, Riddell MC, Cinar A (2015) Classification of physical activity: information to artificial pancreas control systems in real time. *J Diabetes Sci Technol* 9(6):1200–1207. <https://doi.org/10.1177/1932296815609369>
- Turksoy T, Littlejohn E, Cinar A (2018) Multimodule, multivariable artificial pancreas for patients with type 1 diabetes. *IEEE Control Syst Mag* 38(1):105–124
- Wilinska ME, Chassin LJ, Acerini CL, Allen JM, Dunger DB, Hovorka R (2010) Simulation environment to evaluate closed-loop insulin delivery systems in type 1 diabetes. *J Diabetes Sci Technol* 4:132–144

Automated Truck Driving

Valerio Turri¹, Jonas Mårtensson¹, and
Karl H. Johansson^{1,2}

¹Electrical Engineering and Computer Science,
KTH – Royal Institute of Technology,
Stockholm, Sweden

²ACCESS Linnaeus Center, Royal Institute of
Technology, Stockholm, Sweden

Abstract

Truck transportation offers unique use cases for the early deployment of automation technologies. In this article, we first provide an overview of applications of automated vehicles in the truck industry, from simple cruise controllers to complex driverless functionalities. We then focus on two promising use cases of automated truck driving: platoon-based cooperative freight transportation and driverless transportation in mining sites. Platooning allows trucks to reduce energy consumption and greenhouse gas emissions of about 10%. The further removal of the driver in follower vehicles yields consistent economical benefits for the fleet owner. As platoon-capable trucks will always represent a small portion of the overall traffic, here, we discuss a control architecture for freight transportation that can support the efficient creation of platoons and coordinate trucks from different owners. The second use case is on driverless transportation in mining sites. A mining site is a controlled environment where the site operator has a good overview of what is happening and the amount of unforeseen events is limited. Transportation tasks tend to be repetitive, monotone, and suited for autonomous system. Here, we present a functional architecture that is able to support the driverless operation of trucks in such an environment.

Keywords

ADAS · Truck platooning · Driverless truck · Vehicle control · Autonomous truck · Automation in mining sites · Cooperative

freight transportation · Adaptive cruise control · Vehicular chain

Introduction

In the recent years, we witnessed an increasing interest in the development of autonomous driving technologies for freight transportation. Initial efforts in driverless technologies focused on passenger cars thanks to their higher diffusion. Pioneering research carried out in the 1990s and early 2000s showed how automated vehicles can be a reality in the near future (Horowitz and Varaiya 2000; Broggi et al. 1999). Among research efforts, the demonstration within the PATH project of eight driverless vehicles driving at a tight platoon formation in 1997 (Shladover 2007) and the fully autonomous 60-mile drive in urban environment in the 2007 DARPA Grand Challenge (Buehler et al. 2007) are considered milestones in vehicle automation. These results encouraged original equipment manufacturers (OEMs), component suppliers, and technology start-ups to invest in driverless technology research. Thanks to highly viable business cases offered by the automation of freight transportation, a large research effort focused on the development of automated truck technologies.

Unlike the passenger car industry that is human-centered and more adverse to trends, the freight transportation is conservative and almost exclusively driven by operational efficiency. That means that any technology that can reduce the truck operational cost for even a few percent is able to attract significant attention and investments. Figure 1 shows some examples of relevant applications of truck automation currently under investigation by the industry. Figure 1a shows two long-haulage trucks of different brands driving in a platoon formation thanks to vehicle-to-vehicle communication and shared protocols for longitudinal dynamics automation. Due to the large cross-sectional area of trucks, the short inter-vehicular distance creates a slipstream effect that effectively reduces the overall energy consumption and

the greenhouse gas emissions of the platooning vehicles. Figure 1b presents an autonomous cabin-less vehicle hauling a container in a harbor freight terminal. A freight terminal is an example of a close environment where the site management system has an extensive awareness and control on what is happening. This drastically reduces the number of scenarios that automated vehicles have to face, compared to driving on public road. Furthermore, as these sites are typically closed, vehicle automation can be deployed without waiting for regulation for public road. Two other examples of autonomous trucks operating in controlled environments are

displayed in Fig. 1c, d. Figure 1c shows driverless haulers cooperating to transport ore and waste material across a mining site, while Fig. 1d shows the operation of two coordinated autonomous trucks removing snow from an airport runway.

Automating driving in truck operations encourages the optimization of the entire stack of logistic transportation. A glimpse of a possible freight transportation future is sketched in Fig. 2. Here, the fleet management system (FMS) portrayed by the watchtower governs and oversees the freight vehicles in the area. It has full knowledge of the requested flows of goods, their deadlines, and their dependencies. In order



(a) Multi-brand truck platoon driving on public road.



(b) Self-driving hauler operating in a freight terminal.



(c) Driverless haulers operating in an open-pit mining site.



(d) Autonomous snow removal trucks operating in an airport.

Automated Truck Driving, Fig. 1 Examples of applications of automated trucks. (Images courtesy of Volvo Trucks, Scania, and Daimler) (a) Multi-brand truck platoon driving on public road (b) Self-driving hauler oper-

ating in a freight terminal (c) Driverless haulers operating in an open-pit mining site (d) Autonomous snow removal trucks operating in an airport



Automated Truck Driving, Fig. 2 Glimpse of a possible future of automated freight transportation. (Image courtesy of Scania)

to cope with the requests, it creates transport missions and assigns them to the automated trucks. As trucks can greatly vary in type of transported goods and maximum load, the correct dispatch of vehicles can substantially improve the transportation efficiency. The system can assign large flows of goods with the same origin and destination to multiple trucks driving in platoon formation or promote the creation of platoons between trucks driving along the same road to further improve efficiency. Not least, the FMS dispatch solution should be robust to possible delay and adapt when live updates are received.

SAE Levels of Automation Applied to Trucks

In 2015, SAE published the J3016 standard (SAE International 2018) that defines six levels of vehicle automation. The standard applies to all on-road vehicles, and its main intent was to clearly separate the responsibility of the driver versus the automated system. Although other vehicle autonomy classifications have been proposed, SAE J3016 has been widely adopted by industry and, nowadays, is considered the de facto standard for vehicle automation. The six levels of automation are summarized in Table 1, and their application in the truck industry is discussed below.

Level 0: No Automation

At level 0, the driver is fully in control of the vehicle. His/her driving experience can be enhanced by passive advanced driver assistance systems (ADAS) that monitor the environment and trigger audio and visual warnings, e.g., lane departure, forward collision, and blind spot detection. Such systems are nowadays common in commercial trucks.

Level 1: Driver Assistance

At level 1, the driver is assisted by an active ADAS that acts either on the steering or on the acceleration/deceleration. The driver, however, is expected to monitor the automated system at all times and overrule it if needed. The driver is ultimately responsible for the driving task.

Safety-related active ADAS, such as automatic emergency braking, adaptive cruise control, and lane keeping assist, has proven capable of significantly reducing the number of road crashes (IIHS-HLDI 2019). In the truck industry, active ADAS that relies on vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications is also deployed to improve efficiency and safety.

Predictive cruise controllers based on V2I communication allow to exploit information on the road ahead, e.g., road topography, traffic condition, or traffic light timing, to adapt the

Automated Truck Driving, Table 1 SAE levels of automation

SAE level	Description
Level 0 No automation	The driver performs all driving tasks
Level 1 Driver assistance	The driver is assisted by an autonomous driving feature acting on either steering or acceleration/deceleration
Level 2 Partial automation	The driver is supported by one or more driving assistance features acting on both the steering and acceleration/deceleration. The driver is expected to monitor at all times the automated system
Level 3 Conditional automation	Under certain conditions, all driving tasks are automated. The driver can divert the attention temporally but should be ready to take control of the vehicle in a few seconds when the system requires it
Level 4 High automation	Under certain conditions, all driving tasks are automated. Under all conditions, the system is able to take the vehicle to a minimal risk condition
Level 5 Full automation	All driving tasks are automated under all conditions

A

current speed and reduce inefficient braking. Topography-aware predictive cruise controllers are commercially available (Daimler 2020; VolvoTrucks 2020), while cruise controllers that exploit traffic condition and traffic light timing are still subjects of research (Henriksson et al. 2017; Cicic and Johansson 2018).

V2V communication allows adjacent vehicles to share their current state and future intention, enabling safe and efficient operation of truck platooning through longitudinal dynamics automation. An example of a three-truck level 1 platoon is depicted in Fig. 3. Due to the large cross-sectional area of trucks, the short inter-vehicular distance creates a slipstream effect that translates into a reduction of energy consumption and greenhouse gas emissions of the order of 10% (Browand et al. 2004; Roeth 2013; Alam et al. 2015). In long-haulage transportation, this reduction has a notable impact on the overall operational cost and has therefore attracted major investments in OEMs and start-ups, e.g., Ensemble (2020) and Peloton (2020).

Level 2: Partial Automation

The driver is assisted by one or more ADAS that acts on both the steering and acceleration/deceleration in level 3 automation. As in

level 1 automation, however, the driver should monitor the automated systems at all times and be ready to overrule them if needed. The driver is ultimately responsible for the driving task.

Level 2 automation allows the concurrent use of multiple ADAS. In truck platooning, it allows the simultaneous automated control of the longitudinal and lateral dynamics, enabling tighter inter-vehicular spacing (Engström et al. 2019).

Level 3: Conditional Automation

With level 3 automation, the vehicle is fully controlled by the automated system under certain conditions. Such conditions define the so-called operational design domain (ODD) and may include geographic limitations, environmental conditions (e.g., weather and daytime/nighttime), and vehicle status (e.g., driving alone or in platoon formation). Although the driver can diverge his/her attention to other tasks, he/she needs to be ready to take back the control of the vehicles in a few seconds when the system requires it and is ultimately responsible for the driving task.

In the truck industry, level 3 automation would allow the driver to conduct other tasks valuable for the business while sitting in the driver seat. However, level 3 automation is highly contro-

Automated Truck Driving, Fig. 3 Three long-haulage trucks using level 1 truck platooning functionality. (Image courtesy of Scania)



versial because, although the driver can engage in other tasks, he/she is still fully responsible if an emergency situation occurs. For this reasons, many truck manufacturers, despite having the technology, prefer to skip the commercialization of level 3 automation and focus instead on level 4 (TTNews 2020).

Level 4: High Automation

At level 4, the vehicle is fully automated under certain conditions defined by the ODD. If the ODD conditions are no more satisfied, the automated system is able to take the vehicle to a minimal risk state without human intervention. When the vehicle is operating in the autonomous mode, the automated system is fully responsible for the driving task. A human driver is therefore no more required in the vehicle, and if present, he/she can be fully engaged in other tasks. The latter clause is a game changer in vehicle automation and enables promising and highly valuable business cases. Furthermore, it also allows to remove humans from remote and dangerous sites (e.g., underground mining sites) and to respond to the increasing shortage of truck drivers in some parts of the world.

Level 4 automation is relatively broad, and its degree of complexity is highly dependent on the specified ODD. ODD can, for example, restrict the automation to certain locations or roads. Geofencing allows the use of highly detailed maps that may include visual waypoints,

road signs, and road works. These maps can facilitate environmental perception and path planning and, overall, simplify the automated system implementation. Promising use cases of geo-restricted level 4 automation are trucks operating in controlled environments, such as freight terminals, mine sites, and airports, as illustrated in Fig. 1. Another appealing use case is the restriction of level 4 automation to highways, which represent an environment with clear rules and a relatively low number of possible scenarios. Highway automation represents a strong business case that motivated the creation of new start-ups, e.g., Embark (2020). Level 4 platooning can represent one of the first implementations of driverless vehicles on public roads; see test scenarios in Singapore (2020).

Level 5: Full Automation

At level 5, all driving tasks are automated under all circumstances. The automated system is able to fully replace the human driver and, when activated, is fully responsible for the driving task.

Level 5 automation represents a major step from level 4 as the automated system should be able to handle any unforeseen event. Furthermore, as geofencing is not part of level 5 automation, it is probably impossible to make use of highly detailed environment maps. Because of its complexity and limited gain, level 5 automation is generally considered of limited interest currently in the truck industry.

Platoon-Based Cooperative Freight Transportation

In this section, we discuss in some detail the use case of cooperative freight transportation centered on platoons, and we discuss a multilayer control architecture to back it up.

Figure 3 shows three long-haulage trucks driving on a highway in platoon formation. As we have seen, platooning is capable of reducing truck energy consumption and greenhouse gas emissions of the order of 10%. More advanced platooning functionalities (level 4 platooning) would additionally allow to allocate the driver time on other tasks or to operate follower trucks driverless, resulting in significant economical benefits for transportation companies. Platoon-capable trucks, however, will always constitute only a fraction of the overall traffic, and their locations will therefore be relatively sparsely distributed over the road network. To back up the efficient exploitation of truck platooning in freight transportation, truck cooperation will play an essential role.

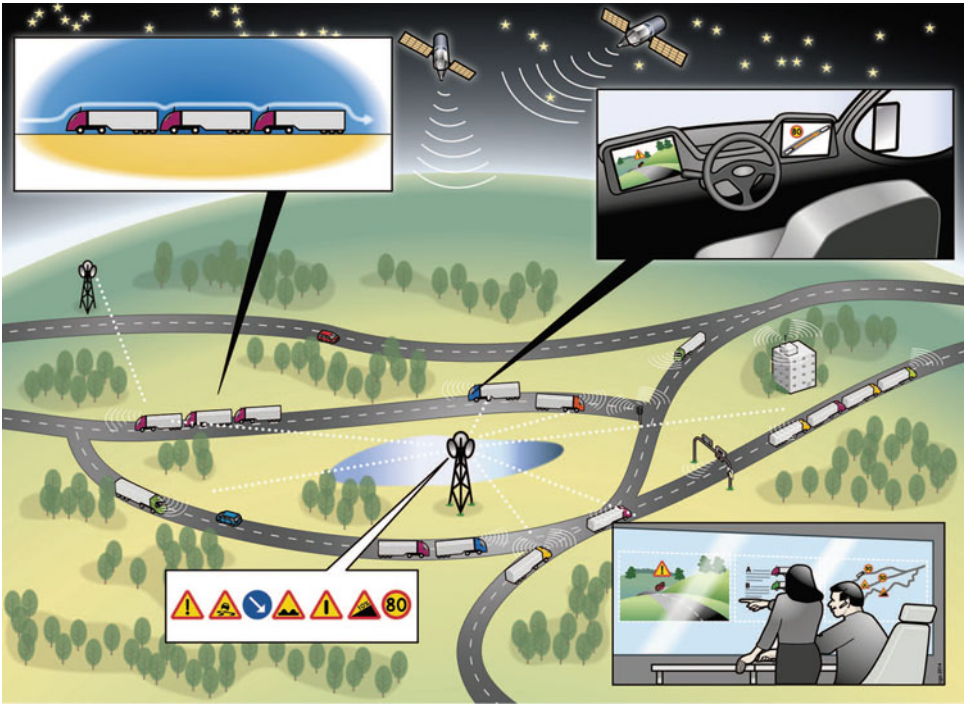
Figure 4 gives a synopsis of how a possible future of cooperative freight transportation based on platooning can look like. The figure shows not only multiple platoons but also trucks driving alone. Trucks generally have different origins; destinations; deadlines; platooning functionalities; and, if a human driver is present, rest constraints. To meet and form platoons, trucks need therefore to adjust their speed or stop at roadside stations. However, these actions cannot be conducted naively as short-sighted decisions can result in higher operation costs for the transportation companies. To cope with the matter, a central office displayed in the bottom right corner of the figure has a complete overview of truck missions and promotes the formation of platoons that are most beneficial for the overall freight transportation.

To support cooperative freight transportation, the multilayer control architecture in Fig. 5 is natural (Van De Hoef et al. 2019). Such architecture is composed of four layers that aim to provide increasingly higher levels of abstraction.

The lowest layer, the so-called operational layer, is where the vehicle controllers reside. Vehicle controllers are distributed in the platooning vehicles and, thanks to V2V communication and local sensors, guarantee the safe and efficient operation of platoons. Their goal is to track the reference speed and inter-vehicular distances by commanding the engine actuators, the brake systems, and the gearbox of platooning vehicles. One step above, in the tactical layer, resides the platoon manager. The platoon manager controls the formation, splitting, reorganization, and operation of platoons. It does so by defining the reference speed and inter-vehicular distances tracked by the vehicle controllers. The platoon manager typically runs in one of the trucks or in an off-board location. It can exploit information about the road ahead, e.g., road topography and traffic condition, to adjust the platoon reference speed and inter-vehicular distances to improve the efficiency (Turri et al. 2017; Cicic and Johansson 2018). One layer up, we find the strategic layer where the platoon coordinator resides. The platoon coordinator has a central role in the truck coordination task and defines which trucks should adjust their speed or momentarily stop in order to platoon together. This is a complex task, and both centralized and decentralized solutions have been proposed, e.g., van de Hoef et al. (2015) and Johansson et al. (2020). Finally, in the service layer, the transport management system (TMS) and the fleet management system (FMS) reside. This layer is responsible for assigning goods flows to vehicles and drivers, ranging from manual planning to complex supply chain optimization. A similar architecture is the base of the Ensemble project that aims at achieving platooning protocols between the major European truck manufacturers (Ensemble 2020).

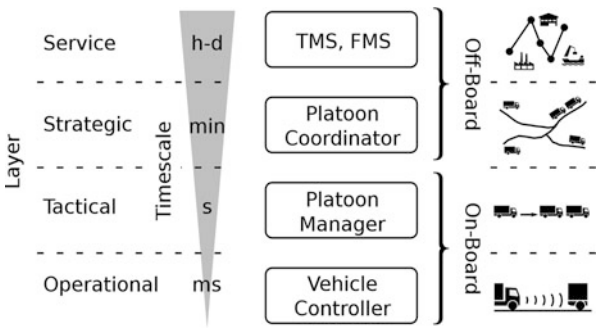
Driverless Transportation in Mining Sites

Controlled environments are locations where we could expect early deployment of fully driverless vehicles. In this section, we discuss the use case



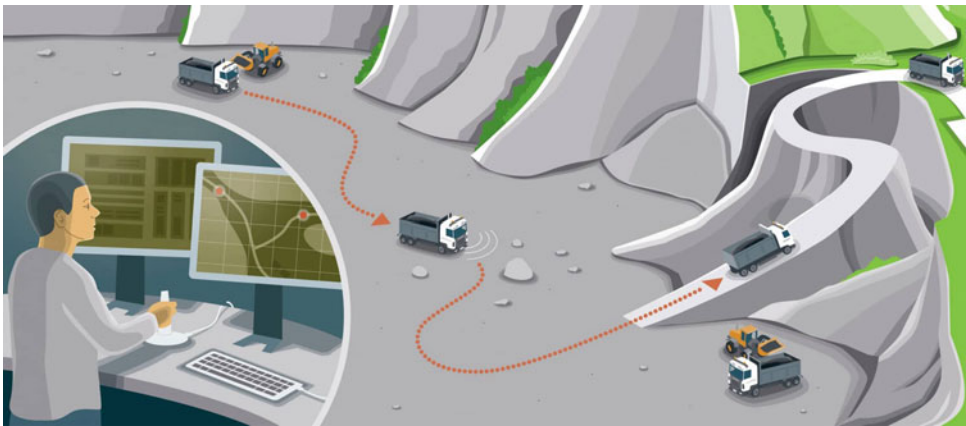
Automated Truck Driving, Fig. 4 Future scenario of cooperative freight transportation centered on truck platooning

Automated Truck Driving, Fig. 5 Control architecture supporting platoon-based cooperative freight transportation (Van De Hoef et al. 2019)



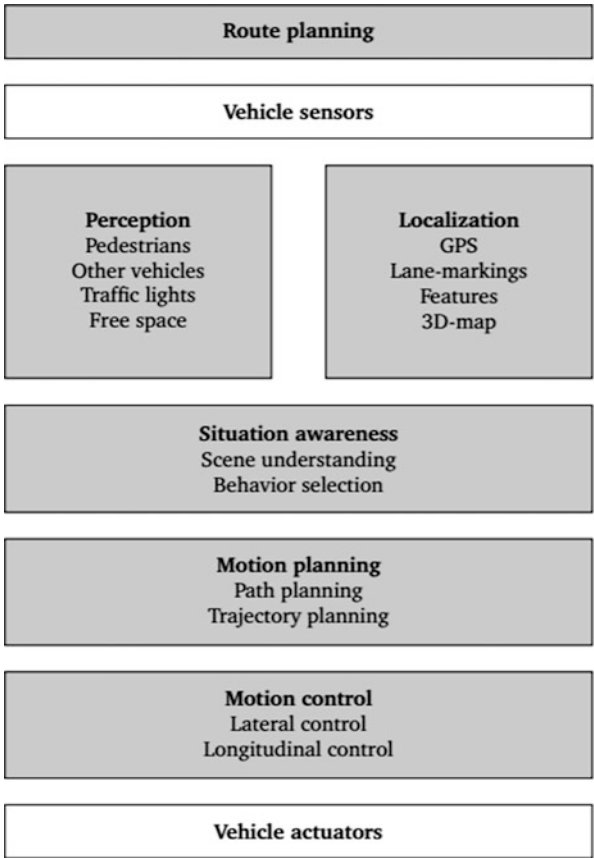
Automated Truck Driving, Fig. 6 Autonomous testing truck for mining operations. (Image courtesy of Scania)





Automated Truck Driving, Fig. 7 A possible scenario of driverless truck operating in a mining site. (Image courtesy of Scania)

Automated Truck Driving, Fig. 8 Functional architecture for autonomous driving (Lima 2018)



of driverless transportation in one such environment, namely, a mining site. Mining sites require a considerable transfer of materials, such as ore and waste, on distances that can stretch up to

hundreds of kilometers. Nowadays, the transportation operations in mining sites are carried out by human drivers. Their job is considered repetitive and dangerous, and due to their

often remote location, the sector experiences a widespread shortage of drivers. Removing human drivers in mines is therefore a quite attractive quest that can result in significant economical and safety benefits.

Figure 6 shows a test vehicle for driverless transportation in mining sites. The vehicle is equipped with short-range radars at each corner for 360-degree obstacle detection, long-range radar to enable high-speed driving, multi-lens cameras to identify objects, and inertial measurement unit (IMU) for acceleration measurements. A possible scenario where such vehicle can operate is illustrated in Fig. 7. Here, we see a truck autonomously driving from a pickup location to an off-load location. This truck does not need a driver onboard as its operation can be monitored by a remote operator working at an off-site office.

Figure 8 shows a functional architecture for a self-driving vehicle system (Lima 2018). The architecture integrates a variety of sensors, which through sensor fusion provides situational awareness to the driverless vehicle, enabling the safe planning and control of the vehicle motion. The route planning specifies the mission for the driverless vehicle, e.g., going from the pickup to the drop-off location. While travelling between these two points, the vehicle needs to detect objects and unexpected obstacles. This task is fulfilled by the perception functionality and is enabled, in its turn, by a multitude of vehicle sensors. A typical set of sensors for autonomous vehicles include (i) radars that measure distances to obstacles in specific directions, (ii) one or more lidars that provide a 3D map of the environment, (iii) an inertial measurement unit (IMU) that measures linear and angular accelerations, and (iv) monocular and stereo cameras that recognize objects and estimate their distance. The perception functionality fuses all the sensor information and returns a labelled map of the environment. Here, labels specify the particular object or obstacle in the map, e.g., site walls, road signs, humans, and other vehicles. The labelled map is then processed by the situational awareness functionality that, by exploiting GPS measures and detailed off-

board maps, understands the scene and selects the proper behavior to follow, e.g., proceed normally, stand still, or circumnavigate obstacle. The selected behavior is communicated to the motion planner that computes the reference vehicle trajectory accordingly. Finally, the *motion controller* governs the actuators, i.e., powertrain, brakes, and steering, while tracking the reference trajectory. The primary control system is often backed up by a secondary system able to operate the vehicle even in the case of partially faulty hardware. If supported, it can take the vehicle to a state of minimal risk, e.g., to a standstill position on the side of the road.

Cross-References

- Adaptive Cruise Control
- Lane Keeping Systems
- Vehicle Dynamics Control
- Vehicular Chains

Bibliography

- Alam A, Besselink B, Turri V, Mårtensson J, Johansson KH (2015) Heavy-duty vehicle platooning for sustainable freight transportation: a cooperative method to enhance safety and efficiency. *IEEE Control Syst Mag* 35(6):34–56. <https://doi.org/10.1109/MCS.2015.2471046>
- Broggi A, Bertozzi M, Fascioli A, Bianco CGL, Piazzi A (1999) The ARGO autonomous vehicle's vision and control systems. *Int J Intell Control Syst* 3(4):409–441
- Browand F, Mc Arthur J, Radovich C (2004) Fuel saving achieved in the field test of two tandem trucks. Technical report June, Institute of Transportation Studies, University of California's Berkeley, Berkeley
- Buehler M, Iagnemma K, Singh S (2007) The 2005 DARPA grand challenge: the great robot race. Springer, Berlin
- Cicic M, Johansson KH (2018) Traffic regulation via individually controlled automated vehicles: a cell transmission model approach. In: *IEEE Conference on Intelligent Transportation Systems, Proceedings*, pp 766–771. <https://doi.org/10.1109/ITSC.2018.8569960>
- Daimler (2020) Predictive powertrain control – clever cruise control helps save fuel. <https://media.daimler.com/marsMediaSite/en/instance/ko/Predictive-Powertrain-Control—Clever-cruise-control-helps-save-fuel.xhtml?oid=9917205>. Accessed 04 Apr 2020

- Embark (2020) Embark homepage. <https://embarktrucks.com/>. Accessed 04 Apr 2020
- Engström J, Bishop R, Shladover SE, Murphy MC, O'Rourke L, Voegt T, Denaro B, Demato R, Demato D (2019) Deployment of automated trucking: challenges and opportunities. In: Meyer G, Beiker S (eds) Road vehicle automation 5. Springer, Berlin
- Ensemble (2020) Ensemble homepage. <https://platooningensemble.eu/>. Accessed 04 Apr 2020
- Henriksson M, Flärdh O, Mårtensson J (2017) Optimal speed trajectories under variations in the driving corridor. IFAC-PapersOnLine 50(1):12551–12556. <https://doi.org/10.1016/j.ifacol.2017.08.2194>
- Horowitz R, Varaiya P (2000) Control design of an automated highway system. Proc IEEE 88(7):913–925. <https://doi.org/10.1109/5.871301>
- IIHS-HLDI (2019) Real-world benefits of crash avoidance technologies. Technical report, Arlington
- Johansson A, Turri V, Nekouei E, Johansson KH, Mårtensson J (2020) Truck platoon formation at hubs: an optimal release time rule. In: World Congress IFAC, Berl
- Lima PF (2018) Optimization-based motion planning and model predictive control for autonomous driving. Ph.D. thesis, KTH – Royal Institute of Technology
- Peloton (2020) Peloton homepage. <https://peloton-tech.com/>. Accessed 04 Apr 2020
- Roeth M (2013) CR England peloton technology platooning test Nov 2013. Technical report, North American Council for Freight Efficiency, Fort Wayne
- SAE International (2018) J3016 – surface vehicle recommended practice – taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. Technical report
- Shladover S (2007) PATH at 20–history and major milestones. IEEE Trans Intell Transp Syst 8(4):584–592. <https://doi.org/10.1109/TITS.2007.903052>
- Singapore (2020) Singapore to start truck platooning trials. <https://www.mot.gov.sg/news-centre/news/Detail/Singapore>. Accessed 04 Apr 2020
- TTNews (2020) Transport topics – why truck makers are skipping level 3. <https://www.ttnews.com/articles/why-truck-makers-are-skipping-level-3>. Accessed 04 Apr 2020
- Turri V, Besselink B, Johansson KH (2017) Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning. IEEE Trans Control Syst Technol 25(1):12–28
- van de Hoef S, Johansson KH, Dimarogonas DV (2015) Fuel-optimal centralized coordination of truck platooning based on shortest paths. In: Proceedings of the IEEE American Control Conference, Chicago, pp 3740–3745
- Van De Hoef S, Mårtensson J, Dimarogonas DV, Johansson KH (2019) A predictive framework for dynamic heavy-duty vehicle platoon coordination. ACM Trans Cyber-Phys Syst 4(1). <https://doi.org/10.1145/3299110>
- VolvoTrucks (2020) Predictive cruise control – I-See. <https://www.volvotrucks.us/trucks/i-shift-transmission/i-see/>. Accessed 04 Apr 2020

Autotuning

Tore Hägglund

Lund University, Lund, Sweden

A

Abstract

Autotuning, or automatic tuning, means that the controller is tuned automatically. Autotuning is normally applied to PID controllers, but the technique can also be used to initialize more advanced controllers. The main approaches to autotuning are based on step response analysis or frequency response analysis obtained using relay feedback. Autotuning has been well received in industry, and today most distributed control systems have some kind of autotuning technique.

Keywords

Automatic tuning · Gain scheduling · PID control · Process control · Proportional-integral-derivative control · Relay feedback

Background

In the late 1970s and early 1980s, there was a quite rapid change of controller implementation in process control. The analog controllers were replaced by computer-based controllers and distributed control systems. The functionality of the new controllers was often more or less a copy of the old analog equipment, but new functions that utilized the computer implementation were gradually introduced. One of the first functions of this type was autotuning. Autotuning is a method to tune the controllers, normally PID controllers, automatically.

What Is Autotuning?

A PID controller in its basic form has the structure

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d}{dt} e(t) \right),$$

where u is the controller output and $e = y_{sp} - y$ is the control error, where y_{sp} is the setpoint and y is the process output. There are three parameters in the controller, gain K , integral time T_i , and derivative time T_d . These parameters have to be set by the user. Their values are dependent of the process dynamics and the specifications of the control loop.

A process control plant may have thousands of control loops, which means that maintaining high-performance controller tuning can be very time consuming. This was the main reason why procedures for automatic tuning were installed so rapidly in the computer-based controllers.

When a controller is to be tuned, the following steps are normally performed by the user:

1. To determine the process dynamics, a minor disturbance is injected by changing the control signal.
2. By studying the response in the process output, the process dynamics can be determined, i.e., a process model is derived.
3. The controller parameters are finally determined based on the process model and the specifications.

Autotuning means simply that these three steps are performed automatically. Instead of having a human to perform these tasks, they are performed automatically on demand from the user. Ideally, the autotuning should be fully automatic, which means that no information about the process dynamics is required from the user.

Automatic tuning can be performed in many ways. The process disturbance can take different forms, e.g., in the form of step changes or some kind of oscillatory excitation. The model obtained can be more or less accurate. There are also many ways to tune the controller based on the process model.

Here, we will discuss two main approaches for autotuning, namely, those that are based on step response analysis and those that are based on frequency response analysis.

Methods Based on Step Response Analysis

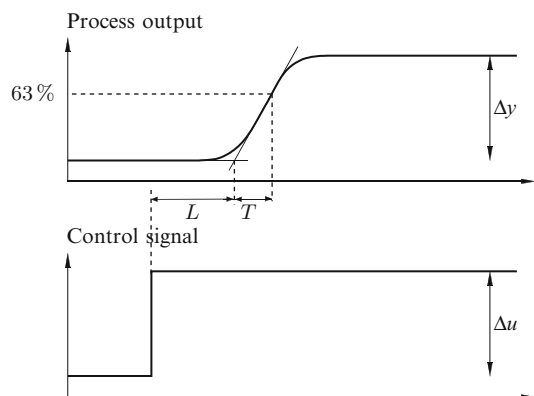
Most methods for automatic tuning of PID controllers are based on step response analysis. When the operator wishes to tune the controller, an open-loop step response experiment is performed. A process model is then obtained from the step response, and controller parameters are determined. This is usually done using simple formulas or look-up tables.

The most common process model used for PID controller tuning based on step response experiments is the first-order plus dead-time model

$$G(s) = \frac{K_p}{1 + sT} e^{-sL}$$

where K_p is the static gain, T is the apparent time constant, and L is the apparent dead time. These three parameters can be obtained from a step response experiment according to Fig. 1.

Static gain K_p is given by the ratio between the steady-state change in process output and the magnitude of the control signal step,



Autotuning, Fig. 1 Determination of K_p , L , and T from a step response experiment

$K_p = \Delta y / \Delta u$. Dead-time L is determined from the time elapsed from the step change to the intersection of the largest slope of the process output with the level of the process output before the step change. Finally, time constant T is the time when the process output has reached 63 % of its final value, subtracted by L .

The greatest difficulty in carrying out tuning automatically is in selecting the amplitude of the step. The user naturally wants the disturbance to be as small as possible so that the process is not disturbed more than necessary. On the other hand, it is easier to determine the process model if the disturbance is large. The result of this dilemma is usually that the user has to decide how large the step in the control signal should be. Another problem is to determine when the step response has reached its final value.

Methods Based on Frequency Response Analysis

Frequency-domain characteristics of the process can be obtained by adding sinusoids to the control signal, but without knowing the frequency response of the process, the interesting frequency range and acceptable amplitudes are not known. A method that automatically provides a relevant frequency response can be determined from experiments with relay feedback according to Fig. 2. Notice that there is a switch that selects either relay feedback or ordinary PID feedback. When it is desired to tune the system, the PID function is disconnected and the system is connected to relay feedback control. Relay feedback control is the same as on/off control, but where the on and off levels are carefully chosen and not

0 and 100 %. The relay feedback makes the control loop oscillate. The period and the amplitude of the oscillation is determined when steady-state oscillation is obtained. This gives the ultimate period and the ultimate gain. The parameters of a PID controller can then be determined from these values. The PID controller is then automatically switched in again, and the control is executed with the new PID parameters.

For large classes of processes, relay feedback gives an oscillation with period close to the ultimate frequency ω_u , as shown in Fig. 3, where the control signal is a square wave and the process output is close to a sinusoid. The gain of the transfer function at this frequency is also easy to obtain from amplitude measurements.

Describing function analysis can be used to determine the process characteristics. The describing function of a relay with hysteresis is

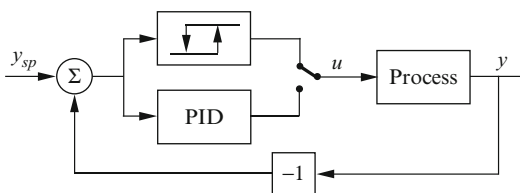
$$N(a) = \frac{4d}{\pi a} \left(\sqrt{1 - \left(\frac{\epsilon}{a}\right)^2} - i \frac{\epsilon}{a} \right)$$

where d is the relay amplitude, ϵ the relay hysteresis, and a the amplitude of the input signal. The negative inverse of this describing function is a straight line parallel to the real axis; see Fig. 4. The oscillation corresponds to the point where the negative inverse describing function crosses the Nyquist curve of the process, i.e., where

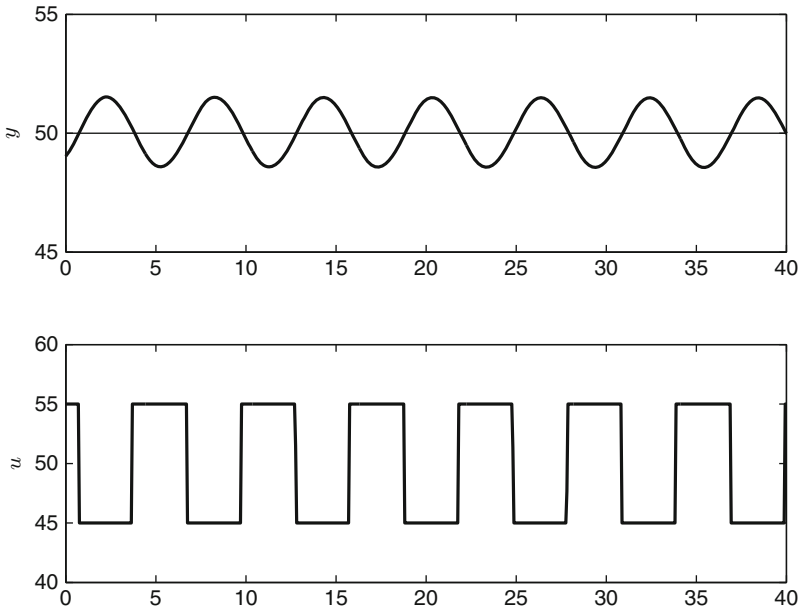
$$G(i\omega) = -\frac{1}{N(a)}$$

Since $N(a)$ is known, $G(i\omega)$ can be determined from the amplitude a and the frequency ω of the oscillation.

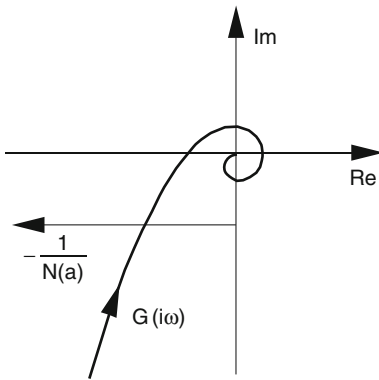
Notice that the relay experiment is easily automated. There is often an initialization phase where the noise level in the process output is determined during a short period of time. The noise level is used to determine the relay hysteresis and a desired oscillation amplitude in the process output. After this initialization phase, the relay function is introduced. Since the amplitude of the oscillation is proportional to the relay output, it is easy to control it by adjusting the relay output.



Autotuning, Fig. 2 The relay autotuner. In the tuning mode the process is connected to relay feedback



Autotuning, Fig. 3 Process output y and control signal u during relay feedback



Autotuning, Fig. 4 The negative inverse describing function of a relay with hysteresis $-1/N(a)$ and a Nyquist curve $G(i\omega)$

Different Adaptive Techniques

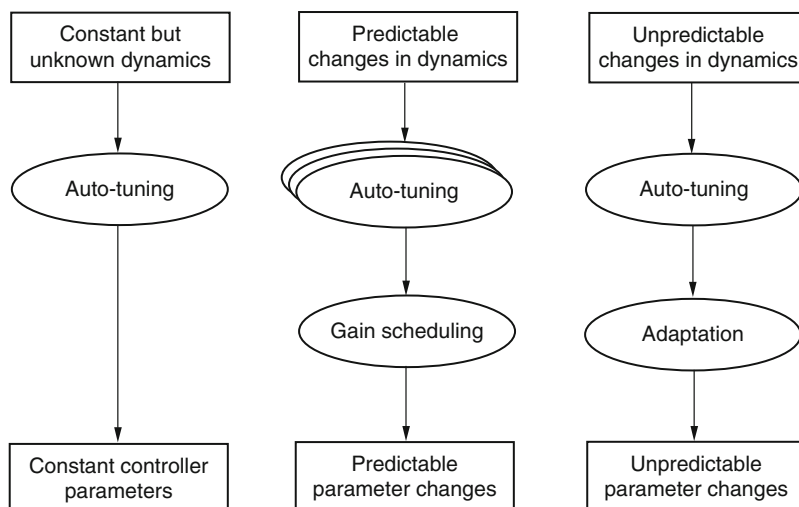
In the late 1970s, at the same time as autotuning procedures were developed and implemented in industrial controllers, there was a large academic interest in adaptive control. These two concepts are often mixed up with each other. Autotuning is sometimes called tuning on demand. An identification experiment is performed, controller

parameters are determined, and the controller is then run with fixed parameters. An adaptive controller is, however, a controller where the controller parameters are adjusted online based on information from routine data. Automatic tuning and adaptive control have, however, one thing in common, namely, that they are methods to adapt the controller parameters to the actual process dynamics. Therefore, they are both called *adaptive techniques*.

There is a third adaptive technique, namely, gain scheduling. Gain scheduling is a system where controller parameters are changed depending on measured operating conditions. The scheduling variable can, for instance, be the measurement signal, controller output, or an external signal. For historical reasons the word gain scheduling is used even if other parameters like integral time or derivative time are changed. Gain scheduling is a very effective way of controlling systems whose dynamics change with the operating conditions. Automatic tuning has made it possible to generate gain schedules automatically.

Although research on adaptive techniques has almost exclusively focused on adaptation,

Autotuning, Fig. 5 When to use different adaptive techniques



experience has shown that autotuning and gain scheduling have much wider industrial applicability. Figure 2 illustrates the appropriate use of the different techniques.

Controller performance is the first issue to consider. If requirements are modest, a controller with constant parameters and conservative tuning can be used. Other solutions should be considered when higher performance is required.

If the process dynamics are constant, a controller with constant parameters should be used. The parameters of the controller can be obtained by autotuning.

If the process dynamics or the character of the disturbances are changing, it is useful to compensate for these changes by changing the controller. If the variations can be predicted from measured signals, gain scheduling should be used since it is simpler and gives superior and more robust performance than continuous adaptation. Typical examples are variations caused by nonlinearities in the control loop. Autotuning can be used to build up the gain schedules automatically.

There are also cases where the variations in process dynamics are not predictable. Typical examples are changes due to unmeasurable variations in raw material, wear, fouling, etc. These variations cannot be handled by gain scheduling but must be dealt with by adaptation. An autotuning procedure is often used to initialize the

adaptive controller. It is then sometimes called pre-tuning or initial tuning.

To summarize, autotuning is a key component in all adaptive techniques and a prerequisite for their use in practice.

Industrial Products

Commercial PID controllers with adaptive techniques have been available since the beginning of the late 1970s, both in single-station controllers and in distributed control systems.

Two important, but distinct, applications of PID autotuners are *temperature controllers* and *process controllers*. Temperature controllers are primarily designed for temperature control, whereas process controllers are supposed to work for a wide range of control loops in the process industry such as flow, pressure, level, temperature, and concentration control loops. Automatic tuning is easier to implement in temperature controllers, since most temperature control loops have several common features. This is the main reason why automatic tuning was introduced more rapidly in these controllers.

Since the processes that are controlled with process controllers may have large differences in their dynamics, tuning becomes more difficult compared to the pure temperature control loops.

Automatic tuning can also be performed by external devices which are connected to the

control loop during the tuning phase. Since these devices are supposed to work together with controllers from different manufacturers, they must be provided with quite a lot of information about the controller structure and parameterization in order to provide appropriate controller parameters. Such information includes signal ranges, controller structure (series or parallel form), sampling rate, filter time constants, and units of the different controller parameters (gain or proportional band, minutes or seconds, time or repeats/time).

Summary and Future Directions

Most of the autotuning methods that are available in industrial products today were developed about 30 years ago, when computer-based controllers started to appear. These autotuners are often based on simple models and simple tuning rules. With the computer power available today, and the increased knowledge about PID controller design, there is a potential for improving the autotuners, and more efficient autotuners will probably appear in industrial products quite soon.

Cross-References

- [Adaptive Control: Overview](#)
- [PID Control](#)

Bibliography

- Åström KJ, Hägglund T (1995) PID controllers: theory, design, and tuning. ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park
- Åström KJ, Hägglund T (2005) Advanced PID control. ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park
- Vilanova R, Visioli A (eds) (2012) PID control in the third millennium. Springer, Dordrecht
- Visioli A (2006) Practical PID control. Springer, London
- Yu C-C (2006) Autotuning of PID controllers – a relay feedback approach. Springer, London

Averaging Algorithms and Consensus

Wei Ren

Department of Electrical Engineering,
University of California, Riverside, CA, USA

Abstract

In this article, we overview averaging algorithms and consensus in the context of distributed coordination and control of networked systems. The two subjects are closely related but not identical. Distributed consensus means that a team of agents reaches an agreement on certain variables of interest by interacting with their neighbors. Distributed averaging aims at computing the average of certain variables of interest among multiple agents by local communication. Hence averaging can be treated as a special case of consensus – average consensus. For distributed consensus, we introduce distributed algorithms for agents with single-integrator, general linear, and nonlinear dynamics. For distributed averaging, we introduce static and dynamic averaging algorithms. The former is useful for computing the average of initial conditions (or constant signals), while the latter is useful for computing the average of time-varying signals. Future research directions are also discussed.

Keywords

Cooperative control · Coordination ·
Distributed control · Multi-agent systems ·
Networked systems

Introduction

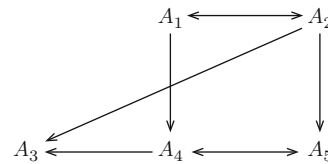
In the area of control of networked systems, low cost, high adaptivity and scalability, great robustness, and easy maintenance are critical factors. To achieve these factors, distributed coordination and control algorithms that rely on only local interaction between neighboring agents to

achieve collective group behavior are more favorable than centralized ones. In this article, we overview averaging algorithms and consensus in the context of distributed coordination and control of networked systems.

Distributed consensus means that a team of agents reaches an agreement on certain variables of interest by interacting with their neighbors. A consensus algorithm is an update law that drives the variables of interest of all agents in the network to converge to a common value (Jadbabaie et al. 2003; Olfati-Saber et al. 2007; Ren and Beard 2008). Examples of the variables of interest include a local representation of the center and shape of a formation, the rendezvous time, the length of a perimeter being monitored, the direction of motion for a multi-vehicle swarm, and the probability that a target has been identified. Consensus algorithms have applications in rendezvous, formation control, flocking, attitude alignment, and sensor networks (Bullo et al. 2009; Qu 2009; Mesbahi and Egerstedt 2010; Ren and Cao 2011; Bai et al. 2011a). Distributed averaging algorithms aim at computing the average of certain variables of interest among multiple agents by local communication. Distributed averaging finds applications in distributed computing, distributed signal processing, and distributed optimization (Tsitsiklis et al. 1986). Hence the variables of interest are dependent on the applications (e.g., a sensor measurement or a network quantity). Consensus and averaging algorithms are closely connected and yet nonidentical. When all agents are able to compute the average, they essentially reach a consensus, the so-called average consensus. On the other hand, when the agents reach a consensus, the consensus value might or might not be the average value.

Graph Theory Notations. Suppose that there are n agents in a network. A *network topology* (equivalently, *graph*) G consisting of a node set $\mathcal{V} = \{1, \dots, n\}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ will be used to model *interaction* (communication or sensing) between the n agents. An *edge* (i, j) in a *directed graph* denotes that agent j can obtain

information from agent i , but not necessarily vice versa. In contrast, an edge (i, j) in an *undirected graph* denotes that agents i and j can obtain information from each other. Agent j is a (*in-*)*neighbor* of agent i if $(j, i) \in \mathcal{E}$. Let $\mathcal{N}_i$ denote the neighbor set of agent i . We assume that $i \in \mathcal{N}_i$. A *directed path* is a sequence of edges in a directed graph of the form $(i_1, i_2), (i_2, i_3), \dots$, where $i_j \in \mathcal{V}$. An *undirected path* in an undirected graph is defined analogously. A directed graph is *strongly connected* if there is a directed path from every agent to every other agent. An undirected graph is *connected* if there is an undirected path between every pair of distinct agents. A directed graph *has a directed spanning tree* if there exists at least one agent that has directed paths to all other agents. For example, Fig. 1 shows a directed graph that has a directed spanning tree but is not strongly connected. The *adjacency matrix* $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n \times n}$ associated with G is defined such that a_{ij} (the weight of edge (j, i)) is positive if agent j is a neighbor of agent i while $a_{ij} = 0$ otherwise. The (nonsymmetric) *Laplacian matrix* (Agaev and Chebotarev 2005) $\mathcal{L} = [\ell_{ij}] \in \mathbb{R}^{n \times n}$ associated with $\mathcal{A}$ and hence G is defined as $\ell_{ii} = \sum_{j \neq i} a_{ij}$ and $\ell_{ij} = -a_{ij}$ for all $i \neq j$. For an undirected graph, we assume that $a_{ij} = a_{ji}$. A graph is *balanced* if for every agent the total edge weights of its incoming links is equal to the total edge weights of its outgoing links ($\sum_{j=1}^n a_{ij} = \sum_{j=1}^n a_{ji}$ for all i).



Averaging Algorithms and Consensus, Fig. 1 A directed graph that characterizes the interaction among five agents, where A_i , $i = 1, \dots, 5$, denotes agent i . An *arrow* from agent j to agent i indicates that agent i receives information from agent j . The directed graph has a directed spanning tree but is not strongly connected. Here both agents 1 and 2 have directed paths to all other agents

Consensus

Consensus has a long history in management science, statistical physics, and distributed computing and finds recent interests in distributed control. While in the area of distributed control of networked systems the term *consensus* was initially more or less dominantly referred to the case of a continuous-time version of a distributed linear averaging algorithm, such a term has been broadened to a great extent later on. Related problems to consensus include synchronization, agreement, and rendezvous. The study of consensus can be categorized in various manners. For example, in terms of the final consensus value, the agents could reach a consensus on the average, a weighted average, the maximum value, the minimum value, or a general function of their initial conditions, or even a (changing) state that serves as a reference. A consensus algorithm could be linear or nonlinear. Consensus algorithms can be designed for agents with linear or nonlinear dynamics. As the agent dynamics become more complicated, so do the algorithm design and analysis. Numerous issues are also involved in consensus such as network topologies (fixed vs. switching, deterministic vs. random, directed vs. undirected, asynchronous vs. synchronous), time delay, quantization, optimality, sampling effects, and convergence speed. For example, in real applications, due to nonuniform communication/sensing ranges or limited field of view of sensors, the network topology could be directed rather than undirected. Also due to unreliable communication/sensing and limited communication/sensing ranges, the network topology could be switching rather than fixed.

Consensus for Agents with Single-Integrator Dynamics

We start with a fundamental consensus algorithm for agents with single-integrator dynamics. The results in this section follow from Jadbabaie et al. (2003), Olfati-Saber et al. (2007), Ren and Beard (2008), Moreau (2005), and Agaev and Chebotarev (2000). Consider agents with single-integrator dynamics

$$\dot{x}_i(t) = u_i(t), \quad i = 1, \dots, n, \quad (1)$$

where x_i is the state and u_i is the control input. A common consensus algorithm for (1) is

$$u_i(t) = \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)[x_j(t) - x_i(t)], \quad (2)$$

where $\mathcal{N}_i(t)$ is the neighbor set of agent i at time t and $a_{ij}(t)$ is the (i, j) entry of the adjacency matrix $\mathcal{A}$ of the graph $\mathcal{G}$ at time t . A consequence of (2) is that the state $x_i(t)$ of agent i is driven toward the states of its neighbors or equivalently toward the weighted average of its neighbors' states. The closed-loop system of (1) using (2) can be written in matrix form as

$$\dot{x}(t) = -\mathcal{L}(t)x(t), \quad (3)$$

where x is a column stack vector of all x_i and $\mathcal{L}$ is the Laplacian matrix. Consensus is *reached* if for all initial states, the agents' states eventually become identical. That is, for all $x_i(0)$, $\|x_i(t) - x_j(t)\|$ approaches zero eventually.

The properties of the Laplacian matrix $\mathcal{L}$ play an important role in the analysis of the closed-loop system (3). When the graph $\mathcal{G}$ (and hence the associated Laplacian matrix $\mathcal{L}$) is fixed, (3) can be analyzed by studying the eigenvalues and eigenvectors of $\mathcal{L}$. Due to its special structure, for any graph $\mathcal{G}$, the associated Laplacian matrix $\mathcal{L}$ has at least one zero eigenvalue with an associated right eigenvector $\mathbf{1}$ (column vector of all ones) and all other eigenvalues have positive real parts. To ensure consensus, it is equivalent to ensure that $\mathcal{L}$ has a simple zero eigenvalue. It can be shown that the following three statements are equivalent: (i) the agents reach a consensus exponentially for arbitrary initial states; (ii) the graph $\mathcal{G}$ has a directed spanning tree; and (iii) the Laplacian matrix $\mathcal{L}$ has a simple zero eigenvalue with an associated right eigenvector $\mathbf{1}$ and all other eigenvalues have positive real parts. When consensus is reached, the final consensus value is a weighted average of the initial states of those agents that have directed paths to all other agents (see Fig. 3 for an illustration).

When the graph $G(t)$ is switching at time instants $t_0, t_1, \dots$, the solution to the closed-loop system (3) is given by $x(t) = \Phi(t, 0)x(0)$, where $\Phi(t, 0)$ is the transition matrix corresponding to $-\mathcal{L}(t)$. Consensus is reached if $\Phi(t, 0)$ eventually converges to a matrix with identical rows. Here $\Phi(t, 0) = \Phi(t, t_k)\Phi(t_k, t_{k-1}) \cdots \Phi(t_1, 0)$, where $\Phi(t_k, t_{k-1})$ is the transition matrix corresponding to $\mathcal{L}(t)$ at time interval $[t_{k-1}, t_k]$. It turns out that each transition matrix is a *row-stochastic* matrix with positive diagonal entries. A square matrix is row stochastic if all its entries are nonnegative and all of its row sums are one. The consensus convergence can be analyzed by studying the product of row-stochastic matrices. Another analysis technique is a Lyapunov approach (e.g., $\max x_i - \min x_i$). It can be shown that the agents' states reach a consensus if there exists an infinite sequence of contiguous, uniformly bounded time intervals, with the property that across each such interval, the union of the graphs $G(t)$ has a directed spanning tree. That is, across each such interval, there exists at least one agent that can directly or indirectly influence all other agents. It is also possible to achieve certain nice features by designing nonlinear consensus algorithms of the form $u_i(t) = \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t) \psi[x_j(t) - x_i(t)]$, where $\psi(\cdot)$ is a nonlinear function satisfying certain properties. One example is a continuous nondecreasing odd function. For example, a saturation type function could be introduced to account for actuator saturation and a signum type function could be introduced to achieve finite-time convergence.

As shown above, for single-integrator dynamics, the consensus convergence is determined entirely by the network topologies. The primary reason is that the single-integrator dynamics are internally stable. However, when more complicated agent dynamics are involved, the consensus algorithm design and analysis become more complicated. On one hand, whether the graph is undirected (respectively, switching) or not has significant influence on the complexity of the consensus analysis. On the other hand, not only the network topology but also the agent dynamics themselves and the parameters in the consensus algorithm play important roles. Next we

introduce consensus for agents with general linear and nonlinear dynamics.

Consensus for Agents with General Linear Dynamics

In some circumstances, it is relevant to deal with agents with general linear dynamics, which can also be regarded as linearized models of certain nonlinear dynamics. The results in this section follow from Li et al. (2010). Consider agents with general linear dynamics

$$\dot{x}_i = Ax_i + Bu_i, \quad y_i = Cx_i, \quad (4)$$

where $x_i \in \mathbb{R}^m$, $u_i \in \mathbb{R}^p$, and $y_i \in \mathbb{R}^q$ are, respectively, the state, the control input, and the output of agent i and A, B, C are constant matrices with compatible dimensions.

When each agent has access to the relative states between itself and its neighbors, a distributed static consensus algorithm is designed for (4) as

$$u_i = cK \sum_{j \in \mathcal{N}_i} a_{ij}(x_i - x_j), \quad (5)$$

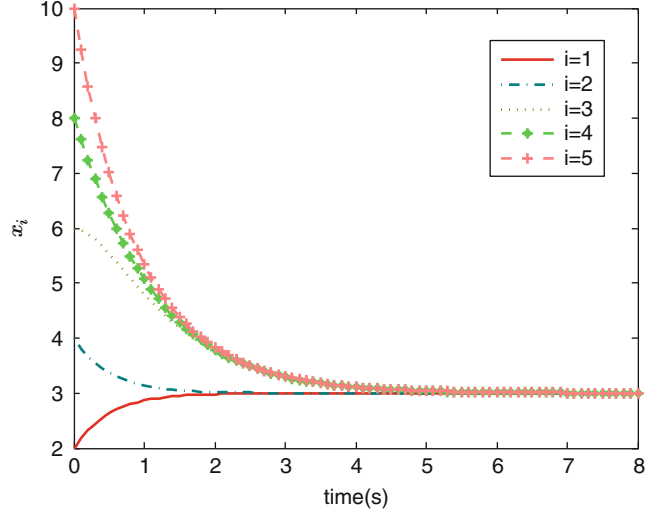
where $c > 0$ is a coupling gain, $K \in \mathbb{R}^{p \times m}$ is the feedback gain matrix, and $\mathcal{N}_i$ and a_{ij} are defined as in (2). It can be shown that if the graph G has a directed spanning tree, consensus is reached using (5) for (4) if and only if all the matrices $A + c\lambda_i(\mathcal{L})BK$, where $\lambda_i(\mathcal{L}) \neq 0$ are Hurwitz. Here $\lambda_i(\mathcal{L})$ denotes the i th eigenvalue of the Laplacian matrix $\mathcal{L}$. A necessary condition for reaching a consensus is that the pair (A, B) is stabilizable. The consensus algorithm (5) can be designed via two steps:

- Solve the linear matrix inequality $A^T P + PA - 2BB^T < 0$ to get a positive-definite solution P . Then let the feedback gain matrix $K = -B^T P^{-1}$.
- Select the coupling strength c larger than the threshold value $1 / \min_{\lambda_i(\mathcal{L}) \neq 0} \text{Re}[\lambda_i(\mathcal{L})]$, where $\text{Re}(\cdot)$ denotes the real part.

Note that here the threshold value depends on the eigenvalues of the Laplacian matrix, which

Averaging Algorithms and Consensus, Fig. 2

Consensus for five agents using the algorithm (2) for (1). Here the graph $\mathcal{G}$ is given by Fig. 1. The initial states are chosen as $x_i(0) = 2i$, where $i = 1, \dots, 5$. Consensus is reached as $\mathcal{G}$ has a directed spanning tree. The final consensus value is a weighted average of the initial states of agents 1 and 2



is in some sense global information. To overcome such a limitation, it is possible to introduce adaptive gains in the algorithm design. The gains could be updated dynamically using local information.

When the relative states between each agent and its neighbors are not available, one is motivated to make use of the output information and employ observer-based design to estimate the relative states. An observer-type consensus algorithm is designed for (4) as

$$\begin{aligned} \dot{v}_i &= (A + BF)v_i + cL \sum_{j \in \mathcal{N}_i} a_{ij} [C(v_i - v_j) \\ &\quad - (y_i - y_j)], \\ u_i &= Fv_i, \quad i = 1, \dots, n, \end{aligned} \quad (6)$$

where $v_i \in \mathbb{R}^m$ are the observer states, $F \in \mathbb{R}^{p \times n}$ and $L \in \mathbb{R}^{m \times q}$ are the feedback gain matrices, and $c > 0$ is a coupling gain. Here the algorithm (6) uses not only the relative outputs between each agent and its neighbors but also its own and neighbors' observer states. While relative outputs could be obtained through local measurements, the neighbors' observer states can only be obtained via communication. It can be shown that if the graph $\mathcal{G}$ has a directed spanning tree, consensus is reached using (6) for (4) if the matrices $A + BF$ and $A + c\lambda_i(L)LC$, where $\lambda_i(L) \neq 0$, are Hurwitz. The

observer-type consensus algorithm (6) can be seen as an extension of the single-system observer design to multi-agent systems. Here the *separation principle* of the traditional observer design still holds in the multi-agent setting in the sense that the feedback gain matrices F and L can be designed separately.

Consensus for Agents with Nonlinear Dynamics

In multi-agent applications, agents usually represent physical vehicles with special dynamics, especially nonlinear dynamics for the most part. Examples include Lagrangian systems for robotic manipulators and autonomous robots, nonholonomic systems for unicycles, attitude dynamics for rigid bodies, and general nonlinear systems. Similar to the consensus algorithms for linear multi-agent systems, the consensus algorithms used for these nonlinear agents are often designed based on state differences between each agent and its neighbors. But due to the inherent nonlinearity, the problem is more complicated and additional terms might be required in the algorithm design. The main techniques used in the consensus analysis for nonlinear multi-agent systems are often Lyapunov-based techniques (Lyapunov functions, passivity theory, nonlinear contraction analysis, and potential functions).

Early results on consensus for agents with nonlinear dynamics primarily focus on

undirected graphs to exploit the symmetry to facilitate the construction of Lyapunov function candidates. Unfortunately, the extension from an undirected graph to a directed one is nontrivial. For example, the directed graph does not preserve the passivity properties in general. Moreover, the directed graph could cause difficulties in the design of (positive-definite) Lyapunov functions. One approach is to integrate the nonnegative left eigenvector of the Laplacian matrix associated with the zero eigenvalue into the Lyapunov function, which is valid for strongly connected graphs and has been applied in some problems. Another approach is based on sliding mode control. The idea is to design a sliding surface for reaching a consensus. Taking multiple Lagrangian systems as an example, the agent dynamics are represented by

$$\begin{aligned} M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + g_i(q_i) &= \tau_i, \\ i &= 1, \dots, n, \end{aligned} \quad (7)$$

where $q_i \in \mathbb{R}^p$ is the vector of generalized coordinates, $M_i(q_i) \in \mathbb{R}^{p \times p}$ is the symmetric positive-definite inertia matrix, $C_i(q_i, \dot{q}_i)\dot{q}_i \in \mathbb{R}^p$ is the vector of Coriolis and centrifugal torques, $g_i(q_i) \in \mathbb{R}^p$ is the vector of gravitational torque, and $\tau_i \in \mathbb{R}^p$ is the vector of control torque on the i th agent. The sliding surface can be designed as

$$s_i = \dot{q}_i - \dot{q}_{ri} = \dot{q}_i + \alpha \sum_{j \in \mathcal{N}_i} a_{ij}(q_i - q_j) \quad (8)$$

where α is a positive scalar. Note that when $s_i = 0$, (8) is actually the closed-loop system of a consensus algorithm for single integrators. Then if the control torque τ_i can be designed using only local information from neighbors to drive s_i to zero, consensus will be reached as s_i can be treated as a vanishing disturbance to a system that reaches consensus exponentially.

It is generally very challenging to deal with general directed or switching graphs for agents with more complicated dynamics other than single-integrator dynamics. In some cases, the challenge could be overcome by introducing and

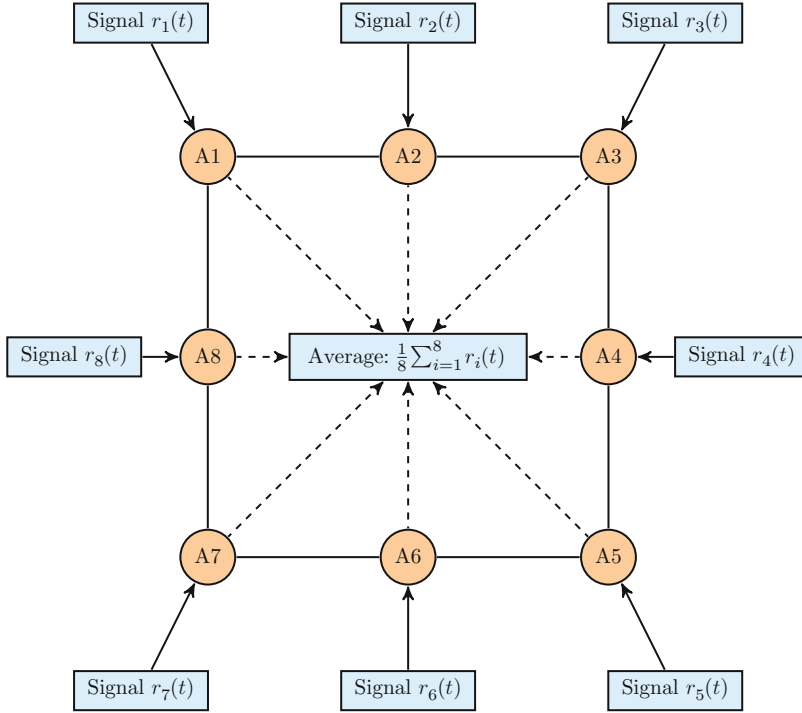
updating additional auxiliary variables (often observer-based algorithms) and exchanging these variables between neighbors (see, e.g., (6)). In the algorithm design, the agents might use not only relative physical states between neighbors but also local auxiliary variables from neighbors. While relative physical states could be obtained through sensing, the exchange of auxiliary variables can only be achieved by communication. Hence such generalization is obtained at the price of increased communication between the neighboring agents. Unlike some other algorithms, it is generally impossible to implement the algorithm relying on purely relative sensing between neighbors without the need for communication.

Averaging Algorithms

Existing distributed averaging algorithms are primarily static averaging algorithms based on linear local average iterations or gossip iterations. These algorithms are capable of computing the average of the initial conditions of all agents (or constant signals) in a network. In particular, the linear local-average-iteration algorithms are usually synchronous, where at each iteration each agent repeatedly updates its state to be the average of those of its neighbors. The gossip algorithms are asynchronous, where at each iteration a random pair of agents are selected to exchange their states and update them to be the average of the two. Dynamic averaging algorithms are of significance when there exist time-varying signals. The objective is to compute the average of these time-varying signals in a distributed manner.

Static Averaging

Take a linear local-average-iteration algorithm as an example. The results in this section follow from Tsitsiklis et al. (1986), Jadbabaie et al. (2003), and Olfati-Saber et al. (2007). Let x_i be the information state of agent i . A linear local-average-iteration-type algorithm has the form



Averaging Algorithms and Consensus, Fig. 3 Illustration of distributed averaging of multiple (time-varying) signals. Here A_i denotes agent i and $r_i(t)$ denotes

a (time-varying) signal associated with agent i . Each agent needs to compute the average of all agents' signals but can communicate with only its neighbors

$$x_i[k+1] = \sum_{j \in \mathcal{N}_i[k]} a_{ij}[k] x_j[k], \quad i = 1, \dots, n, \quad (9)$$

where k denotes a communication event, $\mathcal{N}_i[k]$ denotes the neighbor set of agent i , and $a_{ij}[k]$ is the (i, j) entry of the adjacency matrix $\mathcal{A}$ of the graph $\mathcal{G}$ that represents the communication topology at time k , with the additional assumption that $\mathcal{A}$ is row stochastic and $a_{ii}[k] > 0$ for all $i = 1, \dots, n$. Intuitively, the information state of each agent is updated as the weighted average of its current state and the current states of its neighbors at each iteration. Note that an agent maintains its current state if it does not exchange information with other agents at that event instant. In fact, a discretized version of the closed-loop system of (1) using (2) (with a sufficiently small sampling period) takes in the form of (9). The objective here is for all agents to compute the average of their initial states by communicating with only

their neighbors. That is, each $x_i[k]$ approaches $\frac{1}{n} \sum_{j=1}^n x_j[0]$ eventually. To compute the average of multiple constant signals c_i , we could simply set $x_i[0] = c_i$. The algorithm (9) can be written in matrix form as $x[k+1] = \mathcal{A}[k]x[k]$, where x is a column stack vector of all x_i and $\mathcal{A}[k] = [a_{ij}[k]]$ is a row-stochastic matrix.

When the graph $\mathcal{G}$ (and hence the matrix $\mathcal{A}$) is fixed, the convergence of the algorithm (9) can be analyzed by studying the eigenvalues and eigenvectors of the row-stochastic matrix $\mathcal{A}$. Because all diagonal entries of $\mathcal{A}$ are positive, Gershgorin's disc theorem implies that all eigenvalues of $\mathcal{A}$ are either within the open unit disk or at one. When the graph $\mathcal{G}$ is strongly connected, the Perron-Frobenius theorem implies that $\mathcal{A}$ has a simple eigenvalue at one with an associated right eigenvector $\mathbf{1}$ and an associated positive left eigenvector. Hence when $\mathcal{G}$ is strongly connected, it turns out that $\lim_{k \rightarrow \infty} \mathcal{A}^k = \mathbf{1}v^T$, where v^T is a positive left eigenvector of $\mathcal{A}$ associated with

the eigenvalue one and satisfies $v^T \mathbf{1} = 1$. Note that $x[k] = \mathcal{A}^k x[0]$. Hence, each agent's state $x_i[k]$ approaches $v^T x[0]$ eventually. If it can be further ensured that $v = \frac{1}{n} \mathbf{1}$, then averaging is achieved. It can be shown that the agents' states converge to the average of their initial values if and only if the directed graph $\mathcal{G}$ is both strongly connected and balanced or the undirected graph $\mathcal{G}$ is connected. When the graph is switching, the convergence of the algorithm (9) can be analyzed by studying the product of row-stochastic matrices. Such analysis is closely related to Markov chains. It can be shown that the agents' states converge to the average of their initial values if the directed graph $\mathcal{G}$ is balanced at each communication event and strongly connected in a joint manner or the undirected graph $\mathcal{G}$ is jointly connected.

Dynamic Averaging

In a more general setting, there exist n time-varying signals, $r_i(t)$, $i = 1, \dots, n$, which could be an external signal or an output from a dynamical system. Here $r_i(t)$ is available to only agent i and each agent can exchange information with only its neighbors. Each agent maintains a local estimate, denoted by $x_i(t)$, of the average of all the signals $\bar{r}(t) \triangleq \frac{1}{n} \sum_{k=1}^n r_k(t)$. The objective is to design a distributed algorithm for agent i based on $r_i(t)$ and $x_j(t)$, $j \in \mathcal{N}_i(t)$, such that all agents will finally track the average that changes over time. That is, $\|x_i(t) - \bar{r}(t)\|$, $i = 1, \dots, n$, approaches zero eventually. Such a dynamic averaging idea finds applications in distributed sensor fusion with time-varying measurements (Spanos and Murray 2005; Bai et al. 2011b) and distributed estimation and tracking (Yang et al. 2008).

Figure 3 illustrates the dynamic averaging idea. If there exists a central station that can always access the signals of all agents, then it is trivial to compute the average. Unfortunately, in a distributed context, where there does not exist a central station and each agent can only communicate with its local neighbors, it is challenging for each agent to compute the average that changes

over time. While each agent could compute the average of its own and local neighbors' signals, this will not be the average of all signals.

When the signal $r_i(t)$ can be arbitrary but its derivative exists and is bounded almost everywhere, a distributed nonlinear nonsmooth algorithm is designed in Chen et al. (2012) as

$$\begin{aligned} \dot{\phi}_i(t) &= \alpha \sum_{j \in \mathcal{N}_i} \text{sgn}[x_j(t) - x_i(t)] \\ x_i(t) &= \phi_i(t) + r_i(t), \quad i = 1, \dots, n, \end{aligned} \quad (10)$$

where α is a positive scalar, $\mathcal{N}_i$ denotes the neighbor set of agent i , $\text{sgn}(\cdot)$ is the signum function defined componentwise, ϕ_i is the internal state of the estimator with $\phi_i(0) = 0$, and x_i is the estimate of the average $\bar{r}(t)$. Due to the existence of the discontinuous signum function, the solution of (10) is understood in the Filippov sense (Cortes 2008).

The idea behind the algorithm (10) is as follows. First, (10) is designed to ensure that $\sum_{i=1}^n x_i(t) = \sum_{i=1}^n r_i(t)$ holds for all time. Note that $\sum_{i=1}^n x_i(t) = \sum_{i=1}^n \phi_i(t) + \sum_{i=1}^n r_i(t)$. When the graph $\mathcal{G}$ is undirected and $\phi_i(0) = 0$, it follows that $\sum_{i=1}^n \phi_i(t) = \sum_{i=1}^n \phi_i(0) + \alpha \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} \int_0^t \text{sgn}[x_j(\tau) - x_i(\tau)] d\tau = 0$. As a result, $\sum_{i=1}^n x_i(t) = \sum_{i=1}^n r_i(t)$ holds for all time. Second, when $\mathcal{G}$ is connected, if the algorithm (10) guarantees that all estimates x_i approach the same value in *finite time*, then it can be guaranteed that each estimate approaches the average of all signals in finite time.

Summary and Future Research Directions

Averaging algorithms and consensus play an important role in distributed control of networked systems. While there is significant progress in this direction, there are still numerous open problems. For example, it is challenging to achieve averaging when the graph is not balanced. It is generally not clear how to deal with a general directed or switching graph for

nonlinear agents or nonlinear algorithms when the algorithms are based on only interagent physical state coupling without the need for communicating additional auxiliary variables between neighbors. The study of consensus for multiple underactuated agents remains a challenge. Furthermore, when the agents' dynamics are heterogeneous, it is challenging to design consensus algorithms. In addition, in the existing study, it is often assumed that the agents are cooperative. When there exist faulty or malicious agents, the problem becomes more involved.

Cross-References

- [Distributed Optimization](#)
- [Dynamic Graphs, Connectivity of](#)
- [Flocking in Networked Systems](#)
- [Graphs for Modeling Networked Interactions](#)
- [Networked Systems](#)
- [Oscillator Synchronization](#)
- [Vehicular Chains](#)

Bibliography

- Agaev R, Chebotarev P (2000) The matrix of maximum out forests of a digraph and its applications. *Autom Remote Control* 61(9):1424–1450
- Agaev R, Chebotarev P (2005) On the spectra of non-symmetric Laplacian matrices. *Linear Algebra Appl* 399:157–178
- Bai H, Arcak M, Wen J (2011a) Cooperative control design: a systematic, passivity-based approach. Springer, New York
- Bai H, Freeman RA, Lynch KM (2011b) Distributed Kalman filtering using the internal model average consensus estimator. In: *Proceedings of the American control conference*, San Francisco, pp 1500–1505
- Bullo F, Cortes J, Martinez S (2009) *Distributed control of robotic networks*. Princeton University Press, Princeton
- Chen F, Cao Y, Ren W (2012) Distributed average tracking of multiple time-varying reference signals with bounded derivatives. *IEEE Trans Autom Control* 57(12):3169–3174
- Cortes J (2008) Discontinuous dynamical systems. *IEEE Control Syst Mag* 28(3):36–73
- Jadbabaie A, Lin J, Morse AS (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans Autom Control* 48(6):988–1001
- Li Z, Duan Z, Chen G, Huang L (2010) Consensus of multiagent systems and synchronization of complex networks: a unified viewpoint. *IEEE Trans Circuits Syst I Regul Pap* 57(1):213–224
- Mesbahi M, Egerstedt M (2010) *Graph theoretic methods for multiagent networks*. Princeton University Press, Princeton
- Moreau L (2005) Stability of multi-agent systems with time-dependent communication links. *IEEE Trans Autom Control* 50(2):169–182
- Olfati-Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. *Proc IEEE* 95(1):215–233
- Qu Z (2009) *Cooperative control of dynamical systems: applications to autonomous vehicles*. Springer, London
- Ren W, Beard RW (2008) *Distributed consensus in multi-vehicle cooperative control*. Springer, London
- Ren W, Cao Y (2011) *Distributed coordination of multi-agent networks*. Springer, London
- Spanos DP, Murray RM (2005) Distributed sensor fusion using dynamic consensus. In: *Proceedings of the IFAC world congress*, Prague
- Tsitsiklis JN, Bertsekas DP, Athans M (1986) Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans Autom Control* 31(9):803–812
- Yang P, Freeman RA, Lynch KM (2008) Multi-agent coordination by decentralized estimation and control. *IEEE Trans Autom Control* 53(11):2480–2496

B

Backstepping

► [Backstepping for PDEs](#)

Keywords

Backstepping · Boundary control · Distributed parameter systems · Lyapunov function · Partial differential equations (PDEs) · Stabilization

Synonyms

[Backstepping](#); [Infinite-Dimensional](#)

Backstepping for PDEs

Rafael Vazquez¹ and Miroslav Krstic²

¹Department of Aerospace Engineering,
Universidad de Sevilla, Sevilla, Spain

²Department of Mechanical and Aerospace
Engineering, University of California,
San Diego, La Jolla, CA, USA

Abstract

Backstepping is an elegant constructive method, with roots in finite-dimensional control theory, that allows the solution of numerous boundary control and estimation problems for partial differential equations (PDEs). This entry reviews the main ingredients of the method, namely, the concepts of a backstepping invertible transformation, a target system, and the kernel equations. As a basic example, stabilization of a reaction-diffusion equation is explained.

Introduction to PDE Backstepping

In the context of partial differential equations (PDEs), backstepping is a constructive method, originated in the 2000s, mainly used to design boundary controllers and observers, both adaptive and nonadaptive, for numerous classes of systems. Its name comes from the use of the so-called backstepping transformation (a Volterra-type integral transform) as a tool to find feedback control laws and observer gains.

The method has three main ingredients. First, one needs to select a *target system* which verifies the desired properties (most often stability, proven with a Lyapunov function), but still closely resembles the original system. Next, an integral transformation (the backstepping transformation) is posed to map the original plant into the target system in the appropriate functional spaces. The invertibility of the transformation needs to be shown. Finally, using the original

and target systems and the transformation, the *kernel equations* are found. Their solution is the kernel of the integral transformation, which in turn determines the control law. These equations are typically of Goursat type, that is, hyperbolic boundary problems on a triangular domain (with boundary values on two sides and the third side determining the control law), and can usually be proven solvable by transforming the boundary problems to integral equations and then using the method of successive approximations. Tying up all these elements, stability of the closed-loop system is then obtained, based on the stability of the target system which is inherited by the closed-loop system as a result of the structure and well-posedness of the transformation.

These ingredients are always present when using PDE backstepping and are closely connected among themselves. Judiciously choosing the target system will result in solvable kernel equations and an invertible transformation. On the other hand, an ill-chosen target system typically results in kernel equations that cannot be solved or even properly formulated, or a non-invertible transformation.

Next, the rationale of the method is illustrated for a basic case, namely, stabilization of a reaction-diffusion equation.

Reaction-Diffusion Equation

Consider a reaction-diffusion equation

$$u_t = \epsilon u_{xx} + \lambda(x)u, \quad (1)$$

where $u(t, x)$ is the state, for $x \in [0, 1]$ and $t > 0$, with $\epsilon > 0$ and $\lambda(x)$ a continuous function, with boundary conditions

$$u(t, 0) = 0, \quad u(t, 1) = U(t), \quad (2)$$

where $U(t)$ is the actuation variable. These are the so-called Dirichlet-type boundary conditions, but Neumann- ($u_x(t, 0) = 0$) or Robin-type ($u_x(t, 0) = qu(t, 0)$) boundary conditions are also frequently considered. Denote the initial condition $u(0, x)$ as $u_0(x)$.

The system (1)–(2) has an equilibrium, namely, $u \equiv 0$, which is unstable if $\lambda(x)$ is sufficiently large. Thus, the control problem is to design a feedback $U(t)$ such that the origin becomes stable (stability will be made precise by using a Lyapunov approach).

Target System (and Its Stability)

Since the last term in (1) is a potential source of instability, the most natural objective of feedback would be to eliminate it, thus reaching the following target system

$$w_t = \epsilon w_{xx}, \quad (3)$$

that is, a heat equation, with homogeneous Dirichlet boundary conditions

$$w(t, 0) = 0, \quad w(t, 1) = 0. \quad (4)$$

To study the target system stability, consider the $L^2([0, 1])$ space of functions (whose square is integrable in the interval $[0, 1]$), endowed with the norm $\|f\|^2 = \int_0^1 f^2(x)dx$.

Denoting by w_0 the initial condition of (3)–(4), it is well-known that if $w_0 \in L^2([0, 1])$, then $w(t, \cdot) \in L^2([0, 1])$ for each $t > 0$ and, using $\|w(t, \cdot)\|^2$ as a Lyapunov function, the following exponential stability result is verified:

$$\|w(t, \cdot)\|^2 \leq e^{-\alpha t} \|w_0\|^2, \quad (5)$$

where α is a positive number.

Backstepping Transformation (and Its Inverse)

Now, pose the following coordinate transformation

$$w(t, x) = u(t, x) - \int_0^x k(x, \xi)u(t, \xi)d\xi, \quad (6)$$

to map the system (1) into (3). The transformation (6), whose second term is a Volterra-type integral transformation (because of its lower-triangular structure), is the backstepping transformation, and its kernel $k(x, \xi)$ is known as the backstepping kernel. One of the properties

of a Volterra-type transformation is that it is invertible under very mild conditions on the kernel $k(x, \xi)$, for instance, if it is bounded. Assuming for the moment that the kernel verifies this property, then an inverse transformation can be posed as

$$u(t, x) = w(t, x) + \int_0^x l(x, \xi) w(t, \xi) d\xi, \quad (7)$$

where $l(x, y)$ is known as the inverse kernel, and it is also bounded.

It is easy to see that both transformations (6) and (7) map $L^2([0, 1])$ functions into $L^2([0, 1])$ functions. In particular, if u and w are related by the transformations as in (6) and (7), it follows that $\|u\|^2 \leq C_1 \|w\|^2$ and $\|w\|^2 \leq C_2 \|u\|^2$ for some $C_1, C_2 > 0$.

Kernel Equations

The kernel $k(x, \xi)$ verifies the following equation

$$\epsilon k_{xx}(x, \xi) = \epsilon k_{\xi\xi}(x, \xi) + \lambda(\xi)k(x, \xi), \quad (8)$$

with boundary conditions

$$\epsilon k(x, 0) = -0, \quad k(x, x) = -\frac{1}{2\epsilon} \int_0^x \lambda(\xi) d\xi. \quad (9)$$

Equation (8) is a hyperbolic equation of Goursat type. The way to derive (8), along with (9), is by substituting the transformation (6) into the target system (3) and eliminating w . Then, integrating by parts, one gets a weak formulation of (8)–(9).

The PDE (8)–(9) is well-posed and can be solved numerically fast and efficiently. It can be also reduced to an integral equation, by defining $\delta = x + \xi$ and $\eta = x - \xi$, and denoting $G(\delta, \eta) = k(x, \xi) = k\left(\frac{\delta+\eta}{2}, \frac{\delta-\eta}{2}\right)$. Then, $G(\delta, \eta)$ verifies the following equation:

$$4\epsilon G_{\delta\eta} = \lambda\left(\frac{\delta-\eta}{2}\right) G(\delta, \eta), \quad (10)$$

with boundary conditions

$$G(\delta, \delta) = 0,$$

$$G(\delta, 0) = -\frac{1}{4\epsilon} \int_0^\delta \lambda\left(\frac{\tau}{2}\right) d\tau. \quad (11)$$

Equation (10) for G can be integrated, yielding an integral equation for G

$$\begin{aligned} G(\delta, \eta) = & -\frac{1}{4\epsilon} \int_\eta^\delta \lambda\left(\frac{\tau}{2}\right) d\tau \\ & + \frac{1}{4\epsilon} \int_\eta^\delta \int_0^\eta \lambda\left(\frac{\tau-s}{2}\right) G(\tau, s) ds d\tau. \end{aligned} \quad (12)$$

The method of successive approximations can be used to solve (12). Define $G_0(\delta, \eta) = -\frac{1}{4\epsilon} \int_\eta^\delta \lambda\left(\frac{\tau}{2}\right) d\tau$ and, for $n \geq 1$,

$$\begin{aligned} G_n(\delta, \eta) = & G_{n-1}(\delta, \eta) \\ & + \frac{1}{4\epsilon} \int_\eta^\delta \int_0^\eta \lambda\left(\frac{\tau-s}{2}\right) G_{n-1}(\tau, s) ds d\tau. \end{aligned} \quad (13)$$

Then it can be shown that $G(\delta, \eta) = \lim_{n \rightarrow \infty} G_n(\delta, \eta)$. The function G_n can be computed recursively and used to approximate symbolically G and thus k . This procedure shows that (8)–(9) is well-posed (the limit always exists) and its solution k is continuous (and thus invertible). In fact, for constant λ , the exact solution is

$$k(x, \xi) = -\frac{\lambda}{\epsilon} \xi \frac{I_1\left(\sqrt{\frac{\lambda}{\epsilon}}(x^2 - \xi^2)\right)}{\sqrt{\frac{\lambda}{\epsilon}}(x^2 - \xi^2)},$$

where I_1 is the first-order modified Bessel function of the first kind.

Feedback Law and Closed-Loop System Stability

The feedback U in (2) is what makes the original system behave like the target system and can be found by setting $x = 1$ in the backstepping transformation (6) and using the boundary conditions (2) and (4), which yields the feedback law

$$U(t) = \int_0^1 k(1, \xi) u(t, \xi) d\xi. \quad (14)$$

Finally, using (5) and the properties of the transformations (6) and (7), one can obtain

$$\begin{aligned} \|u(t)\|^2 &\leq C_2 \|w(t)\|^2 \leq C_2 e^{-\alpha t} \|w_0\|^2 \\ &\leq C_1 C_2 e^{-\alpha t} \|u_0\|^2, \end{aligned} \quad (15)$$

thus showing exponential stability for the origin of the system (1)–(2).

Summary and Future Directions

Backstepping as a method for control of PDEs is a fairly recent development, and the area of control of PDEs is in itself an emerging and active area of research with great potential in many engineering applications that cannot be modeled solely by finite-dimensional dynamics. To learn more about the basics, we recommend the book (Anfinsen and Aamo 2019) which contains a didactic exposition of the initial backstepping results, providing details to design controllers and observers for parabolic equations and others. While we only dealt with the basic theoretical developments, the subject on the other hand has been steadily growing in the last decade. The technical literature now contains a wealth of results in application areas as diverse as oil pipe flows, battery health estimation, thermoacoustics, multi-phase flows, 3-D printing, or traffic control, among others, with hundreds of additional references. To name a few, we would like to mention several books written on the subject which include interesting recent and forth coming developments in topics such as flow control (Krstic 2009), adaptive control of parabolic PDEs (Krstic and Smyshlyaev 2008), adaptive control of hyperbolic PDEs (Smyshlyaev and Krstic 2010), or delays (Vazquez and Krstic 2008).

Cross-References

- [Adaptive Control of PDEs](#)
- [Control of Nonlinear Systems with Delays](#)

- [Input-to-State Stability for PDEs](#)
- [Motion Planning for PDEs](#)

Bibliography

- Anfinsen H, Aamo OM (2019) Adaptive control of hyperbolic PDEs. Springer, Cham
- Krstic M (2009) Delay compensation for nonlinear, adaptive, and PDE systems. Birkhauser, Basel
- Krstic M, Smyshlyaev A (2008) Boundary control of PDEs: a course on backstepping designs. SIAM, Philadelphia
- Smyshlyaev A, Krstic M (2010) Adaptive control of parabolic PDEs. Princeton University Press, Princeton
- Vazquez R, Krstic M (2008) Control of turbulent and magnetohydrodynamic channel flow. Birkhauser, Basel

Backward Stochastic Differential Equations and Related Control Problems

Shige Peng

Shandong University, Jinan, Shandong Province, China

Abstract

A conditional expectation of the form $Y_t = E[\xi + \int_t^T f_s ds | \mathcal{F}_t]$ is regarded as a simple and typical example of backward stochastic differential equation (abbreviated by BSDE). BSDEs are widely applied to formulate and solve problems related to stochastic optimal control, stochastic games, and stochastic valuation.

Keywords

Brownian motion · Feynman-Kac formula · Lipschitz condition · Optimal stopping

Synonyms

BSDE

Definition

A typical real valued backward stochastic differential equation defined on a time interval $[0, T]$ and driven by a d -dim. Brownian motion B is

$$\begin{cases} dY_t = -f(t, Y_t, Z_t)dt + Z_t dB_t, \\ Y_T = \xi, \end{cases}$$

or its integral form

$$Y_t = \xi + \int_t^T f(s, \omega, Y_s, Z_s)ds - \int_t^T Z_s dB_s, \quad (1)$$

where ξ is a given random variable depending on the (canonical) Brownian path $B_t(\omega) = \omega(t)$ on $[0, T]$, $f(t, \omega, y, z)$ is a given function of the time t , the Brownian path ω on $[0, t]$, and the pair of variables $(y, z) \in \mathbb{R}^m \times \mathbb{R}^{m \times d}$. A solution of this BSDE is a pair of stochastic processes (Y_t, Z_t) , the solution of the above equation, on $[0, T]$ satisfying the following constraint: for each t , the value of $Y_t(\omega)$, $Z_t(\omega)$ depends only on the Brownian path ω on $[0, t]$. Notice that, because of this constraint, the extra freedom Z_t is needed. For simplicity we set $d = m = 1$.

Often square-integrable conditions for ξ and f and Lipschitz condition for f with respect to (y, z) are assumed under which there exists a unique square-integrable solution (Y_t, Z_t) on $[0, T]$ (existence and uniqueness theorem of BSDE). We can also consider a multidimensional process Y and/or a multidimensional Brownian motion B , L^p -integrable conditions ($p \geq 1$) for ξ and f , as well as local Lipschitz conditions of f with respect to (y, z) . If Y_t is real valued, we often call the equation a real valued BSDE.

We compare this BSDE with the classical stochastic differential equation (SDE):

$$dX_s = \sigma(X_s)dB_s + b(X_s)ds$$

with given initial condition $X_s|_{s=0} = x \in \mathbb{R}^n$. Its integral form is

$$\begin{aligned} X_t(\omega) &= x + \int_0^t \sigma(X_s(\omega))dB_s(\omega) \\ &\quad + \int_0^t b(X_s(\omega))ds. \end{aligned} \quad (2)$$

Linear backward stochastic differential equation was firstly introduced (Bismut 1973) in stochastic optimal control problems to solve the adjoint equation in the stochastic maximum principle of Pontryagin's type. The above existence and uniqueness theorem was obtained by Pardoux and Peng (1990). In the research domain of economics, this type of 1-dimensional BSDE was also independently derived by Duffie and Epstein (1992). Comparison theorem of BSDE was obtained in Peng (1992) and improved in El Karoui et al. (1997a). Nonlinear Feynman-Kac formula was obtained in Peng (1991, 1992) and improved in Pardoux and Peng (1992). BSDE is applied as a nonlinear Black-Scholes option pricing formula in finance. This formulation was given in El Karoui et al. (1997b). We refer to a recent survey in Peng (2010) for more details.

Hedging and Risk Measuring in Finance

Let us consider the following hedging problem in a financial market with a typical model of continuous time asset price: the basic securities consist of two assets, a riskless one called bond, and a risky security called stock. Their prices are governed by $dP_t^0 = P_t^0 r dt$, for the bond, and

$$dP_t = P_t[bdt + \sigma dB_t], \quad \text{for the stock.}$$

Here we only consider the situation where the volatility rate $\sigma > 0$. The case of multidimensional stocks with degenerate volatility matrix σ can be treated by constrained BSDE. Assume that a small investor whose investment behavior cannot affect market prices and who invests at time $t \in [0, T]$ the amount π_t of his or her wealth Y_t in the security and π_t^0 in the bond, thus $Y_t = \pi_t^0 + \pi_t$. If his investment strategy is self-financing, then we

have $dY_t = \pi_t^0 dP_t^0 / P_t^0 + \pi_t dP_t / P_t$, thus

$$dY_t = (rY_t + \pi_t \sigma \theta) dt + \pi_t \sigma dB_t,$$

where $\theta = \sigma^{-1}(b - r)$. A strategy $(Y_t, \pi_t)_{t \in [0, T]}$ is said to be feasible if $Y_t \geq 0, t \in [0, T]$. A European path-dependent contingent claim settled at time T is a given nonnegative function of path $\xi = \xi((P_t)_{t \in [0, T]})$. A feasible strategy (Y, π) is called a hedging strategy against a contingent claim ξ at the maturity T if it satisfies

$$dY_t = (rY_t + \pi_t \sigma \theta) dt + \pi_t \sigma dB_t, \quad Y_T = \xi.$$

This problem can be regarded as finding a stochastic control π and an initial condition Y_0 such that the final state replicates the contingent claim ξ , i.e., $Y_T = \xi$. This type of replications is also called “exact controllability” in terms of stochastic control (see Peng 2005 for more general results).

Observe that $(Y, \pi \sigma)$ is the solution of the above BSDE. It is called a superhedging strategy if there exists an increasing process K_t , often called an accumulated consumption process, such that

$$dY_t = (rY_t + \pi_t \sigma \theta) dt + \pi_t \sigma dB_t - dK_t, \quad Y_T = \xi.$$

This type of strategies is often applied in a constrained market in which certain constraint $(Y_t, \pi_t) \in \Gamma$ is imposed. In fact a real market has many frictions and constraints. An example is the common case where interest rate R for borrowing money is higher than the bond rate r . The above equation for the hedging strategy becomes

$$dY_t = [rY_t + \pi_t \sigma \theta - (R - r)(\pi_t - Y_t)^+] dt + \pi_t \sigma dB_t, \quad Y_T = \xi,$$

where $[\alpha]^+ = \max\{\alpha, 0\}$. A short selling constraint $\pi_t \geq 0$ is also a typical requirement in markets. The method of constrained BSDE can be applied to this type of problems. BSDE theory provides powerful tools to the robust pricing and risk measures for contingent claims (see El Karoui et al. 1997a). For the dynamic risk

measure under Brownian filtration, see Rosazza Gianin (2006), Peng (2004), Barrieu and El Karoui (2005), Hu et al. (2005), and Delbaen et al. (2010).

Comparison Theorem

The comparison theorem, for a real valued BSDE, tells us that, if (Y_t, Z_t) and $(\bar{Y}_t, \bar{Z}_t)$ are two solutions of BSDE (1) with terminal condition $Y_T = \xi, \bar{Y}_T = \bar{\xi}$ such that $\xi(\omega) \geq \bar{\xi}(\omega), \omega \in \Omega$, then one has $Y_t \geq \bar{Y}_t$. This theorem holds if f and $\bar{\xi}, \bar{\xi}$ satisfy the abovementioned L^2 -integrability condition and f is a Lipschitz function in (y, z) . This theorem plays the same important role as the maximum principle in PDE theory. The theorem also has several very interesting generalizations (see Buckdahn et al. 2000).

Stochastic Optimization and Two-Person Zero-Sum Stochastic Games

An important point of view is to regard an expectation value as a solution of a special type of BSDE. Consider an optimal control problem

$$\min_u J(u) : \quad J(u) = E \left[\int_0^T l(X_s, u_s) ds + h(X_T) \right].$$

Here the state process X is controlled by the control process u_t which is valued in a control (compact) domain U through the following d -dimensional SDE

$$dX_s = b(X_s, u_s) ds + \sigma(X_s) dB_s$$

defined in a Wiener probability space $(\Omega, \mathcal{F}, P)$ with the Brownian motion $B_t(\omega) = \omega(t)$ which is the canonical process. Here we only discuss the case $\sigma \equiv I_d$ for simplicity. Observe that in fact the expected value $J(u)$ is $Y_0^u = E[Y_0^u]$, where Y_t^u solves the BSDE

$$Y_t^u = h(X_T) + \int_t^T l(X_s, u_s) ds - \int_t^T Z_s^u dB_s.$$

From Girsanov transformation, under the probability measure $\tilde{P}$ defined by

$$\frac{d\tilde{P}}{dP}|_T = \exp \left\{ \int_0^T b(X_s, u_s) dB_s - \frac{1}{2} \int_0^T |b(X_s, u_s, v_s)|^2 ds \right\}$$

X_t is a Brownian motion, and the above BSDE is changed to

$$Y_t^u = h(X_T) + \int_t^T [l(X_s, u_s) + \langle Z_s^u, b(X_s, u_s) \rangle] ds - \int_t^T Z_s^u dX_s,$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean scalar product in $\mathbb{R}^d$. Notice that P and $\tilde{P}$ are absolutely continuous with each other. Compare this BSDE with the following one:

$$\hat{Y}_t = h(X_T) + \int_t^T H(X_s, \hat{Z}_s) ds - \int_t^T \hat{Z}_s dX_s, \quad (3)$$

where $H(x, z) := \inf_{u \in U} \{l(x, u) + \langle z, b(x, u) \rangle\}$. It is a direct consequence of the comparison theorem of BSDE that $\hat{Y}_0 \leq Y_0^u = J(u)$, for any admissible control u_t . Moreover, one can find a feedback control $\hat{u}$ such that $\hat{Y}_0 = J(\hat{u})$.

The above BSDE method has been introduced to solve the following two-person zero-sum game (Hamadène and Lepeltier 1995):

$$\begin{aligned} & \max_v \min_u J(u, v), \quad J(u, v) \\ & = E \left[\int_0^T l(X_s, u_s, v_s) ds + h(X_T) \right] \end{aligned}$$

with

$$dX_s = b(X_s, u_s, v_s) ds + dB_s,$$

where (u_s, v_s) is formulated as above with compact control domains $u_s \in U$ and $v_s \in V$. In this case the equilibrium of the game exists if the following Isaac condition is satisfied:

$$\begin{aligned} H(x, z) &:= \max_{v \in V} \inf_{u \in U} \{l(x, u, v) + \langle z, b(x, u, v) \rangle\} \\ &= \inf_{u \in U} \max_{v \in V} \{l(x, u, v) + \langle z, b(x, u, v) \rangle\}, \end{aligned}$$

and the equilibrium is also obtained through a BSDE (3) defined above.

Nonlinear Feynman-Kac Formula

A very interesting situation is when $f = g(X_t, y, z)$ and $Y_T = \varphi(X_T)$ in BSDE (1). In this case we have the following relation, called “nonlinear Feynman-Kac formula,”

$$Y_t = u(t, X_t), \quad Z_t = \sigma^T(X_t) \nabla u(t, X_t)$$

where $u = u(t, x)$ is the solution of the following quasilinear parabolic PDE:

$$\partial_t u + \mathcal{L}u + g(x, u, \sigma^T \nabla u) = 0, \quad (4)$$

$$u(x, T) = \varphi(x), \quad (5)$$

where $\mathcal{L}$ is the following, possibly degenerate, elliptic operator:

$$\begin{aligned} \mathcal{L}\varphi(x) &= \frac{1}{2} \sum_{i,j=1}^d a_{ij}(x) \partial_{x_i x_j}^2 \varphi(x) \\ &+ \sum_{i=1}^d b_i(x) \partial_{x_i} \varphi(x), \quad a(x) = \sigma(x) \sigma^T(x). \end{aligned}$$

Nonlinear Feynman-Kac formula can be used to solve a nonlinear PDE of form (4) to (5) by a BSDE (1) coupled with an SDE (2).

A general principle is, once we solve a BSDE driven by a Markov process X for which the terminal condition Y_T at time T depends only on X_T and the generator $f(t, \omega, y, z)$ also depends on the state X_t at each time t , then the corresponding solution of the BSDE is also state dependent, namely, $Y_t = u(t, X_t)$, where u is the solution of the corresponding quasilinear PDE. Once Y_T and g are path functions of X , then the solution of the BSDE becomes also path dependent. In this sense, we can say that the PDE is in fact a “state-dependent BSDE,” and BSDE gives us a new generalization of “path-dependent PDE” of parabolic and/or elliptic types. This principle was illustrated in Peng (2010) for both quasilinear and fully nonlinear situations.

Observe that BSDE (1) and forward SDE (2) are only partially coupled. A fully coupled system of SDE and BSDE is called a forward-backward stochastic differential equation (FBSDE). It has the following form:

$$\begin{aligned} dX_t &= b(t, X_t, Y_t, Z_t)dt + \sigma(t, X_t, Y_t, Z_t)dB_t, \\ X_0 &= x \in \mathbb{R}^n, \\ -dY_t &= f(t, X_t, Y_t, Z_t)dt - Z_t dB_t, \quad Y_T = \varphi(X_T). \end{aligned}$$

In general the Lipschitz assumptions for b, σ, f , and φ w. r. t. (x, y, z) are not enough. Then Ma et al. (1994) have proposed a four-step scheme method of FBSDE for the nondegenerate Markovian case with σ independent of Z . For the case $\dim(x) = \dim(y) = n$, Hu and Peng (1995) proposed a new type of monotonicity condition. This method does not need to assume the coefficients to be deterministic. Peng and Wu (1999) have weakened the monotonicity condition. Observe that in the case where $b = \nabla_y H(x, y, z)$, $\sigma = \nabla_z H(x, y, z)$, and $f = \nabla_x H(x, y, z)$, for a given real valued function H convex in x concave in (y, z) , the above FBSDE is called the stochastic Hamilton equation associated to a stochastic optimal control problem. We also refer to the book of Ma and Yong (1999) for a systematic exposition on this subject. For time-symmetric forward-backward stochastic differential equations and its relation with stochastic optimality, see Peng and Shi (2003) and Han et al. (2010).

Reflected BSDE and Optimal Stopping

If (Y, Z) solves the BSDE

$$dY_s = -g(s, Y_s, Z_s)ds + Z_s dB_s - dK_s, \quad Y_T = \xi, \quad (6)$$

where K is a càdlàg and increasing process with $K_0 = 0$ and $K_t \in L_P^2(\mathcal{F}_t)$, then Y or (Y, Z, K) is called a *supersolution* of the BSDE, or *g-supersolution*. This notion is often used for constrained BSDEs. A typical situation is as follows: for a given continuous adapted process $(L_t)_{t \in [0, T]}$, find a smallest g -supersolution

(Y, Z, K) such that $Y_t \geq L_t$. This problem was initiated in El Karoui et al. (1997b). It is proved that this problem is equivalent to finding a triple (Y, Z, K) satisfying (4) and the following reflecting condition of Skorohod type:

$$Y_s \geq L_s, \quad \int_0^T (Y_s - L_s) dK_s = 0. \quad (7)$$

In fact $\tau^* := \inf\{t \in [0, T] : K_t > 0\}$ is the optimal stopping time associated to this BSDE. A well-known example is the pricing of American option.

Moreover, a new type of nonlinear Feynman-Kac formula was introduced: if all coefficients are given as in the formulation of the above nonlinear Feynman-Kac formula and $L_s = \Phi(X_s)$ where Φ satisfies the same condition as φ , then we have $Y_s = u(s, X_s)$, where $u = u(t, x)$ is the solution of the following variational inequality:

$$\begin{aligned} \min\{\partial_t u + \mathcal{L}u + g(x, u, \sigma^* Du), u - \Phi\} \\ = 0, \quad (t, x) \in [0, T] \times \mathbb{R}^n, \quad (8) \end{aligned}$$

with terminal condition $u|_{t=T} = \varphi$. They also demonstrated that this reflected BSDE is a powerful tool to deal with contingent claims of American types in a financial market with constraints.

BSDE reflected within two barriers, a lower one L and an upper one U , was first investigated by Cvitanic and Karatzas (1996) where a type of nonlinear Dynkin games was formulated for a two-player model with zero-sum utility and each player chooses his own optimal exit time.

Stochastic optimal switching problems can be also solved by new types of oblique-reflected BSDEs.

A more general case of constrained BSDE is to find the smallest g -supersolution (Y, Z, K) with constraint $(Y_t, Z_t) \in \Gamma_t$ where, for each $t \in [0, T]$, Γ_t (El Karoui and Quenez 1995; Cvitanic and Karatzas 1993; El Karoui et al. 1997a) for the problem of superhedging in a market with convex constrained portfolios (Cvitanic et al. 1998). The case with an arbitrary closed constraint was proved in Peng (1999).

Backward Stochastic Semigroup and g -Expectations

Let $\mathcal{E}_{t,T}^g[\xi] = Y_t$ where Y is the solution of BSDE (1). $(\mathcal{E}_{t,T}^g[\cdot])_{0 \leq t \leq T < \infty}$ has the (backward) semigroup property (Peng 1997)

$$\begin{aligned}\mathcal{E}_{s,t}^g[\mathcal{E}_{t,T}^g[\xi]] &= \mathcal{E}_{s,T}^g[\xi], \quad \mathcal{E}_{T,T}^g[\xi] \\ &= \xi, \quad 0 \leq s \leq t \leq T.\end{aligned}$$

For a real valued BSDE, by the comparison theorem, the semigroup is monotone: $\mathcal{E}_{t,T}^g[\xi] \geq \mathcal{E}_{t,T}^g[\tilde{\xi}]$, if $\xi \geq \tilde{\xi}$. If moreover $g|_{z=0} = 0$, then the semigroup is constant preserving: $\mathcal{E}_{t,T}^g[c] \equiv c$. Thus the semigroup forms in fact a nonlinear expectation called g -expectation (since this nonlinear expectation is totally determined by the generator g).

This notion allows us to establish a nonlinear g -martingale theory, e.g., g -supermartingale decomposition theorem. Peng (1999) claims that, if Y is a square-integrable càdlàg g -supermartingale, then it has the unique decomposition: there exists a unique predictable, increasing, and càdlàg process A such that Y solves

$$-dY_t = g(t, Y_t, Z_t)dt + dA_t - Z_t dB_t.$$

A theoretically challenging and practically important problem is as follows: given an abstract family of expectations $(\mathcal{E}_{s,t}[\cdot])_{s \leq t}$ satisfying the same backward semigroup properties as these of g -expectation, can we find a function g such that $\mathcal{E}_{s,t} \equiv \mathcal{E}_{s,t}^g$? Coquet, Hu et al. (2005) proved that if $\mathcal{E}$ is dominated by g_μ -expectation with $g_\mu(z) = \mu|z|$ for a large enough constant $\mu > 0$, then there exists a unique function $g = g(t, \omega, z)$ satisfying μ -Lipschitz condition such that $(\mathcal{E}_{s,t}[\cdot])_{s \leq t}$ is in fact a g -expectation. For a concave dynamic expectation with an assumption much weaker than the above domination condition, we can still find a function $g = g(t, z)$ with possibly singular values (Delbaen et al. 2010). For the case without the assumption of constant preservation, see Peng (2005). In practice, the above criterion is very useful to test whether a dynamic pricing mechanism of contingent contracts can be represented through a concrete function g .

A serious challenging problem in the stochastic control theory is as follows: it is based on a given probability space $(\Omega, \mathcal{F}, P)$. But in most practical situations, it is far from being true. In many risky situations, it is necessary to consider the uncertainty of the probability measures themselves, e.g., $\{P_\theta\}_{\theta \in \Theta}$, namely, the well-known Knightian uncertainty (Knight 1921). A new framework of G -expectation space $(\Omega, \mathcal{H}, \hat{\mathbb{E}})$ and the corresponding random and stochastic analysis (Itô's analysis) is introduced (see Peng 2007, 2010 and Sonner et al. 2012) to replace the probability framework $(\Omega, \mathcal{F}, P)$. g -expectation is a special and typical case in this new theory.

Cross-References

- [Numerical Methods for Continuous-Time Stochastic Control Problems](#)
- [Risk-Sensitive Stochastic Control](#)
- [Stochastic Dynamic Programming](#)
- [Stochastic Linear-Quadratic Control](#)
- [Stochastic Maximum Principle](#)

Recommended Reading

BSDE theory applied in maximization of stochastic control can be found in the book of Yong and Zhou (1999); stochastic control problem in finance in El Karoui et al. (1997a); optimal stopping and reflected BSDE in El Karoui et al. (1997b); Maximization under Knightian uncertainty using nonlinear expectation can be found in Chen and Epstein (2002) and a survey paper in Peng (2010).

Bibliography

- Barrieu P, El Karoui N (2005) Inf-convolution of risk measures and optimal risk transfer. *Financ Stoch* 9:269–298
 Bismut JM (1973) Conjugate convex functions in optimal stochastic control. *J Math Anal Appl* 44: 384–404

- Buckdahn R, Quincampoix M, Rascanu A (2000) Viability property for a backward stochastic differential equation and applications to partial differential equations. *Probab Theory Relat Fields* 116(4): 485–504
- Chen Z, Epstein L (2002) Ambiguity, risk and asset returns in continuous time. *Econometrica* 70(4): 1403–1443
- Coquet F, Hu Y, Memin J, Peng S (2002) Filtration consistent nonlinear expectations and related g-Expectations. *Probab Theory Relat Fields* 123: 1–27
- Cvitanic J, Karatzas I (1993) Hedging contingent claims with constrained portfolios. *Ann Probab* 3(4): 652–681
- Cvitanic J, Karatzas I (1996) Backward stochastic differential equations with reflection and Dynkin games. *Ann Probab* 24(4):2024–2056
- Cvitanic J, Karatzas I, Soner M (1998) Backward stochastic differential equations with constraints on the gains-process. *Ann Probab* 26(4):1522–1551
- Delbaen F, Rosazza Gianin E, Peng S (2010) Representation of the penalty term of dynamic concave utilities. *Finance Stoch* 14:449–472
- Duffie D, Epstein L (1992) Appendix C with costis skiadas, stochastic differential utility. *Econometrica* 60(2):353–394
- El Karoui N, Quenez M-C (1995) Dynamic programming and pricing of contingent claims in an incomplete market. *SIAM Control Optim* 33(1): 29–66
- El Karoui N, Peng S, Quenez M-C (1997a) Backward stochastic differential equation in finance. *Math Financ* 7(1):1–71
- El Karoui N, Kapoudjian C, Pardoux E, Peng S, Quenez M-C (1997b) Reflected solutions of backward SDE and related obstacle problems for PDEs. *Ann Probab* 25(2):702–737
- Hamadène S, Lepeltier JP (1995) Zero-sum stochastic differential games and backward equations. *Syst Control Lett* 24(4):259–263
- Han Y, Peng S, Wu Z (2010) Maximum principle for backward doubly stochastic control systems with applications. *SIAM J Control* 48(7):4224–4241
- Hu Y, Peng S (1995) Solution of forward-backward stochastic differential-equations. *Probab Theory Relat Fields* 103(2):273–283
- Hu Y, Imkeller P, Müller M (2005) Utility maximization in incomplete markets. *Ann Appl Probab* 15(3): 1691–1712
- Knight F (1921) Risk, uncertainty and profit. Houghton Mifflin Company, Boston. (Dover, 2006)
- Ma J, Yong J (1999) Forward-backward stochastic differential equations and their applications. Lecture notes in mathematics, vol 1702. Springer, Berlin/ New York
- Ma J, Protter P, Yong J (1994) Solving forward–backward stochastic differential equations explicitly, a four step scheme. *Probab Theory Relat Fields* 98: 339–359
- Pardoux E, Peng S (1990) Adapted solution of a backward stochastic differential equation. *Syst Control Lett* 14(1):55–61
- Pardoux E, Peng S (1992) Backward stochastic differential equations and quasilinear parabolic partial differential equations, Stochastic partial differential equations and their applications. In: Proceedings of the IFIP. Lecture notes in CIS, vol 176. Springer, pp 200–217
- Peng S (1991) Probabilistic interpretation for systems of quasilinear parabolic partial differential equations. *Stochastics* 37:61–74
- Peng S (1992) A generalized dynamic programming principle and hamilton-jacobi-bellmen equation. *Stochastics* 38:119–134
- Peng S (1994) Backward stochastic differential equation and exact controllability of stochastic control systems. *Prog Nat Sci* 4(3):274–284
- Peng S (1997) BSDE and stochastic optimizations. In: Yan J, Peng S, Fang S, Wu LM (eds) Topics in stochastic analysis. Lecture notes of xiangfan summer school, chap 2. Science Publication (in Chinese, 1995)
- Peng S (1999) Monotonic limit theorem of BSDE and nonlinear decomposition theorem of Doob-Meyer's type. *Probab Theory Relat Fields* 113(4): 473–499
- Peng S (2004) Nonlinear expectation, nonlinear evaluations and risk measurs. In: Back K, Bielecki TR, Hipp C, Peng S, Schachermayer W (eds) Stochastic methods in finance lectures, C.I.M.E.-E.M.S. Summer School held in Bressanone/Brixen, LNM vol 1856. Springer, pp 143–217. (Edit. M. Frittelli and W. Runggaldier)
- Peng S (2005) Dynamically consistent nonlinear evaluations and expectations. *arXiv:math. PR/ 0501415 v1*
- Peng S (2007) G -expectation, G -Brownian motion and related stochastic calculus of Itô's type. In: Benth et al. (eds) Stochastic analysis and applications, The Abel Symposium 2005, Abel Symposia, pp 541–567. Springer
- Peng S (2010) Backward stochastic differential equation, nonlinear expectation and their applications. In: Proceedings of the international congress of mathematicians, Hyderabad
- Peng S, Shi Y (2003) A type of time-symmetric forward-backward stochastic differential equations. *C R Math Acad Sci Paris* 336:773–778
- Peng S, Wu Z (1999) Fully coupled forward-backward stochastic differential equations and applications to optimal control. *SIAM J Control Optim* 37(3): 825–843
- Rosazza Gianin E (2006) Risk measures via G -expectations. *Insur Math Econ* 39:19–34
- Soner M, Touzi N, Zhang J (2012) Wellposedness of second order backward SDEs. *Probab Theory Relat Fields* 153(1–2):149–190
- Yong J, Zhou X (1999) Stochastic control. Applications of mathematics, vol 43. Springer, New York

Basic Numerical Methods and Software for Computer Aided Control Systems Design

Volker Mehrmann¹ and Paul Van Dooren²

¹Institut für Mathematik MA 4-5, Technische Universität Berlin, Berlin, Germany

²ICTEAM: Department of Mathematical Engineering, Catholic University of Louvain, Louvain-la-Neuve, Belgium

Abstract

Basic principles for the development of computational methods for the analysis and design of linear time-invariant systems are discussed. These have been used in the design of the subroutine library SLICOT. The principles are illustrated on the basis of a method to check the controllability of a linear system.

Keywords

Accuracy · Basic numerical methods · Benchmarking · Controllability · Documentation and implementation standards · Efficiency · Software design

Introduction

Basic numerical methods for the analysis and design of dynamical systems are at the heart of most techniques in systems and control theory that are used to describe, control, or optimize industrial and economical processes. There are many methods available for all the different tasks in systems and control, but even though most of these methods are based on sound theoretical principles, many of them still fail when applied to real-life problems. The reasons for this may be quite diverse, such as the fact that the system dimensions are very large, that the underlying problem is very sensitive to small changes in the data, or that the method lacks numerical robustness when implemented in a finite precision environment.

To overcome such failures, major efforts have been made in the last few decades to develop robust, well-implemented, and standardized software packages for computer-aided control systems design (Grübel 1983; Nag Slicot 1990; Wieslander 1977). Following the standards of modern software design, such packages should consist of numerically robust routines with known performance in terms of reliability and efficiency that can be used to form the basis of more complex control methods. Also to avoid duplication and to achieve efficiency and portability to different computational environments, it is essential to make maximal use of the established standard packages that are available for numerical computations, e.g., the **Basic Linear Algebra Subroutines (BLAS)** (Dongarra et al. 1990) or the **Linear Algebra Packages (LAPACK)** (Anderson et al. 1992). On the basis of such standard packages, the next layer of more complex control methods can then be built in a robust way.

In the late 1980s, a working group was created in Europe to coordinate efforts and integrate and extend the earlier software developments in systems and control. Thanks to the support of the European Union, this eventually led to the development of the **Subroutine Library in Control Theory (SLICOT)** (Benner et al. 1999; SLICOT 2012). This library contains most of the basic computational methods for control systems design of linear time-invariant control systems.

An important feature of this and similar kind of subroutine libraries is that the development of further higher level methods is not restricted by specific requirements of the languages or data structures used and that the routines can be easily incorporated within other more user-friendly software systems (Gomez et al. 1997; MATLAB 2013). Usually, this *low-level reusability* can only be achieved by using a general-purpose programming language like C or Fortran.

We cannot present all the features of the SLICOT library here. Instead, we discuss its general philosophy in section “[The Control Subroutine Library SLICOT](#)” and illustrate these concepts in section “[An Illustration](#)” using one specific task,

namely, checking the controllability of a system. We refer to SLICOT (2012) for more details on SLICOT and to Varga (2004) for a general discussion on numerical software for systems and control.

The Control Subroutine Library SLICOT

When designing a subroutine library of basic algorithms, one should make sure that it satisfies certain basic requirements and that it follows a strict standardization in implementation and documentation. It should also contain standardized test sets that can be used for benchmarking, and it should provide means for maintenance and portability to new computing environments. The subroutine library SLICOT was designed to satisfy the following basic recommendations that are typically expected in this context (Benner et al. 1999).

Robustness: A subroutine must either return reliable results or it must return an error or warning indicator, if the problem has not been well posed or if the problem does not fall in the class to which the algorithm is applicable or if the problem is too ill-conditioned to be solved in a particular computing environment.

Numerical stability and accuracy: Subroutines are supposed to return results that are as good as can be expected when working at a given precision. They also should provide an option to return a parameter estimating the accuracy actually achieved.

Efficiency: An algorithm should never be chosen for its speed if it fails to meet the usual standards of robustness, numerical stability, and accuracy, as described above. Efficiency must be evaluated, e.g., in terms of the number of floating-point operations, the memory requirements, or the number and cost of iterations to be performed.

Modern computer architectures: The requirements of modern computer architectures must be taken into account, such as shared or distributed memory parallel processors,

which are the standard environments of today. The differences in the various architectures may imply different choices of algorithms.

Comprehensive functional coverage: The routines of the library should solve control systems relevant computational problems and try to cover a comprehensive set of routines to make it functional for a wide range of users. The SLICOT library covers most of the numerical linear algebra methods needed in systems analysis and synthesis problems for standard and generalized state space models, such as Lyapunov, Sylvester, and Riccati equation solvers, transfer matrix factorizations, similarity and equivalence transformations, structure exploiting algorithms, and condition number estimators.

The implementation of subroutines for a library should be highly standardized, and it should be accompanied by a well-written online documentation as well as a user manual (see, e.g., standard Denham and Benson 1981; Working Group Software 1996) which is compatible with that of the LAPACK library (Anderson et al. 1992). Although such highly restricted standards often put a heavy burden on the programmer, it has been observed that it has a high importance for the reusability of software and it also has turned out to be a very valuable tool in teaching students how to implement algorithms in the context of their studies.

Benchmarking

In the validation of numerical software, it is extremely important to be able to test the correctness of the implementation as well as the performance of the method, which is one of the major steps in the construction of a software library. To achieve this, one needs a standardized set of benchmark examples that allows an evaluation of a method with respect to correctness, accuracy, and efficiency and to analyze the behavior of the method in extreme situations, i.e., on problems where the limit of the possible accuracy is reached. In the context of basic systems and control methods, several such

benchmark collections have been developed (see, e.g., Benner et al. 1997; Frederick 1998, or <http://www.slicot.org/index.php?site=benchmarks>).

Maintenance, Open Access, and Archives

It is a major challenge to maintain a well-developed library accessible and usable over time when computer architectures and operating systems are changing rapidly, while keeping the library open for access to the user community. This usually requires financial resources that either have to be provided by public funding or by licensing the commercial use.

In the SLICOT library, this challenge has been addressed by the formation of the Niconet Association (<http://www.niconet-ev.info/en/>) which provides the current versions of the codes and all the documentations. Those of Release 4.5 are available under the GNU General Public License or from the archives of <http://www.slicot.org/>.

An Illustration

To give an illustration for the development of a basic control system routine, we consider the specific problem of checking controllability of a linear time-invariant control system. A linear time-invariant control problem has the form

$$\frac{dx}{dt} = Ax + Bu, \quad t \in [t_0, \infty) \quad (1)$$

Here x denotes the *state* and u the *input function*, and the system matrices are typically of the form $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$.

One of the most important topics in control is the question whether by an appropriate choice of input function $u(t)$ we can control the system from an arbitrary state to the null state. This property, called *controllability*, can be characterized by one of the following equivalent conditions (see Paige 1981).

Theorem 1 *The following are equivalent:*

- (i) *System (1) is controllable.*
- (ii) *Rank $[B, AB, A^2B, \dots, A^{n-1}B] = n$.*
- (iii) *Rank $[B, A - \lambda I] = n \quad \forall \lambda \in \mathbb{C}$.*

- (iv) *$\exists F$ such that A and $A + BF$ have no common eigenvalues.*

The conditions of Theorem 1 are nice for theoretical purposes, but none of them is really adequate for the implementation of an algorithm that satisfies the requirements described in the previous section. Condition (ii) creates difficulties because the controllability matrix $K = [B, AB, A^2B, \dots, A^{n-1}B]$ will be highly corrupted by roundoff errors. Condition (iii) can simply not be checked in finite time. However, it is sufficient to check this condition only for the eigenvalues of A , but this is extremely expensive. And finally, condition (iv) will almost always give disjoint spectra between A and $A + BF$ since the computation of eigenvalues is sensitive to roundoff.

To devise numerical procedures, one often resorts to the computation of canonical or condensed forms of the underlying system. To obtain such a form one employs controllability preserving linear transformations $x \mapsto Px$, $u \mapsto Qu$ with nonsingular matrices $P \in \mathbb{R}^{n,n}$, $Q \in \mathbb{R}^{m,m}$. The canonical form under these transformations is the Luenberger form (see Luenberger 1967). This form allows to check the controllability using the above criterion (iii) by simple inspection of the condensed matrices. This is ideal from a theoretical point of view but is very sensitive to small perturbations in the data, in particular because the transformation matrices may have arbitrary large norm, which may lead to large errors.

For the implementation as robust numerical software one uses instead transformations with real orthogonal matrices P, Q that can be implemented in a backward stable manner, i.e., the resulting backward error is bounded by a small constant times the unit roundoff u of the finite precision arithmetic, and employs for reliable rank determinations the well-known singular value decomposition (SVD) (see, e.g., Golub and Van Loan 1996).

Theorem 2 (Singular value decomposition) *Given $A \in \mathbb{R}^{n,m}$, then there exist orthogonal matrices U, V with $U \in \mathbb{R}^{n,n}$, $V \in \mathbb{R}^{m,m}$,*

such that $A = U\Sigma V^T$ and $\Sigma \in \mathbb{R}^{n,m}$ is quasi-diagonal, i.e.,

$$\Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \text{ where } \Sigma_r = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix},$$

and the nonzero singular values σ_i are ordered as $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$.

The SVD presents the *best* way to determine (numerical) ranks of matrices in finite precision arithmetic by counting the number of singular values satisfying $\sigma_j \geq \mathbf{u}\sigma_1$ and by putting those for which $\sigma_j < \mathbf{u}\sigma_1$ equal to zero. The computational method for the SVD is well established and analyzed, and it has been implemented in the LAPACK routine SGESVD (see <http://www.netlib.org/lapack/>). A faster but less reliable alternative to compute the numerical rank of a matrix A is its QR factorization with pivoting (see, e.g., Golub and Van Loan 1996).

Theorem 3 (QRE decomposition) Given $A \in \mathbb{R}^{n,m}$, then there exists an orthogonal matrix $Q \in \mathbb{R}^{n,n}$ and a permutation $E \in \mathbb{R}^{m,m}$, such that $A = QRE^T$ and $R \in \mathbb{R}^{n,m}$ is trapezoidal, i.e.,

$$R = \begin{bmatrix} r_{11} & \dots & r_{1l} & \dots & r_{1m} \\ & \ddots & & & \vdots \\ & & r_{ll} & \dots & r_{lm} \\ 0 & & & 0 & \end{bmatrix}.$$

and the nonzero diagonal entries r_{ii} are ordered as $r_{11} \geq \dots \geq r_{ll} > 0$.

The (numerical) rank in this case is again obtained by counting the diagonal elements $r_{ii} \geq \mathbf{u}r_{11}$.

One can use such orthogonal transformations to construct the controllability staircase form (see Van Dooren 1981).

Theorem 4 (Staircase form) Given matrices $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$, then there exist orthogonal matrices P, Q with $P \in \mathbb{R}^{n,n}$, $Q \in \mathbb{R}^{m,m}$, so that

$$\begin{aligned} PAP^T &= \left[\begin{array}{cccc|c} A_{11} & \dots & \dots & A_{1,r-1} & A_{1,r} \\ A_{21} & \ddots & & \vdots & \vdots \\ & \ddots & & \vdots & \vdots \\ & & A_{r-1,r-2} & A_{r-1,r-1} & A_{r-1,r} \\ 0 & \dots & 0 & 0 & A_{rr} \end{array} \right] \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_{r-1} \\ n_r \end{matrix} \\ PBQ &= \left[\begin{array}{cc} B_1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ \vdots & \vdots \\ 0 & 0 \end{array} \right] \begin{matrix} n_1 \\ n_2 \\ \vdots \\ \vdots \\ n_r \\ n_1 \quad m - n_1 \end{matrix} \end{aligned} \quad (2)$$

where $n_1 \geq n_2 \geq \dots \geq n_{r-1} \geq n_r \geq 0$, $n_{r-1} > 0$, $A_{i,i-1} = [\Sigma_{i,i-1} \ 0]$, with nonsingular blocks $\Sigma_{i,i-1} \in \mathbb{R}^{n_i, n_i}$ and $B_1 \in \mathbb{R}^{n_1, n_1}$.

Notice that when using the reduced pair in condition (iii) of Theorem 1, the controllability condition is just $n_r = 0$, which is simply checked by inspection. A numerically stable algorithm to compute the staircase form of Theorem 4 is given below. It is based on the use of the singular value decomposition, but one could also have used instead the QR decomposition with column pivoting.

Staircase Algorithm

Input: $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$

Output: PAP^T, PBQ in the form (2), P, Q orthogonal

Step 0: Perform an SVD $B = U_B \begin{bmatrix} \Sigma_B & 0 \\ 0 & 0 \end{bmatrix} V_B^T$ with nonsingular and diagonal $\Sigma_B \in \mathbb{R}^{n_1, n_1}$. Set $P := U_B^T$, $Q := V_B$, so that

$$A := U_B^T A U_B = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

$$B := U_B^T B V_B = \begin{bmatrix} \Sigma_B & 0 \\ 0 & 0 \end{bmatrix}$$

with A_{11} of size $n_1 \times n_1$.

Step 1: Perform an SVD $A_{21} = U_{21} \begin{bmatrix} \Sigma_{21} & 0 \\ 0 & 0 \end{bmatrix} V_{21}^T$ with nonsingular and diagonal $\Sigma_{21} \in \mathbb{R}^{n_2, n_2}$. Set

$$P_2 := \begin{bmatrix} V_{21}^T & 0 \\ 0 & U_{21}^T \end{bmatrix}, \quad P := P_2 P$$

so that

$$A := P_2 A P_2^T =: \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix},$$

$$B := P_2 B =: \begin{bmatrix} B_1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

where $A_{21} = [\Sigma_{21} \ 0]$, and $B_1 := V_{21}^T \Sigma_B$.

Step 2:

$i = 3$

DO WHILE ($n_{i-1} > 0$ AND $A_{i,i-1} \neq 0$).

 Perform an SVD of $A_{i,i-1} = U_{i,i-1}$

$$\begin{bmatrix} \Sigma_{i,i-1} & 0 \\ 0 & 0 \end{bmatrix} V_{i,i-1}^T \text{ with}$$

$\Sigma_{i,i-1} \in \mathbb{R}^{n_i, n_i}$ nonsingular and diagonal.

Set

$$P_i := \begin{bmatrix} I_{n_1} & & & \\ & \ddots & & \\ & & I_{n_{i-2}} & \\ & & & V_{i,i-1}^T \\ & & & & U_{i,i-1}^T \end{bmatrix}, P := P_i P,$$

so that

$$A := P_i A P_i^T =: \begin{bmatrix} A_{11} & \cdots & A_{1,i+1} \\ A_{21} & \ddots & A_{2,i+1} \\ & \ddots & \vdots \\ & & A_{i,i-1} & A_{i,i} & \vdots \\ 0 & & A_{i+1,i} & A_{i+1,i+1} \end{bmatrix}$$

where $A_{i,i-1} = [\Sigma_{i,i-1} \ 0]$.

$i := i + 1$

END

$r := i$

It is clear that this algorithm will stop with $n_i = 0$ or $A_{i,i-1} = 0$. In every step, the remaining block shrinks at least by 1 row/column, as long as $\text{Rank } A_{i,i-1} > 1$, so that the algorithm stops after maximally $n - 1$ steps. It has been shown in Van Dooren (1981) that system (1) is controllable if and only if in the staircase form of (A, B) one has $n_r = 0$.

It should be noted that the updating transformations P_i of this algorithm will affect previously created “stairs” so that the blocks denoted as $\Sigma_{i,i-1}$ will not be diagonal anymore, but their singular values are unchanged. This is critical in the decision about the controllability of the pair (A, B) since it depends on the numerical rank of the submatrices $A_{i,i-1}$ and B (see Demmel and Kågström 1993). Based on this and a detailed error and perturbation analysis, the Staircase Algorithm has been implemented in the SLICOT routine AB01OD, and it uses in the worst-case $\mathcal{O}(n^4)$ flops (a “flop” is an elementary floating-point operation $+$, $-$, $*$, or $/$). For efficiency reasons, the SLICOT routine AB01OD does not use SVDs for rank decisions, but QR decompositions with column pivoting. When applying the corresponding orthogonal transformations to the system without accumulating them, the complexity can be reduced to $\mathcal{O}(n^3)$ flops. It has been provided with error bounds, condition estimates, and warning strategies.

Summary and Future Directions

We have presented the SLICOT library and the basic principles for the design of such basic subroutine libraries. To illustrate these principles, we have presented the development of a method for checking controllability for a linear time-invariant control system. But the SLICOT library contains much more than that. It essentially covers most of the problems listed in the selected reprint volume (Patel et al. 1994). This volume contained in 1994 the state of the art in numerical methods for systems and control, but the field has strongly evolved since then. Examples of areas that were not in this volume but that are included in SLICOT are periodic systems, differential algebraic equations, and model reduction. Areas which still need new results and software are the control of large-scale systems, obtained either from discretizations of partial differential equations or from the interconnection of a large number of interacting systems. But it is unclear for the moment which will be the

methods of choice for such problems. We still need to understand the numerical challenges in such areas, before we can propose numerically reliable software for these problems: the area is still quite open for new developments.

Cross-References

- [Computer-Aided Control Systems Design: Introduction and Historical Overview](#)
- [Interactive Environments and Software Tools for CACSD](#)

Bibliography

- Anderson E, Bai Z, Bischof C, Demmel J, Dongarra J, Du Croz J, Greenbaum A, Hammarling S, McKenney A, Ostrouchov S, Sorensen D (1995) LAPACK users' guide, 2nd edn. SIAM, Philadelphia. <http://www.netlib.org/lapack/>
- Benner P, Laub AJ, Mehrmann V (1997) Benchmarks for the numerical solution of algebraic Riccati equations. *Control Syst Mag* 17:18–28
- Benner P, Mehrmann V, Sima V, Van Huffel S, Varga A (1999) SLICOT-A subroutine library in systems and control theory. *Appl Comput Control Signals Circuits* 1:499–532
- Demmel JW, Kågström B (1993) The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: robust software with error bounds and applications. Part I: theory and algorithms. *ACM Trans Math Softw* 19:160–174
- Denham MJ, Benson CJ (1981) Implementation and documentation standards for the software library in control engineering (SLICE). Technical report 81/3, Kingston Polytechnic, Control Systems Research Group, Kingston
- Dongarra JJ, Du Croz J, Duff IS, Hammarling S (1990) A set of level 3 basic linear algebra subprograms. *ACM Trans Math Softw* 16:1–17
- Frederick DK (1988) Benchmark problems for computer aided control system design. In: *Proceedings of the 4th IFAC symposium on computer-aided control systems design*, Beijing, pp 1–6
- Golub GH, Van Loan CF (1996) *Matrix computations*, 3rd edn. The Johns Hopkins University Press, Baltimore
- Gomez C, Bunks C, Chancelior J-P, Delebecque F (1997) *Integrated scientific computing with scilab*. Birkhäuser, Boston. <https://www.scilab.org/>
- Grübel G (1983) Die regelungstechnische Programmbibliothek RASP. *Regelungstechnik* 31: 75–81
- Luenberger DG (1967) Canonical forms for linear multivariable systems. *IEEE Trans Autom Control* 12(3):290–293
- Paige CC (1981) Properties of numerical algorithms related to computing controllability. *IEEE Trans Autom Control* AC-26:130–138
- Patel R, Laub A, Van Dooren P (eds) (1994) *Numerical linear algebra techniques for systems and control*. IEEE, Piscataway
- The Control and Systems Library SLICOT (2012) The NICONET society. NICONET e.V. <http://www.niconet-ev.info/en/>
- The MathWorks, Inc. (2013) MATLAB version 8.1. The MathWorks, Inc., Natick
- The Numerical Algorithms Group (1993) NAG SLICOT library manual, release 2. The Numerical Algorithms Group, Wilkinson House, Oxford. Updates Release 1 of May 1990
- The Working Group on Software (1996) SLICOT implementation and documentation standards 2.1. WGS-report 96-1. <http://www.icm.tu-bs.de/NICONET/reports.html>
- Van Dooren P (1981) The generalized eigenstructure problem in linear system theory. *IEEE Trans Autom Control* AC-26:111–129
- Varga A (ed) (2004) Special issue on numerical awareness in control. *Control Syst Mag* 24-1: 14–17
- Wieslander J (1977) *Scandinavian control library*. A subroutine library in the field of automatic control. Technical report, Department of Automatic Control, Lund Institute of Technology, Lund

Bilinear Control of Schrödinger PDEs

Karine Beauchard¹ and Pierre Rouchon²

¹CNRS, CMLS, Ecole Polytechnique, Palaiseau, France

²Centre Automatique et Systèmes, Mines ParisTech, Paris Cedex 06, France

Abstract

This entry is an introduction to modern issues about controllability of Schrödinger PDEs with bilinear controls. This model is pertinent for a quantum particle, controlled by an electric field. We review recent developments in the field, with discrimination between exact and approximate controllabilities, in finite or

infinite time. We also underline the variety of mathematical tools used by various teams in the last decade. The results are illustrated on several classical examples.

Keywords

Approximate controllability · Global exact controllability · Local exact controllability · Quantum particles · Schrödinger equation · Small-time controllability

Introduction

A quantum particle, in a space with dimension N ($N = 1, 2, 3$), in a potential $V = V(x)$, and in an electric field $u = u(t)$, is represented by a wave function $\psi : (t, x) \in \mathbb{R} \times \Omega \rightarrow \mathbb{C}$ on the $L^2(\Omega, \mathbb{C})$ sphere S

$$\int_{\Omega} |\psi(t, x)|^2 dx = 1, \quad \forall t \in \mathbb{R},$$

where $\Omega \subset \mathbb{R}^N$ is a possibly unbounded open domain. In first approximation, the time evolution of the wave function is given by the Schrödinger equation,

$$\begin{cases} i \partial_t \psi(t, x) = (-\Delta + V) \psi(t, x) \\ -u(t) \mu(x) \psi(t, x), \quad t \in (0, +\infty), x \in \Omega, \\ \psi(t, x) = 0, \quad x \in \partial\Omega \end{cases} \quad (1)$$

where μ is the dipolar moment of the particle and $\hbar = 1$ here. Sometimes, this equation is considered in the more abstract framework

$$i \frac{d}{dt} \psi = (H_0 + u(t)H_1) \psi \quad (2)$$

where ψ lives on the unit sphere of a separable Hilbert space $\mathcal{H}$ and the Hamiltonians H_0, H_1 are Hermitian operators on $\mathcal{H}$. A natural question, with many practical applications, is the existence of a control u that steers the wave function ψ from a given initial state ψ_0 , to a prescribed target ψ_f .

The goal of this survey is to present well-established results concerning exact and approximate controllabilities for the bilinear control system (1), with applications to relevant

examples. The main difficulties are the infinite dimension of $\mathcal{H}$ and the nonlinearity of the control system.

Preliminary Results

When the Hilbert space $\mathcal{H}$ has finite dimension n , then controllability of Eq. (2) is well understood (D'Alessandro 2008). If, for example, the Lie algebra spanned by H_0 and H_1 coincides with $u(n)$, the set of skew-Hermitian matrices, then system (2) is globally controllable: for any initial and final states $\psi_0, \psi_f \in \mathcal{H}$ of length one, there exist $T > 0$ and a bounded open-loop control $[0, T] \ni t \mapsto u(t)$ steering ψ from $\psi(0) = \psi_0$ to $\psi(T) = \psi_f$.

In infinite dimension, this idea served to intuit a negative controllability result in Mirrahimi and Rouchon (2004), but the above characterization cannot be generalized because iterated Lie brackets of unbounded operators are not necessarily well defined. For example, the quantum harmonic oscillator

$$\begin{aligned} i \partial_t \psi(t, x) = & -\partial_x^2 \psi(t, x) + x^2 \psi(t, x) \\ & - u(t) x \psi(t, x), \quad x \in \mathbb{R}, \end{aligned} \quad (3)$$

is not controllable (in any reasonable sense) (Mirrahimi and Rouchon 2004) even if all its Galerkin approximations are controllable (Fu et al. 2001). Thus, much care is required in the use of Galerkin approximations to prove controllability in infinite dimension. This motivates the search of different methods to study exact controllability of bilinear PDEs of form (1).

In infinite dimension, the norms need to be specified. In this article, we use Sobolev norms. For $s \in \mathbb{N}$, the Sobolev space $H^s(\Omega)$ is the space of functions $\psi: \Omega \rightarrow \mathbb{C}$ with square integrable derivatives $d^k \psi$ for $k = 0, \dots, s$ (derivatives are well defined in the distribution sense). $H^s(\Omega)$ is endowed with the norm $\|\psi\|_{H^s} := \left(\sum_{k=0}^s \|d^k \psi\|_{L^2(\Omega)}^2 \right)^{1/2}$. We also use the space $H_0^1(\Omega)$ which contains functions $\psi \in H^1(\Omega)$ that

vanish on the boundary $\partial\Omega$ (in the trace sense) (Brézis 1999).

The first control result of the literature states the noncontrollability of system (1) in $(H^2 \cap H_0^1)(\Omega) \cap S$ with controls $u \in L^2((0, T), \mathbb{R})$ (Turinici 2000; Ball et al. 1982). More precisely, by applying $L^2(0, T)$ controls u , the reachable wave functions $\psi(T)$ form a subset of $(H^2 \cap H_0^1)(\Omega) \cap S$ with empty interior. This statement does not give obstructions for system (1) to be controllable in different functional spaces as we will see below, but it indicates that controllability issues are much more subtle in infinite dimension than in finite dimension.

Local Exact Controllability

In 1D and with Discrete Spectrum

This section is devoted to the 1D PDE:

$$\begin{cases} i \partial_t \psi(t, x) = -\partial_x^2 \psi(t, x) \\ -u(t) \mu(x) \psi(t, x), \quad x \in (0, 1), t \in (0, T), \\ \psi(t, 0) = \psi(t, 1) = 0. \end{cases} \quad (4)$$

We call “ground state” the solution of the free system ($u = 0$) built with the first eigenvalue and eigenvector of $-\partial_x^2$: $\psi_1(t, x) = \sqrt{2} \sin(\pi x) e^{-i\pi^2 t}$. Under appropriate assumptions on the dipolar moment μ , then system (4) is controllable around the ground state, locally in $H_0^3(0, 1) \cap S$, with controls in $L^2((0, T), \mathbb{R})$, as stated below.

Theorem 1 Assume $\mu \in H^3((0, 1), \mathbb{R})$ and

$$\left| \int_0^1 \mu(x) \sin(\pi x) \sin(k\pi x) dx \right| \geq \frac{c}{k^3}, \quad \forall k \in \mathbb{N}^* \quad (5)$$

for some constant $c > 0$. Then, for every $T > 0$, there exists $\delta > 0$ such that for every $\psi_0, \psi_f \in S \cap H_0^3((0, 1), \mathbb{C})$ with $\|\psi_0 - \psi_1(0)\|_{H^3} + \|\psi_f - \psi_1(T)\|_{H^3} < \delta$, there exists $u \in L^2((0, T), \mathbb{R})$ such that the solution of (4) with initial condition $\psi(0, x) = \psi_0(x)$ satisfies $\psi(T) = \psi_f$.

Here, $H_0^3(0, 1) := \{\psi \in H^3((0, 1), \mathbb{C}); \psi = \psi'' = 0 \text{ at } x = 0, 1\}$. We refer to Beauchard and Laurent (2010) and Beauchard et al. (2013) for proof and generalizations to nonlinear PDEs. The proof relies on the linearization principle, by applying the classical inverse mapping theorem to the endpoint map. Controllability of the linearized system around the ground state is a consequence of assumption (5) and classical results about trigonometric moment problems. A subtle smoothing effect allows to prove C^1 regularity of the endpoint map.

The assumption (5) holds for generic $\mu \in H^3((0, 1), \mathbb{R})$ and plays a key role for local exact controllability to hold in *small* time T . In Beauchard and Morancey (2014), local exact controllability is proved under the weaker assumption, namely, $\mu'(0) \pm \mu'(1) \neq 0$, but only in *large* time T .

Moreover, under appropriate assumptions on μ , references Coron (2006) and Beauchard and Morancey (2014) propose explicit motions that are impossible in small time T , with small controls in L^2 . Thus, a positive minimal time is required for local exact controllability, even if information propagates at infinite speed. This minimal time is due to nonlinearities; its characterization is an open problem.

Actually, assumption $\mu'(0) \pm \mu'(1) \neq 0$ is not necessary for local exact controllability in large time. For instance, the quantum box, i.e.,

$$\begin{cases} i \partial_t \psi(t, x) = -\partial_x^2 \psi(t, x) \\ -u(t)x \psi(t, x), \quad x \in (0, 1), \\ \psi(t, 0) = \psi(t, 1) = 0, \end{cases} \quad (6)$$

is treated in Beauchard (2005). Of course, these results are proved with additional techniques: power series expansions and Coron’s return method (Coron 2007).

There is no contradiction between the negative result of section “Preliminary Results” and the positive result of Theorem 1. Indeed, the wave function cannot be steered between any two points ψ_0, ψ_f of $H^2 \cap H_0^1$, but it can be steered between any two points ψ_0, ψ_f of

$H_{(0)}^3$, which is smaller than $H^2 \cap H_0^1$. In particular, $H_{(0)}^3((0, 1), \mathbb{C})$ has an empty interior in $H_{(0)}^2((0, 1), \mathbb{C})$. Thus, there is no incompatibility between the reachable set to have empty interior in $H^2 \cap H_0^1$ and the reachable set to coincide with $H_{(0)}^3$.

Open Problems in Multi-D or with Continuous Spectrum

The linearization principle used to prove Theorem 1 does not work in multi-D: the trigonometric moment problem, associated to the controllability of the linearized system, cannot be solved. Indeed, its frequencies, which are the eigenvalues of the Dirichlet Laplacian operator, do not satisfy a required gap condition (Loreti and Komornik 2005).

The study of a toy model (Beauchard 2011) suggests that if local controllability holds in 2D (with a priori bounded L^2 -controls) then a positive minimal time is required, whatever μ is. The appropriate functional frame for such a result is an open problem.

In 3D or in the presence of continuous spectrum, we conjecture that local exact controllability does not hold (with a priori bounded L^2 -controls) because the gap condition in the spectrum of the Dirichlet Laplacian operator is violated (see Beauchard et al. (2010) for a toy model from nuclear magnetic resonance and ensemble controllability as originally stated in Li and Khaneja (2009)). Thus, exact controllability should be investigated with controls that are not a priori bounded in L^2 ; this requires new techniques. We refer to Nersesyan and Nersisyan (2012a) for precise negative results.

Finally, we emphasize that exact controllability in multi-D but in *infinite* time has been proved in Nersesyan and Nersisyan (2012a, b), with techniques similar to one used in the proof of Theorem 1.

Approximate Controllability

Different approaches have been developed to prove approximate controllability.

Lyapunov Techniques

Due to measurement effect and back action, closed-loop controls in the Schrödinger frame are not appropriate. However, closed-loop controls may be computed via numerical simulations and then applied to real quantum systems in open loop, without measurement. Then, the strategy consists in designing damping feedback laws, thanks to a controlled Lyapunov function, which encodes the distance to the target. In finite dimension, the convergence proof relies on LaSalle invariance principle. In infinite dimension, this principle works when the trajectories of the closed-loop system are compact (in the appropriate space), which is often difficult to prove. Thus, two adaptations have been proposed: approximate convergence (Beauchard and Mirrahimi 2009; Mirrahimi 2009) and weak convergence (Beauchard and Nersesyan 2010) to the target.

Variational Methods and Global Exact Controllability

The global approximate controllability of (1), in any Sobolev space, is proved in Nersesyan (2010), under generic assumptions on (V, μ) , with Lyapunov techniques and variational arguments.

Theorem 2 *Let $V, \mu \in C^\infty(\bar{\Omega}, \mathbb{R})$ and $(\lambda_j)_{j \in \mathbb{N}^*}, (\phi_j)_{j \in \mathbb{N}^*}$ be the eigenvalues and normalized eigenvectors of $(-\Delta + V)$. Assume $\langle \mu \phi_j, \phi_1 \rangle \neq 0$, for all $j \geq 2$ and $\lambda_1 - \lambda_j \neq \lambda_p - \lambda_q$ for all $j, p, q \in \mathbb{N}^*$ such that $\{1, j\} \neq \{p, q\}; j \neq 1$. Then, for every $s > 0$, the system (1) is globally approximately controllable in $H_{(V)}^s : D[(-\Delta + V)^{s/2}]$, the domain of $(-\Delta + V)^{s/2}$: for every $\epsilon, \delta > 0$ and $\psi_0 \in S \cap H_{(V)}^s$, there exist a time $T > 0$ and a control $u \in C_0^\infty((0, T), \mathbb{R})$ such that the solution of (1) with initial condition $\psi(0) = \psi_0$ satisfies $\|\psi(T) - \phi_1\|_{H_{(V)}^{s-\delta}} < \epsilon$.*

This theorem is of particular importance. Indeed, in 1D and for appropriate choices of (V, μ) , global exact controllability of (1) in H^{3+} can be proved by combining the following:

- Global approximate controllability in H^3 given by Theorem 2,
- Local exact controllability in H^3 given by Theorem 1,
- Time reversibility of the Schrödinger equation (i.e., if $(\psi(t, x), u(t))$ is a trajectory, then so is $(\psi^*(T - t, x), u(T - t))$ where ψ^* is the complex conjugate of ψ).

Let us expose this strategy on the quantum box (6). First, one can check the assumptions of Theorem 2 with $V(x) = \gamma x$ and $\mu(x) = (1 - \gamma)x$ when $\gamma > 0$ is small enough. This means that, in (6), we consider controls $u(t)$ of the form $\gamma + u(t)$. Thus, an initial condition $\psi_0 \in H_{(0)}^{3+}$ can be steered arbitrarily close to the first eigenvector $\phi_{1,\gamma}$ of $(-\partial_x^2 + \gamma x)$, in H^3 norm. Moreover, by a variant of Theorem 1, the local exact controllability of (6) holds in $H_{(0)}^3$ around $\phi_{1,\gamma}$. Therefore, the initial condition $\psi_0 \in H_{(0)}^{3+}$ can be steered exactly to $\phi_{1,\gamma}$ in finite time. By the time reversibility of the Schrödinger equation, we can also steer exactly the solution from $\phi_{1,\gamma}$ to any target $\psi_f \in H^{3+}$. Therefore, the solution can be steered exactly from any initial condition $\psi_0 \in H_{(0)}^{3+}$ to any target $\psi_f \in H_{(0)}^{3+}$ in finite time.

Geometric Techniques Applied to Galerkin Approximations

In Boscain et al. (2012, 2013) and Chambrion et al. (2009) the authors study the control of Schrödinger PDEs, in the abstract form (2) and under technical assumptions on the (unbounded) operators H_0 and H_1 that ensure the existence of solutions with piecewise constant controls u :

1. H_0 is skew-adjoint on its domain $D(H_0)$.
2. There exists a Hilbert basis $(\phi_k)_{k \in \mathbb{N}}$ of $\mathcal{H}$ made of eigenvectors of $H_0 : H_0 \phi_k = i\lambda_k \phi_k$ and $\phi_k \in D(H_1), \forall k \in \mathbb{N}$.
3. $H_0 + uH_1$ is essentially skew-adjoint (not necessarily with domain $D(H_0)$) for every $u \in [0, \delta]$ for some $\delta > 0$.

4. $\langle H_1 \phi_j, \phi_k \rangle = 0$ for every $j, k \in \mathbb{N}$ such that $\lambda_j = \lambda_k$ and $j \neq k$.

Theorem 3 Assume that, for every $j, k \in \mathbb{N}$, there exists a finite number of integers $p_1, \dots, p_r \in \mathbb{N}$ such that

$$p_1 = j, \quad p_r = k, \quad \langle H_1 \phi_{p_l}, \phi_{p_{l+1}} \rangle \neq 0, \quad \forall l = 1, \dots, r-1$$

$|\lambda_L - \lambda_M| \neq |\lambda_{p_l} - \lambda_{p_{l+1}}|, \forall 1 \leq l \leq r-1, LM \in \mathbb{N}$ with $\{L, M\} \neq \{p_l, p_{l+1}\}$.

Then for every $\epsilon > 0$ and ψ_0, ψ_f in the unit sphere of $\mathcal{H}$, there exists a piecewise constant function $u : [0, T_\epsilon] \rightarrow [0, \delta]$ such that the solution of (2) with initial condition $\psi(0) = \psi_0$ satisfies $\|\psi(T_\epsilon) - \psi_f\|_{\mathcal{H}} < \epsilon$.

We refer to Boscain et al. (2012, 2013) and Chambrion et al. (2009) for proof and additional results such as estimates on the L^1 norm of the control. Note that H_0 is not necessarily of the form $(-\Delta + V)$, H_1 can be unbounded, δ may be arbitrary small, and the two assumptions are generic with respect to (H_0, H_1) . The connectivity and transition frequency conditions in Theorem 3 mean physically that each pair of H_0 eigenstates is connected via a finite number of first-order (one-photon) transitions and that the transition frequencies between pairs of eigenstates are all different.

Note that, contrary to Theorems 2, Theorem 3 cannot be combined with Theorem 1 to prove global exact controllability. Indeed, functional spaces are different: $\mathcal{H} = L^2(\Omega)$ in Theorem 3, whereas H^3 -regularity is required for Theorem 1.

This kind of results applies to several relevant examples such as the control of a particule in a quantum box by an electric field (6) and the control of the planar rotation of a linear molecule by means of two electric fields:

$$i \partial_t \psi(t, \theta) = \left(-\partial_\theta^2 + u_1(t) \cos(\theta) + u_2(t) \sin(\theta) \right) \psi(t, \theta), \quad \theta \in \mathbb{T}$$

where $\mathbb{T}$ is the 1D-torus. However, several other systems of physical interest are not covered by these results such as trapped ions modeled by two coupled quantum harmonic oscillators. In Ervedoza and Puel (2009), specific methods have been used to prove their approximate controllability.

Concluding Remarks

The variety of methods developed by different authors to characterize controllability of Schrödinger PDEs with bilinear control is the sign of a rich structure and subtle nature of control issues. New methods will probably be necessary to answer the remaining open problems in the field.

This survey is far from being complete. In particular, we do not consider numerical methods to derive the steering control such as those used in NMR (Nielsen et al. 2010) to achieve robustness versus parameter uncertainties or such as monotone algorithms (Baudouin and Salomon 2008; Liao et al. 2011) for optimal control (Cancès et al. 2000). We do not consider also open quantum systems where the state is then the density operator ρ , a nonnegative Hermitian operator with unit trace on $\mathcal{H}$. The Schrödinger equation is then replaced by the Lindblad equation:

$$\begin{aligned} \frac{d}{dt}\rho = & -i[H_0 + uH_1, \rho] + \sum_v L_v \rho L_v^\dagger \\ & - \frac{1}{2} \left(L_v^\dagger L_v \rho + \rho L_v^\dagger L_v \right) \end{aligned}$$

with operator L_v related to the decoherence channel v . Even in the case of finite dimensional Hilbert space $\mathcal{H}$, controllability of such system is not yet well understood and characterized (see Altafini (2003) and Kurniawan et al. (2012)).

Cross-References

- [Control of Quantum Systems](#)
- [Robustness Issues in Quantum Control](#)

Acknowledgments The authors were partially supported by the “Agence Nationale de la Recherche” (ANR), Projet Blanc EMAQS number ANR-2011-BS01-017-01.

Bibliography

- Altafini C (2003) Controllability properties for finite dimensional quantum Markovian master equations. *J Math Phys* 44(6):2357–2372
- Ball JM, Marsden JE, Slemrod M (1982) Controllability for distributed bilinear systems. *SIAM J Control Optim* 20:575–597
- Baudouin L, Salomon J (2008) Constructive solutions of a bilinear control problem for a Schrödinger equation. *Syst Control Lett* 57(6):453–464
- Beauchard K (2005) Local controllability of a 1-D Schrödinger equation. *J Math Pures Appl* 84:851–956
- Beauchard K (2011) Local controllability and non controllability for a 1D wave equation with bilinear control. *J Diff Equ* 250:2064–2098
- Beauchard K, Laurent C (2010) Local controllability of 1D linear and nonlinear Schrödinger equations with bilinear control. *J Math Pures Appl* 94(5):520–554
- Beauchard K, Mirrahimi M (2009) Practical stabilization of a quantum particle in a one-dimensional infinite square potential well. *SIAM J Control Optim* 48(2):1179–1205
- Beauchard K, Morancey M (2014) Local controllability of 1D Schrödinger equations with bilinear control and minimal time, vol 4. *Mathematical Control and Related Fields*
- Beauchard K, Nersisyan V (2010) Semi-global weak stabilization of bilinear Schrödinger equations. *CRAS* 348(19–20):1073–1078
- Beauchard K, Coron J-M, Rouchon P (2010) Controllability issues for continuous spectrum systems and ensemble controllability of Bloch equations. *Commun Math Phys* 290(2):525–557
- Beauchard K, Lange H, Teismann H (2013, preprint) Local exact controllability of a Bose-Einstein condensate in a 1D time-varying box. [arXiv:1303.2713](#)
- Boscain U, Caponigro M, Chambrion T, Sigalotti M (2012) A weak spectral condition for the controllability of the bilinear Schrödinger equation with application to the control of a rotating planar molecule. *Commun Math Phys* 311(2):423–455
- Boscain U, Chambrion T, Sigalotti M (2013) On some open questions in bilinear quantum control. [arXiv:1304.7181](#)
- Brézis H (1999) *Analyse fonctionnelles: théorie et applications*. Dunod, Paris
- Cancès E, Le Bris C, Pilot M (2000) Contrôle optimal bilinéaire d’une équation de Schrödinger. *CRAS Paris* 330:567–571

- Chambrion T, Mason P, Sigalotti M, Boscain M (2009) Controllability of the discrete-spectrum Schrödinger equation driven by an external field. *Ann Inst Henri Poincaré Anal Nonlinéaire* 26(1): 329–349
- Coron J-M (2006) On the small-time local controllability of a quantum particle in a moving one-dimensional infinite square potential well. *C R Acad Sci Paris I* 342:103–108
- Coron J-M (2007) Control and nonlinearity. Mathematical surveys and monographs, vol 136. American Mathematical Society, Providence
- D'Alessandro D (2008) Introduction to quantum control and dynamics. Applied mathematics and nonlinear science. Chapman & Hall/CRC, Boca Raton
- Ervedoza S, Puel J-P (2009) Approximate controllability for a system of Schrödinger equations modeling a single trapped ion. *Ann Inst Henri Poincaré Anal Nonlinéaire* 26(6):2111–2136
- Fu H, Schirmer SG, Solomon AI (2001) Complete controllability of finite level quantum systems. *J Phys A* 34(8):1678–1690
- Kurniawan I, Dirr G, Helmke U (2012) Controllability aspects of quantum dynamics: unified approach for closed and open systems. *IEEE Trans Autom Control* 57(8):1984–1996
- Li JS, Khaneja N (2009) Ensemble control of Bloch equations. *IEEE Trans Autom Control* 54(3): 528–536
- Liao S-K, Ho T-S, Chu S-I, Rabitz HH (2011) Fast-kick-off monotonically convergent algorithm for searching optimal control fields. *Phys Rev A* 84(3): 031401
- Loreti P, Komornik V (2005) Fourier series in control theory. Springer, New York
- Mirrahimi M (2009) Lyapunov control of a quantum particle in a decaying potential. *Ann Inst Henri Poincaré (c) Nonlinear Anal* 26:1743–1765
- Mirrahimi M, Rouchon P (2004) Controllability of quantum harmonic oscillators. *IEEE Trans Autom Control* 49(5):745–747
- Nersesyan V (2010) Global approximate controllability for Schrödinger equation in higher Sobolev norms and applications. *Ann IHP Nonlinear Anal* 27(3): 901–915
- Nersisyan V, Nersisyan H (2012a) Global exact controllability in infinite time of Schrödinger equation. *J Math Pures Appl* 97(4):295–317
- Nersisyan V, Nersisyan H (2012b) Global exact controllability in infinite time of Schrödinger equation: multidimensional case. Preprint: arXiv:1201.3445
- Nielsen NC, Kehlet C, Glaser SJ and Khaneja N (2010) Optimal Control Methods in NMR Spectroscopy. eMagRes
- Turinici G (2000) On the controllability of bilinear quantum systems. In: Le Bris C, Defranceschi M (eds) Mathematical models and methods for ab initio quantum chemistry. Lecture notes in chemistry, vol 74. Springer

Boundary Control of 1-D Hyperbolic Systems

Georges Bastin¹ and Jean-Michel Coron²

¹Department of Mathematical Engineering, ICTEAM, UCLouvain, Louvain-La-Neuve, Belgium

²Laboratoire Jacques-Louis Lions, Sorbonne Université, Paris, France

Abstract

One-dimensional hyperbolic systems are commonly used to describe the evolution of various physical systems. For many of these systems, controls are available on the boundary. There are then two natural questions: controllability (steer the system from a given state to a desired target) and stabilization (construct feedback laws leading to a good behavior of the closed-loop system around a given set point).

Keywords

Hyperbolic systems · Controllability · Stabilization · Electrical lines · Open channels · Road traffic · Chromatography

One-Dimensional Hyperbolic Systems

The operation of many physical systems may be represented by hyperbolic systems in one space dimension. These systems are described by the following partial differential equation:

$$\begin{aligned} Y_t + F(Y)Y_x + G(Y) &= 0, \\ t \in [0, T], \quad x \in [0, L], \end{aligned} \quad (1)$$

where

- t and x are two independent variables: a time variable $t \in [0, T]$ and a space variable $x \in [0, L]$ over a finite interval;

- $Y : [0, T] \times [0, L] \rightarrow \mathbb{R}^n$ is the vector of state variables;
- $F : \mathbb{R}^n \rightarrow \mathcal{M}_{n,n}(\mathbb{R})$ where $\mathcal{M}_{n,n}(\mathbb{R})$ denotes the set of $n \times n$ real matrices;
- $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$;
- Y_t and Y_x denote the partial derivatives of Y with respect to t and x , respectively.

The system (1) is **hyperbolic** which means that $F(Y)$ has n distinct real eigenvalues (called characteristic velocities) for all Y in a domain $\mathcal{Y}$ of $\mathbb{R}^n$. It is furthermore assumed that these eigenvalues never vanish and therefore do not change sign along the system trajectories in $\mathcal{Y}$.

Here are some typical examples of physical models having the form of a hyperbolic system.

Electrical Lines

First proposed by Heaviside in 1885, the equations of (lossless) electrical lines (also called telegrapher equations) describe the propagation of current and voltage along electrical transmission lines (see Fig. 1). It is a hyperbolic system of the following form:

$$\begin{pmatrix} I_t \\ V_t \end{pmatrix} + \begin{pmatrix} 0 & L_s^{-1} \\ C_s^{-1} & 0 \end{pmatrix} \begin{pmatrix} I_x \\ V_x \end{pmatrix} + \begin{pmatrix} R_s L_s^{-1} I \\ G_s C_s^{-1} V \end{pmatrix} = 0, \quad (2)$$

where $I(t, x)$ is the current intensity, $V(t, x)$ is the voltage, L_s is the line self-inductance per unit of length, C_s is the line self-capacitance per unit of length, R_s is the resistance of the two conductors per unit of length, and G_s is the admittance per unit of length of the dielectric material separating the conductors. The system has two characteristic velocities (which are the two distinct non-zero eigenvalues of the matrix F):

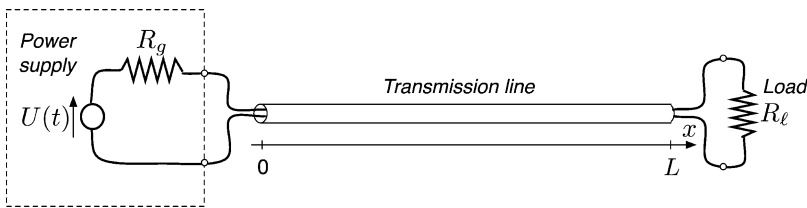
$$\lambda_1 = \frac{1}{\sqrt{L_s C_s}} > 0 > \lambda_2 = -\frac{1}{\sqrt{L_s C_s}}. \quad (3)$$

Saint-Venant Equation for Open Channels

First proposed by Barré-de-Saint-Venant in 1871, the Saint-Venant equations (also called *shallow water equations*) describe the propagation of water in open channels (see Fig. 2). In the case of a horizontal channel with rectangular cross section, unit width, and friction, the Saint-Venant model is a hyperbolic system of the form

$$\begin{pmatrix} H_t \\ V_t \end{pmatrix} + \begin{pmatrix} V & H \\ g & V \end{pmatrix} \begin{pmatrix} H_x \\ V_x \end{pmatrix} + \begin{pmatrix} 0 \\ c_f V^2/H \end{pmatrix} = 0, \quad (4)$$

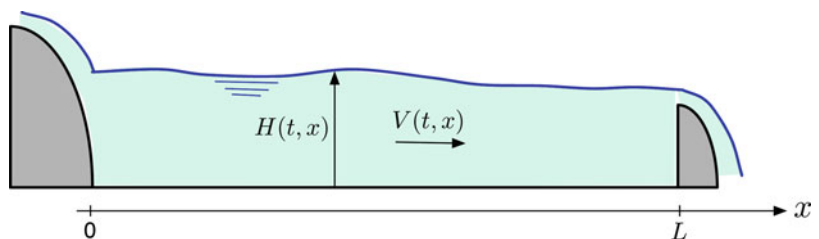
where $H(t, x)$ is the water depth, $V(t, x)$ is the water horizontal velocity, g is the gravity acceleration, and c_f is a friction coefficient. Under sub-



Boundary Control of 1-D Hyperbolic Systems, Fig. 1 Transmission line connecting a power supply to a resistive load R_ℓ ; the power supply is represented by a Thevenin equivalent with $efm U(t)$ and internal resistance R_g

Boundary Control of 1-D Hyperbolic Systems,

Fig. 2 Lateral view of a pool of a horizontal open channel



critical flow conditions, the system is hyperbolic with characteristic velocities

$$\lambda_1 = V + \sqrt{gH} > 0 > \lambda_2 = V - \sqrt{gH}. \quad (5)$$

Aw-Rascle Equations for Fluid Models of Road Traffic

In the fluid paradigm for road traffic modeling, the traffic is described in terms of two basic macroscopic state variables: the density $\varrho(t, x)$ and the speed $V(t, x)$ of the vehicles at position x along the road at time t . The following dynamical model for road traffic was proposed by Aw and Rascle in (2000):

$$\begin{aligned} \begin{pmatrix} \varrho_t \\ V_t \end{pmatrix} + \begin{pmatrix} V & \varrho \\ 0 & V - \varrho P'(\varrho) \end{pmatrix} \begin{pmatrix} \varrho_t \\ V_t \end{pmatrix} \\ + \begin{pmatrix} 0 \\ \sigma(V - V_o(\varrho)) \end{pmatrix} = 0. \end{aligned} \quad (6)$$

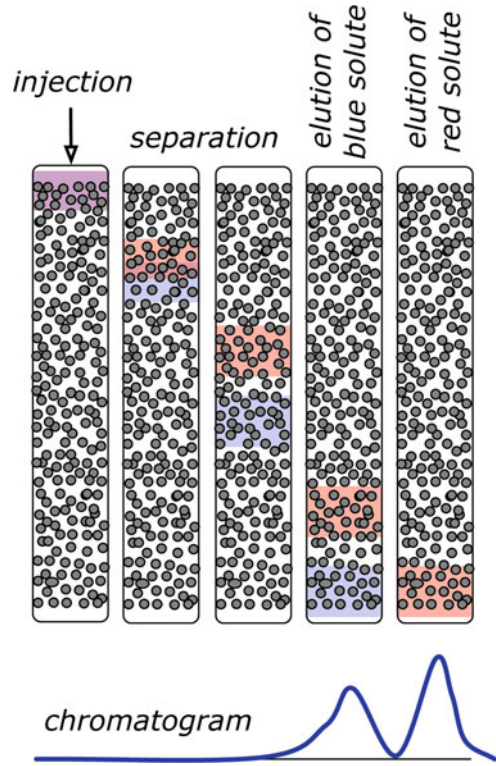
The system is hyperbolic with characteristic velocities

$$\lambda_1 = V > \lambda_2 = V - \varrho P'(\varrho). \quad (7)$$

In this model the first equation is a continuity equation that represents the conservation of the number of vehicles on the road. The second equation is a phenomenological model describing the speed variations induced by the drivers behavior. The function $P(\varrho)$ is a monotonically increasing function called *traffic pressure*. The function $V_o(\rho)$ represents the monotonically decreasing relation between the average speed of the vehicles and the traffic density. The parameter σ is a relaxation constant.

Chromatography

In chromatography, a mixture of species with different affinities is injected in the carrying fluid at the entrance of the process as illustrated in Fig. 3. Various substances travel at different propagation speeds and are ultimately separated in different bands. The dynamics of the mixture are described by a system of partial differential equations:



Boundary Control of 1-D Hyperbolic Systems, Fig. 3
Principle of chromatography

$$(P_i + L_i(P))_t + (V P_i)_x = 0 \quad i = 1, \dots, n,$$

$$L_i(P) = \frac{k_i P_i}{1 + \sum_j k_j P_j / P_{\max}}, \quad (8)$$

where V denotes the speed of the carrying fluid while P_i ($i = 1, \dots, n$) denote the densities of the n carried species. The function $L_i(P)$ (called the “Langmuir isotherm”) was proposed by Langmuir in 1916.

Boundary Control

Boundary control of 1-D hyperbolic systems refers to situations where manipulable control inputs are physically located at the boundaries. Formally, this means that the system (1) is considered under n boundary conditions having the general form

$$\mathcal{B}(Y(t, 0), Y(t, L), U(t)) = 0, \quad (9)$$

with $\mathcal{B} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^q \rightarrow \mathbb{R}^n$. The dependence of the map $\mathcal{B}$ on $(Y(t, 0), Y(t, L))$ refers to natural physical constraints on the system. The function $U(t) \in \mathbb{R}^q$ represents a set of q exogenous control inputs. The following examples illustrate how the control boundary conditions (9) may be defined for some commonly used control devices.

1. **Electrical lines.** For the circuit represented in Fig. 1, the line model (2) is to be considered under the following boundary conditions:

$$V(t, 0) + R_g I(t, 0) = U(t),$$

$$V(t, L) - R_\ell I(t, L) = 0.$$

The telegrapher equations (2) coupled with these boundary conditions constitute therefore a boundary control system with the voltage $U(t)$ as control input.

2. **Open channels.** A standard situation is when the boundary conditions are assigned by tunable hydraulic gates as in irrigation canals and navigable rivers; see Fig. 4. The hydraulic

model of overshoot gates gives the boundary conditions

$$H(t, 0)V(t, 0) = k_G \sqrt{[Z_0(t) - U_0(t)]^3},$$

$$H(t, L)V(t, L) = k_G \sqrt{[H(t, L) - U_L(t)]^3}.$$

In these equations $H(t, 0)$ and $H(t, L)$ denote the water depth at the boundaries inside the pool, $Z_0(t)$ and $Z_L(t)$ are the water levels on the other side of the gates, k_G is a constant gate shape parameter, and U_0 and U_L represent the weir elevations. The Saint-Venant equations coupled to these boundary conditions constitute a boundary control system with $U_0(t)$ and $U_L(t)$ as command signals.

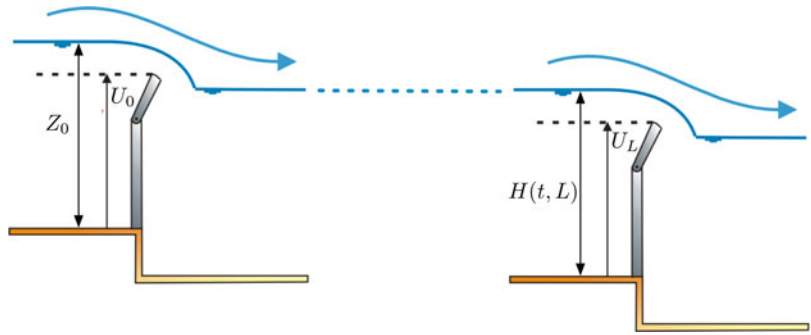
3. **Ramp metering.** Ramp metering is a strategy that uses traffic lights to regulate the flow of traffic entering freeways according to measured traffic conditions as illustrated in Fig. 5. For the stretch of motorway represented in this figure, the boundary conditions are

$$q(t, 0)V(t, 0) = Q_{in}(t) + U(t),$$

$$q(t, L)V(t, L) = Q_{out}(t),$$

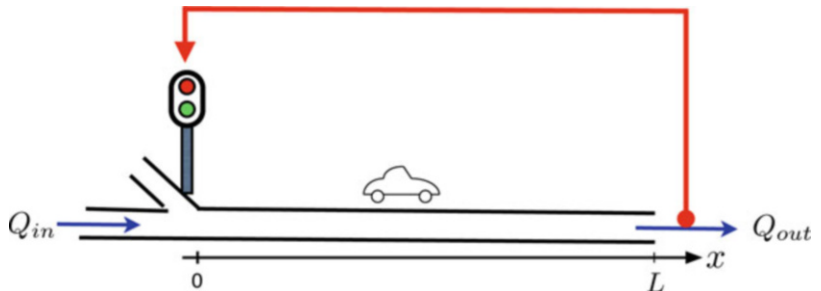
Boundary Control of 1-D Hyperbolic Systems,

Fig. 4 Hydraulic overshoot gates at the input and the output of a pool



Boundary Control of 1-D Hyperbolic Systems,

Fig. 5 Ramp metering on a stretch of a motorway



where $U(t)$ is the inflow rate controlled by the traffic lights. The Aw-Rascle equations (6) coupled to these boundary conditions constitute a boundary control system with $U(t)$ as the command signal. In a feedback implementation of the ramp metering strategy, $U(t)$ may be a function of the measured disturbances $Q_{\text{int}}(t)$ or $Q_{\text{out}}(t)$ that are imposed by the traffic conditions.

4. **Simulated moving bed chromatography** is a technology where several interconnected chromatographic columns are switched periodically against the fluid flow. This allows for a continuous separation with a better performance than the discontinuous single-column chromatography. An efficient operation of SMB chromatography requires a tight control of the process by manipulating the inflow rates in the columns. This process

is therefore a typical example of a periodic boundary control hyperbolic system.

Steady State and Change of Coordinates

A steady state (or equilibrium) of the system (1)–(9) is a time-invariant solution $Y(t, x) = Y^*(x)$, $U(t) = U^*$, $\forall t \in [0, +\infty)$. It satisfies the ordinary differential equation

$$F(Y^*)Y_x^* + G(Y^*) = 0, \quad x \in [0, L], \quad (10)$$

together with the boundary condition

$$\mathcal{B}(Y^*(0), Y^*(L), U^*) = 0. \quad (11)$$

Having distinct eigenvalues by assumption, the matrix $F(Y^*(x))$ is diagonalizable:

$$\begin{aligned} \exists N(x) \in \mathcal{M}_{n,n}(\mathbb{R}) \quad \text{such that} \quad N(x)F(Y^*(x)) &= \Lambda(x)N(x), \\ \text{with} \quad \Lambda(x) &= \text{diag}\{\lambda_1(x), \dots, \lambda_n(x)\}, \end{aligned}$$

where $\lambda_i(x)$ is the i -th eigenvalue of the matrix $F(Y^*(x))$. Furthermore, it is assumed that for all x , there are m positive and $n - m$ negative eigenvalues as follows:

$$\begin{aligned} \lambda_i(x) &> 0 \quad \forall i \in \{1, 2, \dots, m\}, \\ \lambda_i(x) &< 0 \quad \forall i \in \{m+1, \dots, n\}, \end{aligned} \quad (12)$$

such that the matrix $\Lambda(x)$ is written

$$\Lambda(x) = \begin{pmatrix} \Lambda^+(x) & 0 \\ 0 & \Lambda^-(x) \end{pmatrix} \quad (13)$$

with $\Lambda^+ = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_m\}$ and $\Lambda^- = \text{diag}\{\lambda_{m+1}, \lambda_{m+2}, \dots, \lambda_n\}$.

Then, with the change of coordinates

$$Z(t, x) = N(x)(Y(t, x) - Y^*(x)), \quad (14)$$

the control system (1)–(9) is equivalent to the system

$$Z_t + A(Z, x)Z_x + B(Z, x)Z = 0, \quad (15)$$

$$\begin{aligned} \mathcal{B}(N(0)^{-1}Z(t, 0) + Y^*(0), \\ N(L)^{-1}Z(t, L) + Y^*(L), U(t)) &= 0, \end{aligned} \quad (16)$$

where

$$A(Z, x) = N(x)F(N(x)^{-1}Z + Y^*(x))N(x)^{-1} \text{ with } A(0, x) = \Lambda(x), \quad (17)$$

$$\begin{aligned} B(Z, x)Z &= N(x) \left[F(N^{-1}(x)Z + Y^*(x))(Y_x^*(x) \right. \\ &\quad \left. - N(x)^{-1}N'(x)N(x)^{-1}Z) + G(N(x)^{-1}Z + Y^*(x)) \right]. \end{aligned} \quad (18)$$

Controllability

For the boundary control system (1)–(9), the local controllability issue is to investigate if, starting from a given initial state $Y_0 : x \in [0, L] \mapsto Y_0(x) \in \mathbb{R}^n$, it is possible to reach in time T a desired target state $Y_1 : x \in [0, L] \mapsto Y_1(x) \in \mathbb{R}^n$, with $Y_0(x)$ and $Y_1(x)$ close to $Y^*(x)$. Using the state transformation of the previous section, this controllability problem for the physical system (1)–(9) is equivalent to, starting from a given initial state $Z_0 : x \in [0, L] \mapsto Z_0(x) \in \mathbb{R}^n$, to reach in time T a desired target state $Z_1 : x \in$

$[0, L] \mapsto Z_1(x) \in \mathbb{R}^n$, with $Z_0(x)$ and $Z_1(x)$ close to 0, for the auxiliary system (15)–(16).

Let $Z^+ \in \mathbb{R}^m$ and $Z^- \in \mathbb{R}^{n-m}$ be such that $Z^T = (Z^{+T} \ Z^{-T})^T$.

Theorem 1 *If there exist control inputs $U^+(t)$ and $U^-(t)$ such that the boundary conditions (16) are equivalent to*

$$Z^+(t, 0) = U^+(t), \quad Z^-(t, L) = U^-(t), \quad (19)$$

then the boundary control system (15) and (16) is locally controllable for the C^1 -norm if $T > T_c$ with

$$T_c = \max \left\{ \frac{L}{\min \{|\lambda_1(x)|; x \in [0, L]\}}, \dots, \frac{L}{\min \{|\lambda_n(x)|; x \in [0, L]\}} \right\}.$$

The proof of this theorem can be found in Wang (2006). For the special case where Y^* is constant, see also Li and Rao (2003).

Feedback Stabilization

For the boundary control system (1)–(9), the problem of local boundary feedback stabilization is the problem of finding a boundary feedback $U(Y(t, 0), Y(t, L))$ such that the system trajectory exponentially converges to a desired steady-state $Y^*(x)$ from any initial condition $Y_0(x)$ close to $Y^*(x)$. In such case, the steady state is said to be exponentially stabilizable.

From the system transformation of section “Steady State and Change of Coordinates”, it is clear that looking for a boundary stabilizing feedback $U(Y(t, 0), Y(t, L))$ for the physical

system (1)–(9) is equivalent to looking for a feedback $U(Z(t, 0), Z(t, L))$ that exponentially stabilizes the zero steady state of the auxiliary system (15)–(16).

Theorem 2 *Assume that there exists a boundary feedback $U(Z(t, 0), Z(t, L))$ such that the boundary condition (16) is written in the form*

$$\begin{pmatrix} Z^+(t, 0) \\ Z^-(t, L) \end{pmatrix} = \mathcal{H} \begin{pmatrix} Z^+(t, L) \\ Z^-(t, 0) \end{pmatrix}, \quad \mathcal{H}(0) = 0, \quad (20)$$

where $\mathcal{H} : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

Then, for the boundary control system (15)–(16)–(20), the zero steady state is locally exponentially stable for the H^2 -norm if there exists a map $Q = \text{diag}\{Q^+, Q^-\}$ with $Q^+ : [0, L] \rightarrow \mathcal{D}^m$ and $Q^- : [0, L] \rightarrow \mathcal{D}^{n-m}$ such that the following matrix inequalities hold:

$$\begin{pmatrix} Q^+(L)\Lambda^+(L) & 0 \\ 0 & -Q^-(0)\Lambda^-(0) \end{pmatrix} - K^T \begin{pmatrix} Q^+(0)\Lambda^+(0) & 0 \\ 0 & -Q^-(L)\Lambda^-(L) \end{pmatrix} K > 0, \quad (21)$$

$$-(Q(x)\Lambda(x))_x + Q(x)B(0, x) + B^T(0, x)Q(x) > 0 \quad \forall x \in [0, L], \quad (22)$$

where $K = \mathcal{H}'(0)$ is the Jacobian matrix of the map $\mathcal{H}$ at 0 and $\mathcal{D}^k$ denotes the set of diagonal $k \times k$ matrices with positive diagonal entries.

The proof of this theorem can be found in Bastin and Coron (2016, Chapter 6). For the stabilization in the C^1 -norm, alternative sufficient conditions are given in Hayat (2017).

When the physical system is linear or when the steady state is uniform (i.e., Y^* is constant and independent of x), we have the interesting and important special case where the linearization of the control system (15)–(16)–(20) is written

$$Z_t + \Lambda Z_x + BZ = 0 \quad (23)$$

$$\begin{pmatrix} Z^+(t, 0) \\ Z^-(t, L) \end{pmatrix} = K \begin{pmatrix} Z^+(t, L) \\ Z^-(t, 0) \end{pmatrix}, \quad (24)$$

with Λ , B , and K constant matrices. In this special case, simpler and much more explicit stability conditions can be formulated as the one given in the next theorem.

Theorem 3 *If the feedback control $U(Z(t, 0), Z(t, L))$ can be selected such that the matrix $K = \mathcal{H}'(0)$ satisfies the inequality*

$$\inf \{ \|\Delta K \Delta^{-1}\|; \Delta \in \mathcal{D}^n \} < 1, \quad (25)$$

then there exists $\varepsilon > 0$, function of K , such that, if $\|B\| < \varepsilon$, the linear system (23) and (24) is exponentially stable for the L^2 -norm and the zero steady state of the nonlinear system (15)–(20) is locally exponentially stable for the H^2 -norm.

The proof of this theorem can be found in Bastin and Coron (2016, Chapters 3 and 6).

Future Directions

For further information on the controllability problem, in particular the case where there are controls on only one side of $[0, L]$, see Li (2010), Hu (2015), Coron and Nguyen (2019) and Hu and Olive (2019).

Concerning the stabilization problem, as shown in Bastin and Coron (2016, Section 5.6), there are cases where there are no feedback laws of the form (20) stabilizing Y^* . One needs to use nonlocal feedback laws. To construct such feedback laws, a possibility is to use the Krstic's backstepping approach Krstic and Smyshlyaev (2008): it was first done in Coron et al. (2013) for the case $n = 2$ and $m = 1$ and (Di Meglio et al. 2013; Auriol and Meglio 2016; Hu et al. 2016,

2019; Coron et al. 2017; Coron and Nguyen 2019) for more general cases.

In this note, we have only addressed the stabilization by proportional feedback of local measurable boundary states. In many practical situations, it is however well-known that there are load disturbances which cannot be measured and therefore not directly compensated in the control. In this case it is useful to implement integral actions in order to eliminate offsets and to attenuate the incidence of load disturbances. The stability of hyperbolic systems with proportional-integral (PI) control has been investigated, for instance, in Bastin and Coron (2016, Chapters 2, 5, and 8) and (Dos Santos et al. 2008; Bastin et al. 2015; Lamare and Bekiaris-Liberis 2015; Trinh et al. 2017; Coron and Hayat 2018; Hayat 2019). Another important issue for the system (1) is the observability problem: assume that the state is measured on the boundary during the interval of time $[0, T]$, can one recover the initial data? As shown in Li (2010), this problem has strong connections with the controllability problem, and the system (1) is observable if the time T is large enough.

The above results are on smooth solutions of (1). However, the system (1) is known to be well posed in class of BV -solutions (bounded variations), with extra conditions (e.g., entropy type); see in particular (Bressan 2000). There are partial results on the controllability in this class. See, in particular, (Ancona and Marson 1998; Horsin 1998) for $n = 1$. For $n = 2$, it is shown in Bressan and Coclite (2002) that Theorem 1 no longer holds in general the BV class. However there are positive results for important physical systems; see, for example, Glass (2007) for the 1-D isentropic Euler equation. Concerning the stabilization problem in the BV -class, see Coron et al. (2017).

Cross-References

- [Control of Fluid Flows and Fluid-Structure Models](#)
- [Control of Linear Systems with Delays](#)

- **Controllability and Observability**
- **Feedback Stabilization of Nonlinear Systems**
- **Lyapunov's Stability Theory**

Bibliography

- Ancona F, Marson A (1998) On the attainable set for scalar nonlinear conservation laws with boundary control. *SIAM J Control Optim* 36(1):290–312 (electronic)
- Auriol J, Di Meglio F (2016) Minimum time control of heterodirectional linear coupled hyperbolic PDEs. *Automatica J IFAC* 71:300–307
- Aw A, Rascle M (2000) Resurrection of “second order” models of traffic flow. *SIAM J Appl Math* 60(3): 916–938 (electronic)
- Bastin G, Coron J-M (2016) Stability and boundary stabilisation of 1-d hyperbolic systems. Number 88 in progress in nonlinear differential equations and their applications. Springer International
- Bastin G, Coron J-M, Tamasoiu SO (2015) Stability of linear density-flow hyperbolic systems under PI boundary control. *Automatica J IFAC* 53: 37–42
- Bressan A (2000) Hyperbolic systems of conservation laws, volume 20 of *Oxford lecture series in mathematics and its applications*. Oxford University Press, Oxford. The one-dimensional Cauchy problem
- Bressan A, Coclite GM (2002) On the boundary control of systems of conservation laws. *SIAM J Control Optim* 41(2):607–622 (electronic)
- Coron J-M, Hayat A (2018) PI controllers for 1-D nonlinear transport equation. *IEEE Trans Automat Control* 64(11):4570–4582
- Coron J-M, Nguyen H-M (2019) Optimal time for the controllability of linear hyperbolic systems in one-dimensional space. *SIAM J Control Optim* 57(2):1127–1156
- Coron J-M, Vazquez R, Krstic M, Bastin G (2013) Local exponential H^2 stabilization of a 2×2 quasilinear hyperbolic system using backstepping. *SIAM J Control Optim* 51(3):2005–2035
- Coron J-M, Ervedoza S, Ghoshal SS, Glass O, Perrollaz V (2017) Dissipative boundary conditions for 2×2 hyperbolic systems of conservation laws for entropy solutions in BV. *J Differ Equ* 262(1): 1–30
- Coron J-M, Hu L, Olive G (2017) Finite-time boundary stabilization of general linear hyperbolic balance laws via Fredholm backstepping transformation. *Automatica J IFAC* 84:95–100
- Di Meglio F, Vazquez R, Krstic M (2013) Stabilization of a system of $n + 1$ coupled first-order hyperbolic linear PDEs with a single boundary input. *IEEE Trans Automat Control* 58(12): 3097–3111
- Dos Santos V, Bastin G, Coron JM, d’Andréa Novel B (2008) Boundary control with integral action for hyperbolic systems of conservation laws: stability and experiments. *Automatica J IFAC* 44(5): 1310–1318
- Glass O (2007) On the controllability of the 1-D isentropic Euler equation. *J Eur Math Soc (JEMS)* 9(3): 427–486
- Hayat A (2017) Exponential stability of general 1-D quasilinear systems with source terms for the C^1 norm under boundary conditions. Accepted for publication in *Siam J Control*
- Hayat A (2019) PI controller for the general Saint-Venant equations. Preprint, hal-01827988
- Horsin T (1998) On the controllability of the Burgers equation. *ESAIM Control Optim Calc Var* 3:83–95 (electronic)
- Hu L (2015) Sharp time estimates for exact boundary controllability of quasilinear hyperbolic systems. *SIAM J Control Optim* 53(6):3383–3410
- Hu L, Olive G (2019) Minimal time for the exact controllability of one-dimensional first-order linear hyperbolic systems by one-sided boundary controls. Preprint, hal-01982662
- Hu L, Di Meglio F, Vazquez R, Krstic M (2016) Control of homodirectional and general heterodirectional linear coupled hyperbolic PDEs. *IEEE Trans Automat Control* 61(11):3301–3314
- Hu L, Vazquez R, Di Meglio F, Krstic M (2019) Boundary exponential stabilization of 1-dimensional inhomogeneous quasi-linear hyperbolic systems. *SIAM J Control Optim* 57(2):963–998
- Krstic M, Smyshlyaev A (2008) Boundary control of PDEs, volume 16 of *advances in design and control*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia. A course on backstepping designs.
- Lamare P-O, Bekiaris-Liberis N (2015) Control of 2×2 linear hyperbolic systems: backstepping-based trajectory generation and PI-based tracking. *Syst Control Lett* 86:24–33
- Li T (2010) Controllability and observability for quasilinear hyperbolic systems, volume 3 of *AIMS series on applied mathematics*. American Institute of Mathematical Sciences (AIMS), Springfield
- Li T, Rao B-P (2003) Exact boundary controllability for quasi-linear hyperbolic systems. *SIAM J Control Optim* 41(6):1748–1755 (electronic)
- Trinh N-T, Andrieu V, Xu C-Z (2017) Design of integral controllers for nonlinear systems governed by scalar hyperbolic partial differential equations. *IEEE Trans Automat Control* 62(9): 4527–4536
- Wang Z (2006) Exact controllability for nonautonomous first order quasilinear hyperbolic systems. *Chinese Ann Math Ser B* 27(6):643–656

Boundary Control of Korteweg-de Vries and Kuramoto-Sivashinsky PDEs

Eduardo Cerpa

Departamento de Matemática, Universidad
Técnica Federico Santa María, Valparaíso, Chile

Abstract

The Korteweg-de Vries (KdV) and the Kuramoto-Sivashinsky (KS) partial differential equations are used to model nonlinear propagation of one-dimensional phenomena. The KdV equation is used in fluid mechanics to describe wave propagation in shallow water surfaces, while the KS equation models front propagation in reaction-diffusion systems. In this article, the boundary control of these equations is considered when they are posed on a bounded interval. Different choices of controls are studied for each equation.

Keywords

Controllability · Stabilizability ·
Higher-order partial differential equations ·
Dispersive equations · Parabolic equations

Introduction

The Korteweg-de Vries (KdV) and the Kuramoto-Sivashinsky (KS) equations have very different properties because they do not belong to the same class of partial differential equations (PDEs). The first one is a third-order nonlinear dispersive equation

$$y_t + y_x + y_{xxx} + yy_x = 0, \quad (1)$$

and the second one is a fourth-order nonlinear parabolic equation

$$u_t + u_{xxxx} + \lambda u_{xx} + uu_x = 0, \quad (2)$$

where $\lambda > 0$ is called the anti-diffusion parameter. However, they have one important characteristic in common. They are both used to model nonlinear propagation phenomena in the space x -direction when the variable t stands for time. The KdV equation serves as a model for wave propagation in shallow water surfaces (Korteweg and de Vries 1895), and the KS equation models front propagation in reaction-diffusion phenomena including some instability effects (Kuramoto and Tsuzuki 1975; Sivashinsky 1977).

From a control point of view, a new common characteristic arises. Because of the order of the spatial derivatives involved, when studying these equations on a bounded interval $[0, L]$, two boundary conditions have to be imposed at the same point, for instance, at $x = L$. Thus, we can consider control systems where we control one boundary condition but not all the boundary data at one endpoint of the interval. This configuration is not possible for the classical wave and heat equations where at each extreme, only one boundary condition exists and therefore controlling one or all the boundary data at one point is the same.

Being the KdV equation of third order in space, three boundary conditions have to be imposed: one at the left endpoint $x = 0$ and two at the right endpoint $x = L$. For the KS equation, four boundary conditions are needed to get a well-posed system, two at each extreme. We will focus on the cases where Dirichlet and Neumann boundary conditions are considered because lack of controllability phenomena appears. This holds for some special values of the length of the interval for the KdV equation and depends on the anti-diffusion coefficient λ for the KS equation.

The particular cases where the lack of controllability occurs can be seen as isolated anomalies. However, those phenomena give us important information on the systems. In particular, any method insensible to the value of those constants cannot control or stabilize the system when acting from the corresponding control input where trouble appears. In all of these cases, for both the KdV and the KS equations, the space of uncontrollable states is finite dimensional, and therefore, some

methods coming from the control of ordinary differential equations can be applied.

General Definitions

Infinite-dimensional control systems described by PDEs have attracted a lot of attention since the 1970s. In this framework, the state of the control system is given by the solution of an evolution PDE. This solution can be seen as a trajectory in an infinite-dimensional Hilbert space H , for instance, the space of square-integrable functions or some Sobolev space. Thus, for any time t , the state belongs to H . Concerning the control input, this is either an internal force distributed in the domain or a punctual force localized within the domain or some boundary data as considered in this article. For any time t , the control belongs to a control space U , which can be, for instance, the space of bounded functions. The main control properties to be mentioned in this article are controllability, stability, and stabilization. A control system is said to be exactly controllable if the system can be driven from any initial state to another one in finite time. This kind of properties holds, for instance, for hyperbolic system as the wave equation. The notion of null controllability means that the system can be driven to the origin from any initial state. The main example for this property is the heat equation, which presents regularizing effects. Even if the initial data is discontinuous, right after $t = 0$, the solution of the heat equation becomes very smooth, and therefore, it is not possible to impose a discontinuous final state. A system is said to be asymptotically stable if the solutions of the system without any control converge as the time goes to infinity to a stationary solution of the PDE. When this convergence holds with a control depending at each time on the state of the system (feedback control), the system is said to be stabilizable by means of a feedback control law.

All these properties have local versions when a smallness condition for the initial and/or the final state is added. This local character is normally due to the nonlinearity of the system.

The KdV Equation

The classical approach to deal with nonlinearities is first to linearize the system around a given state or trajectory, then to study the linear system, and finally to go back to the nonlinear one by means of an inversion argument or a fixed-point theorem. Linearizing (1) around the origin, we get the equation

$$y_t + y_x + y_{xxx} = 0, \quad (3)$$

which can be studied on a finite interval $[0, L]$ under the following three boundary conditions:

$$\begin{aligned} y(t, 0) = h_1(t), \quad y(t, L) = h_2(t), \\ \text{and} \quad y_x(t, L) = h_3(t). \end{aligned} \quad (4)$$

Thus, viewing $h_1(t)$, $h_2(t)$, $h_3(t) \in \mathbb{R}$ as controls and the solution $y(t, \cdot) : [0, L] \rightarrow \mathbb{R}$ as the state, we can consider the linear control system (3)–(4) and the nonlinear one (1), (2), (3), and (4).

We will report on the role of each input control when the other two are off. The tools used are mainly the duality controllability-observability, Carleman estimates, the multiplier method, the compactness-uniqueness argument, the backstepping method, and fixed-point theorems. Surprisingly, the control properties of the system depend strongly on the location of the controls.

Theorem 1 *The linear KdV system (3)–(4) is:*

1. *Null-controllable when controlled from h_1 (i.e., $h_2 = h_3 = 0$) (Glass and Guerrero 2008; Xiang 2019).*
2. *Exactly controllable when controlled from h_2 (i.e., $h_1 = h_3 = 0$) if and only if L does not belong to a set O of critical lengths defined in Glass and Guerrero (2010).*
3. *Exactly controllable when controlled from h_3 (i.e., $h_1 = h_2 = 0$) if and only if L does not belong to a set of critical lengths N defined in Rosier (1997).*
4. *Asymptotically stable to the origin if $L \notin N$ and no control is applied (Perla 2002).*

5. *Stabilizable by means of a feedback law using either h_1 only (i.e., $h_2 = h_3 = 0$) (Cerpa and Coron 2013; Xiang 2018) or h_2 and h_3 (i.e., $h_1 = 0$) (Özsari and Batal 2019). If $L \notin N$, then the stabilization is obtained using h_3 only (Coron and Lü 2014).*

If $L \in N$ or $L \in O$, one says that L is a critical length since the linear control system (3)–(4) loses controllability properties when only one control input is applied. In those cases, there exists a finite-dimensional subspace of $L^2(0, L)$ which is unreachable from 0 for the linear system. The sets N and O contain infinitely many critical lengths, but they are countable sets.

When one is allowed to use more than one boundary control input, there is no critical spatial domain, and the exact controllability holds for any $L > 0$. This is proved in Zhang (1999) when three boundary controls are used. The case of two control inputs is solved in Rosier (1997), Glass and Guerrero (2010), and Cerpa et al. (2013).

Previous results concern the linearized control system. Considering the nonlinearity yy_x , we obtain the original KdV control system and the following results.

Theorem 2 *The nonlinear KdV system (1) (2), (3), and (4) is:*

1. *Locally null-controllable when controlled from h_1 (i.e., $h_2 = h_3 = 0$) (Glass and Guerrero 2008).*
2. *Locally exactly controllable when controlled from h_2 (i.e., $h_1 = h_3 = 0$) if L does not belong to the set O of critical lengths (Glass and Guerrero 2010).*
3. *Locally exactly controllable when controlled from h_3 (i.e., $h_1 = h_2 = 0$). If L belongs to the set of critical lengths N , then a minimal time of control may be required (Cerpa 2014).*
4. *Asymptotically stable to the origin if $L \notin N$ and no control is applied (Perla 2002). For some lengths $L \in N$, the asymptotical stability has been proven in Chu et al. (2015) and Tang et al. (2018).*
5. *Locally stabilizable by means of a feedback law using either h_1 only (i.e., $h_2 = h_3 = 0$) (Cerpa and Coron 2013; Xiang 2018) or*

h_2 and h_3 (i.e., $h_1 = 0$) (Özsari and Batal 2019). If $L \notin N$, then the local stabilization is obtained using h_3 only (Coron et al. 2017; Coron and Lü 2014).

Item 2 in Theorem 2 is a truly nonlinear result obtained by applying a power series method introduced in Coron and Crépeau (2004). Stability results in Chu et al. (2015) and Tang et al. (2018) also use nonlinear methods. All other items are implied by perturbation arguments based on the linear control system. The related control system formed by (1) with boundary controls

$$\begin{aligned} y(t, 0) &= h_1(t), & y_x(t, L) &= h_2(t), \\ \text{and } y_{xx}(t, L) &= h_3(t) \end{aligned} \quad (5)$$

is studied in Cerpa et al. (2013) and Guilleron (2014). The same phenomenon of critical lengths appears.

The KS Equation

Applying the same strategy than for KdV, we linearize (2) around the origin to get the equation

$$u_t + u_{xxx} + \lambda u_{xx} = 0, \quad (6)$$

which can be studied on the finite interval $[0, 1]$ under the following four boundary conditions:

$$\begin{aligned} u(t, 0) &= v_1(t), & u_x(t, 0) &= v_2(t), \\ u(t, 1) &= v_3(t), & \text{and } u_x(t, 1) &= v_4(t). \end{aligned} \quad (7)$$

Thus, viewing $v_1(t), v_2(t), v_3(t), v_4(t) \in \mathbb{R}$ as controls and the solution $u(t, \cdot) : [0, 1] \rightarrow \mathbb{R}$ as the state, we can consider the linear control system (6)–(7) and the nonlinear one (2)–(7). The role of the parameter λ is crucial. The KS equation is parabolic and the eigenvalues of system (6)–(7) with no control ($v_1 = v_2 = v_3 = v_4 = 0$) go to $-\infty$. If λ increases, then the eigenvalues move to the right. When $\lambda > 4\pi^2$, the system becomes unstable because there are a finite

number of positive eigenvalues. In this unstable regime, the system loses control properties for some values of λ .

Theorem 3 *The linear KS control system (6)–(7) is:*

1. *Null-controllable when controlled from v_1 and v_2 (i.e., $v_3 = v_4 = 0$). The same is true when controlling v_3 and v_4 (i.e., $v_1 = v_2 = 0$) (Lin 2002; Cerpa and Mercado 2011).*
2. *Null-controllable when controlled from v_2 (i.e., $v_1 = v_2 = v_3 = 0$) if and only if λ does not belong to a countable set M defined in Cerpa (2010).*
3. *Null-controllable when controlled from v_1 (i.e., $v_2 = v_3 = v_4 = 0$) if and only if λ does not belong to a countable set M' defined in Cerpa et al. (2017).*
4. *Asymptotically stable to the origin if $\lambda < 4\pi^2$ and no control is applied (Liu and Krstic 2001).*
5. *Stabilizable by means of a feedback law using v_2 only (i.e., $v_2 = v_3 = v_4 = 0$) if and only if $\lambda \notin M$ (Cerpa 2010).*

In the critical case $\lambda \in M$, the linear system is not null-controllable anymore if we control v_2 only (item 3 in Theorem 3). The space of noncontrollable states is finite dimensional. To obtain the null controllability of the linear system in these cases, we have to add another control. Controlling with v_2 and v_4 does not improve the situation in the critical cases. Unlike that, the system becomes null-controllable if we can act on v_1 and v_2 . This result with two input controls has been proved in Lin (2002) for the case $\lambda = 0$ and in Cerpa and Mercado (2011) in the general case (item 3 in Theorem 3).

It is known from Liu and Krstic (2001) that if $\lambda < 4\pi^2$, then the system is exponentially stable in $L^2(0, 1)$. On the other hand, if $\lambda = 4\pi^2$, then zero becomes an eigenvalue of the system, and therefore the asymptotic stability fails. When $\lambda > 4\pi^2$, the system has positive eigenvalues and becomes unstable. In order to stabilize this system, a finite-dimensional-based feedback law can be designed by using the pole placement method (item 3 in Theorem 3).

Previous results concern the linearized control system. If we add the nonlinearity uu_x , we obtain the original KS control system and the following results.

Theorem 4 *The KS control system (2)–(7) is:*

1. *Locally null-controllable when controlled from v_1 and v_2 (i.e., $v_3 = v_4 = 0$). The same is true when controlling v_3 and v_4 (i.e., $v_1 = v_2 = 0$) (Cerpa and Mercado 2011).*
2. *Locally null-controllable when controlled from v_2 (i.e., $v_1 = v_2 = v_3 = 0$) if λ does not belong to a countable set M defined in Cerpa (2010). This result is due to Takahashi (2017).*
3. *Asymptotically stable to the origin if $\lambda < 4\pi^2$ and no control is applied (Liu and Krstic 2001).*

There are less results for the nonlinear systems than for the linear one. This is due to the fact that the spectral techniques used to study the linear system with only one control input are not always robust enough to deal with perturbations in order to address the nonlinear control system. One important exception is the result in item 4. When considering the KS equation with boundary conditions

$$\begin{aligned} u(t, 0) &= v_1(t), & u_{xx}(t, 0) &= v_2(t), \\ u(t, 1) &= v_3(t), & \text{and } u_{xx}(t, 1) &= v_4(t), \end{aligned} \quad (8)$$

we find stabilization results in Coron and Lü (2015) and cost-of-control results in Carreño (2016).

Summary and Future Directions

The KdV and the KS equations possess both noncontrol results when one boundary control input is applied. This is due to the fact that both are higher-order equations, and therefore, when posed on a bounded interval, more than one boundary condition should be imposed at

the same point. The KdV equation is exact controllable when acting from the right and null-controllable when acting from the left. On the other hand, the KS equation, being parabolic as the heat equation, is not exact controllable but null-controllable. Most of the results are implied by the behaviors of the corresponding linear system, which are very well understood.

For the KdV equation, the main directions to investigate at this moment are the controllability and the stability for the nonlinear equation in critical domains. Among others, some questions concerning controllability, minimal time of control, and decay rates for the stability are open. Regarding the KS equation, there are few results for the nonlinear system with one control input even if we are not in a critical value of the anti-diffusion parameter. In the critical cases, the controllability and stability issues are wide open.

In general, for PDEs, there are few results about delay phenomena, output feedback laws, adaptive control, and other classical questions in control theory. The existing results on these topics mainly concern the more popular heat and wave equations. As KdV and KS equations are one dimensional in space, many mathematical tools are available to tackle those problems. For all that, to our opinion, the KdV and KS equations are excellent candidates to continue investigating these control properties in a PDE framework, as shown in recent articles (Marx and Cerpa 2018; Baudouin et al. 2019; Guzmán et al. 2019).

Cross-References

- [Boundary Control of 1-D Hyperbolic Systems](#)
- [Control of Fluid Flows and Fluid-Structure Models](#)
- [Controllability and Observability](#)
- [Feedback stabilization of nonlinear systems](#)
- [Stability: Lyapunov, Linear Systems](#)

Recommended Reading

The book Coron (2007) is a very good reference to study the control of PDEs. In Cerpa (2014),

a tutorial presentation of the KdV control system is given. Control systems for PDEs with boundary conditions and internal controls are considered in Rosier and Zhang (2009) and the references therein for the KdV equation and in Armaou and Christofides (2000) and Christofides and Armaou (2000) for the KS equation. Control topics as delay and adaptive control are studied in the framework of PDEs in Krstic (2009) and Smyshlyaev and Krstic (2010), respectively.

Bibliography

- Armaou A, Christofides PD (2000) Feedback control of the Kuramoto-Sivashinsky equation. *Physica D* 137:49–61
- Baudouin L, Crépeau E, Valein J (2019) Two approaches for the stabilization of nonlinear KdV equation with boundary time-delay feedback. *IEEE Trans Autom Control* 64:1403–1414
- Carreño N, Guzmán P (2016) On the cost of null controllability of a fourth-order parabolic equation. *J Differ Equ* 261:6485–6520
- Cerpa E (2010) Null controllability and stabilization of a linear Kuramoto-Sivashinsky equation. *Commun Pure Appl Anal* 9:91–102
- Cerpa E (2014) Control of a Korteweg-de Vries equation: a tutorial. *Math Control Rel Fields* 4:45–99
- Cerpa E, Coron J-M (2013) Rapid stabilization for a Korteweg-de Vries equation from the left Dirichlet boundary condition. *IEEE Trans Autom Control* 58:1688–1695
- Cerpa E, Mercado A (2011) Local exact controllability to the trajectories of the 1-D Kuramoto-Sivashinsky equation. *J. Differ Equ* 250:2024–2044
- Cerpa E, Rivas I, Zhang B-Y (2013) Boundary controllability of the Korteweg-de Vries equation on a bounded domain. *SIAM J Control Optim* 51:2976–3010
- Cerpa E, Guzmán P, Mercado A (2017) On the control of the linear Kuramoto-Sivashinsky equation. *ESAIM Control Optim Calc Var* 23:165–194
- Christofides PD, Armaou A (2000) Global stabilization of the Kuramoto-Sivashinsky equation via distributed output feedback control. *Syst Control Lett* 39: 283–294
- Chu J, Coron J-M, Shang P (2015) Asymptotic stability of a nonlinear Korteweg-de Vries equation with critical lengths. *J Differ Equ* 259:4045–4085
- Coron J-M (2007) Control and nonlinearity. American Mathematical Society, Providence
- Coron J-M, Crépeau E (2004) Exact boundary controllability of a nonlinear KdV equation with critical lengths. *J Eur Math Soc* 6:367–398

- Coron J-M, Lü Q (2014) Local rapid stabilization for a Korteweg-de Vries equation with a Neumann boundary control on the right. *J Math Pures Appl* 102:1080–1120
- Coron J-M, Lü Q (2015) Fredholm transform and local rapid stabilization for a Kuramoto-Sivashinsky equation. *J Differ Equ* 259:3683–3729
- Coron J-M, Rivas I, Xiang S (2017) Local exponential stabilization for a class of Korteweg-de Vries equations by means of time-varying feedback laws. *Anal PDE* 10:1089–1122
- Guilleron J-P (2014) Null controllability of a linear KdV equation on an interval with special boundary conditions. *Math Control Signals Syst* 26:375–401
- Guzmán P, Marx S, Cerpa E (2019) Stabilization of the linear Kuramoto-Sivashinsky equation with a delayed boundary control. IFAC workshop on control of systems governed by partial differential equations, Oaxaca
- Korteweg DJ, de Vries G (1895) On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves. *Philos Mag* 39: 422–443
- Krstic M (2009) Delay compensation for nonlinear, Adaptive, and PDE systems. Birkhauser, Boston
- Kuramoto Y, Tsuzuki T (1975) On the formation of dissipative structures in reaction-diffusion systems. *Theor Phys* 54:687–699
- Glass O, Guerrero S (2008) Some exact controllability results for the linear KdV equation and uniform controllability in the zero-dispersion limit. *Asymptot Anal* 60:61–100
- Glass O, Guerrero S (2010) Controllability of the KdV equation from the right Dirichlet boundary condition. *Syst Control Lett* 59:390–395
- Lin Guo Y-J (2002) Null boundary controllability for a fourth order parabolic equation. *Taiwan J Math* 6: 421–431
- Liu W-J, Krstic M (2001) Stability enhancement by boundary control in the Kuramoto-Sivashinsky equation. *Nonlinear Anal Ser A Theory Methods* 43: 485–507
- Marx S, Cerpa E (2018) Output feedback stabilization of the Korteweg-de Vries equation. *Autom J IFAC* 87:210–217
- Özsari T, Batal A (2019) Pseudo-backstepping and its application to the control of Korteweg-de Vries equation from the right endpoint on a finite domain. *SIAM J Control Optim* 57:1255–1283
- Perla Menzala G, Vasconcellos CF, Zuazua E (2002) Stabilization of the Korteweg-de Vries equation with localized damping. *Q Appl Math LX*: 111–129
- Rosier L (1997) Exact boundary controllability for the Korteweg-de Vries equation on a bounded domain. *ESAIM Control Optim Calc Var* 2:33–55
- Rosier L, Zhang B-Y (2009) Control and stabilization of the Korteweg-de Vries equation: recent progresses. *J Syst Sci Complex* 22:647–682
- Sivashinsky GI (1977) Nonlinear analysis of hydrodynamic instability in laminar flames – I Derivation of basic equations. *Acta Astronaut* 4:1177–1206
- Smyshlyaev A, Krstic M (2010) Adaptive control of parabolic PDEs. Princeton University Press, Princeton
- Takahashi T (2017) Boundary local null-controllability of the Kuramoto-Sivashinsky equation. *Math Control Signals Syst* 29:Art. 2, 1–21
- Tang S, Chu J, Shang P, Coron, J-M (2018) Asymptotic stability of a Korteweg-de Vries equation with a two-dimensional center manifold. *Adv Nonlinear Anal* 7:497–515
- Xiang S (2018) Small-time local stabilization for a Korteweg-de Vries equation. *Syst Control Lett* 111:64–69
- Xiang S (2019) Null controllability of a linearized Korteweg-de Vries equation by backstepping approach. *SIAM J Control Optim* 57:1493–1515
- Zhang BY (1999) Exact boundary controllability of the Korteweg-de Vries equation. *SIAM J Control Optim* 37:543–565

Bounds on Estimation

Arye Nehorai¹ and Gongguo Tang²

¹Preston M. Green Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, USA

²Department of Electrical Engineering and Computer Science, Colorado School of Mines, Golden, CO, USA

Abstract

We review several universal lower bounds on statistical estimation, including deterministic bounds on unbiased estimators such as Cramér-Rao bound and Barankin-type bound, as well as Bayesian bounds such as Ziv-Zakai bound. We present explicit forms of these bounds, illustrate their usage for parameter estimation in Gaussian additive noise, and compare their tightness.

Keywords

Barankin-type bound · Cramér-Rao bound · Mean-squared error · Statistical estimation · Ziv-Zakai bound

Introduction

Statistical estimation involves inferring the values of parameters specifying a statistical model from data. The performance of a particular statistical algorithm is measured by the error between the true parameter values and those estimated by the algorithm. However, explicit forms of estimation error are usually difficult to obtain except for the simplest statistical models. Therefore, performance bounds are derived as a way of quantifying estimation accuracy while maintaining tractability.

In many cases, it is beneficial to quantify performance using universal bounds that are independent of the estimation algorithms and rely only upon the model. In this regard, universal lower bounds are particularly useful as it provides means to assess the difficulty of performing estimation for a particular model and can act as benchmarks to evaluate the quality of any algorithm: the closer the estimation error of the algorithm to the lower bound, the better the algorithm. In the following, we review three widely used universal lower bounds on estimation: Cramér-Rao bound (CRB), Barankin-type bound (BTB), and Ziv-Zakai bound (ZZB). These bounds find numerous applications in determining the performance of sensor arrays, radar, and nonlinear filtering; in benchmarking various algorithms; and in optimal design of systems.

Statistical Model and Related Concepts

To formalize matters, we define a statistical model for estimation as a family of parameterized probability density functions in $\mathbb{R}^N$: $\{p(x; \theta) : \theta \in \Theta \subset \mathbb{R}^d\}$. We observe a realization of $x \in \mathbb{R}^N$ generated from a distribution $p(x; \theta)$, where $\theta \in \Theta$ is the true parameter to be estimated from data x . Though we assume a single observation x , the model is general enough to encompass multiple independent, identically distributed samples (i.i.d.) by considering the joint probability distribution.

An estimator of θ is a measurable function of the observation $\hat{\theta}(x) : \mathbb{R}^N \rightarrow \Theta$. An unbiased estimator is one such that

$$\mathbb{E}_\theta \left\{ \hat{\theta}(x) \right\} = \theta, \forall \theta \in \Theta. \quad (1)$$

Here we used the subscript θ to emphasize that the expectation is taken with respect to $p(x; \theta)$. We focus on the performance of unbiased estimators in this entry. There are various ways to measure the error of the estimator $\hat{\theta}(x)$. Two typical ones are the error covariance matrix:

$$\mathbb{E}_\theta \left\{ (\hat{\theta} - \theta)(\hat{\theta} - \theta)^T \right\} = \text{Cov}(\hat{\theta}), \quad (2)$$

where the equation holds only for unbiased estimators, and the mean-squared error (MSE):

$$\mathbb{E}_\theta \left\{ \|\hat{\theta}(x) - \theta\|_2^2 \right\} = \text{trace} \left(\mathbb{E}_\theta \left\{ (\hat{\theta} - \theta)(\hat{\theta} - \theta)^T \right\} \right). \quad (3)$$

Example 1 (Signal in additive Gaussian noise (SAGN)) To illustrate the usage of different estimation bounds, we use the following statistical model as a running example:

$$x_n = s_n(\theta) + w_n, n = 0, \dots, N-1. \quad (4)$$

Here $\theta \in \Theta \subset \mathbb{R}$ is a scalar parameter to be estimated and the noise w_n follows i.i.d. Gaussian distribution with mean 0 and known variance σ^2 . Therefore, the density function for x is

$$\begin{aligned} p(x; \theta) &= \prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{(x_n - s_n(\theta))^2}{2\sigma^2} \right\} \\ &= \frac{1}{(\sqrt{2\pi}\sigma)^N} \exp \left\{ -\sum_{n=0}^{N-1} \frac{(x_n - s_n(\theta))^2}{2\sigma^2} \right\}. \end{aligned}$$

In particular, we consider the frequency estimation problem where $s_n(\theta) = \cos(2\pi n\theta)$ with $\Theta = [0, \frac{1}{4})$.

Cramér-Rao Bound

The Cramér-Rao bound (CRB) (Kay 2001a; Van Trees 2001; Stoica and Nehorai 1989) is arguably the most well-known lower bounds on estimation. Define the Fisher information matrix $I(\theta)$ via

$$\begin{aligned} I_{i,j}(\theta) &= \mathbb{E}_{\theta} \left\{ \frac{\partial}{\partial \theta_i} \log p(x; \theta) \frac{\partial}{\partial \theta_j} \log p(x; \theta) \right\} \\ &= -\mathbb{E}_{\theta} \left\{ \frac{\partial^2}{\partial \theta_i \partial \theta_j} \log p(x; \theta) \right\}. \end{aligned}$$

Then for *any* unbiased estimator $\hat{\theta}$, the error covariance matrix is bounded by

$$\mathbb{E}_{\theta} \left\{ (\hat{\theta} - \theta)(\hat{\theta} - \theta)^T \right\} \succeq [I(\theta)]^{-1}, \quad (5)$$

where $A \succeq B$ for two symmetric matrices means $A - B$ is positive semidefinite. The inverse of the Fisher information matrix $\text{CRB}(\theta) = [I(\theta)]^{-1}$ is called the Cramér-Rao bound.

When θ is scalar, $I(\theta)$ measures the expected sensitivity of the density function with respect to changes in the parameter. A density family that is more sensitive to parameter changes (larger $I(\theta)$) will generate observations that look more different when the true parameter varies, making it easier to estimate (smaller error).

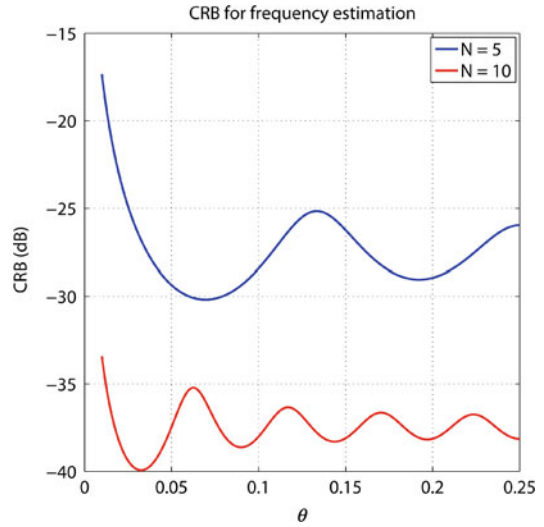
Example 2 For the SAGN model (4), the CRB is

$$\text{CRB}(\theta) = I(\theta)^{-1} = \frac{\sigma^2}{\sum_{n=0}^{N-1} \left[\frac{\partial s_n(\theta)}{\partial \theta} \right]^2}. \quad (6)$$

The inverse dependence on the ℓ_2 norm of signal derivative suggests that signals more sensitive to parameter change are easier to estimate.

For the frequency estimation problem with $s_n(\theta) = \cos(2\pi n\theta)$, the CRB as a function of θ is plotted in Fig. 1.

There are many modifications of the basic CRB such as the posterior CRB (Van Trees 2001; Tichavsky et al. 1998), the hybrid CRB (Rockah and Schultheiss 1987), the modified CRB (D'Andrea et al. 1994), the concentrated



Bounds on Estimation, Fig. 1 Cramér-Rao bound on frequency estimation: $N = 5$ vs. $N = 10$

CRB (Hochwald and Nehorai 1994), and constrained CRB (Stoica and Ng 1998; Gorman and Hero 1990; Marzetta 1993). The posterior CRB takes into account the prior information of the parameters when they are modeled as random variables, while the hybrid CRB considers the case that the parameters contain both random and deterministic parts. The modified CRB and the concentrated CRB focus on handling nuisance parameters in a tractable manner. The application of these CRBs requires a regular parameter space (e.g., an open set in $\mathbb{R}^d$). However, in many cases, the parameter space Θ is a low-dimensional manifold in $\mathbb{R}^d$ specified by equalities and inequalities. In this case, the constrained CRB provides tighter lower bounds by incorporating knowledge of the constraints.

Barankin Bound

CRB is a local bound in the sense that it involves only local properties (the first or second order derivatives) of the log-likelihood function. So if two families of log-likelihood functions coincide at a region near θ^0 , the CRB at θ^0 would be the same, even if they are drastically different in other regions of the parameter space.

However, the entire parametric space should play a role in determining the difficulty of parameter estimation. To see this, imagine that there are two statistical models. In the first model there is another point $\theta^1 \in \Theta$ such that the likelihood family $p(x; \theta)$ behaves similarly around θ^0 and θ^1 , but these two points are not in neighborhoods of each other. Then it would be difficult to distinguish these two points for any estimation algorithm, and the estimation performance for the first statistical model would be bad (an extreme case is $p(x; \theta^0) \equiv p(x; \theta^1)$ in which case the model is non-identifiable; more discussions on identifiability and Fisher information matrix can be found in Hochwald and Nehorai (1997)). In the second model, we remove the point θ^1 and its near neighborhood from Θ , then the performance should get better. However, CRB for both models would remain the same whether we exclude θ^1 from Θ or not. As a matter of fact, $\text{CRB}(\theta^0)$ uses only the fact that the estimator is unbiased in a neighborhood of the true parameter θ^0 .

Barankin bound addresses CRB's shortcoming of not respecting the global structure of the statistical model by introducing finitely many test points $\{\theta^i, i = 1, \dots, M\}$ and ensures that the estimator is unbiased at the neighborhood of θ^0 as well as these test points (Forster and Larzabal 2002). The original Barankin bound (Barankin 1949) is derived for scalar parameter $\theta \in \Theta \subset \mathbb{R}$ and any unbiased estimator $\widehat{g(\theta)}$ for a function $g(\theta)$:

$$\mathbb{E}_\theta (\widehat{g(\theta)} - g(\theta))^2 \geq \sup_{M, \theta^i, a^i} \frac{\left[\sum_{m=1}^M a^i (g(\theta^i) - g(\theta)) \right]^2}{\mathbb{E}_\theta \left[\sum_{m=1}^M a^i \frac{p(x; \theta^i)}{p(x; \theta)} \right]^2} \quad (7)$$

Using (7), we can derive a Barankin-type bound on the error covariance matrix of any unbiased estimator $\hat{\theta}(x)$ for a vector parameter $\theta \in \Theta \subset \mathbb{R}^d$ (Forster and Larzabal 2002):

$$\mathbb{E}_\theta \left\{ (\hat{\theta} - \theta)(\hat{\theta} - \theta)^T \right\} \geq \Phi(B - 11^T)^{-1} \Phi^T, \quad (8)$$

where the matrices are defined via

$$B_{i,j} = \mathbb{E}_\theta \left\{ \frac{p(x; \theta^i)}{p(x; \theta)} \frac{p(x; \theta^j)}{p(x; \theta)} \right\}, \quad 1 \leq i, j \leq M,$$

$$\Phi = [\theta^1 - \theta \dots \theta^M - \theta]$$

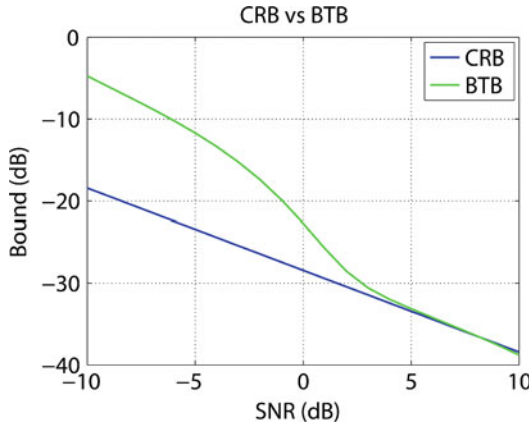
and 1 is the vector in $\mathbb{R}^M$ with all ones. Note that we have used θ^i with a superscript to denote different points in Θ , while θ_i with a subscript to denote the i th component of a point θ .

Since the bound (8) is valid for any M and any choice of test points $\{\theta^i\}$, we obtain the tightest bound by taking the supremum over all finite families of test points. Note that when we have d test points that approach θ in d linearly independent directions, the Barankin-type bound (8) converges to the CRB. If we have more than d test points, however, the Barankin-type bound is always not worse than the CRB. Particularly, the Barankin-type bound is much tighter in the regime of low signal-to-noise ratio (SNR) and small number of measurements, which allows one to investigate the “threshold” phenomena as shown in the next example.

Example 3 For the SAGN model, if we have M test points, the elements of matrix B are of the following form:

$$B_{i,j} = \exp \left\{ \frac{1}{\sigma^2} \sum_{n=0}^{N-1} [s_n(\theta^i) - s_n(\theta)] [s_n(\theta^j) - s_n(\theta)] \right\}$$

In most cases, it is extremely difficult to derive an analytical form of the Barankin bound by optimizing with respect to M and the test points $\{\theta^j\}$. In Fig. 2, we plot the Barankin-type bounds for $s_n(\theta) = \cos(2\pi n\theta)$ for $M = 10$ randomly selected test points. We observe that Barankin-type bound is tighter than the CRB when SNR is small. There is a SNR region around 0 dB that the Barankin-type bound drops drastically. This is usually called the “threshold” phenomenon. Practical systems operate much better in the region above the threshold.



Bounds on Estimation, Fig. 2 Cramér-Rao bound vs. Barankin-type bound on frequency estimation when $\theta^0 = 0.1$. The BTB is obtained using $M = 10$ uniform random points

The basic CRB and BTB belong to the family of deterministic “covariance inequality” bounds in the sense that the unknown parameter is assumed to be a *deterministic* quantity (as opposed to a *random* quantity). Additionally, both bounds work only for unbiased estimators, making them inappropriate performance indicators for biased estimators such as many regularization-based estimators.

Ziv-Zakai Bound

In this section, we introduce the Ziv-Zakai bound (ZZB) (Bell et al. 1997) that is applicable to any estimator (not necessarily unbiased). Unlike the CRB and BTB, the ZZB is a Bayesian bound and the errors are averaged by the prior distribution $p_\theta(\phi)$ of the parameter. For any $a \in \mathbb{R}^d$, the ZZB states that

$$a^T \mathbb{E} \left\{ (\hat{\theta}(x) - \theta)(\hat{\theta}(x) - \theta)^T \right\} a \geq \frac{1}{2} \int_0^\infty \mathcal{V} \left\{ \max_{\delta: a^T \delta = h} \left[\int_{\mathbb{R}^d} (p_\theta(\phi) + p_\theta(\phi + \delta)) \right. \right. \\ \left. \left. \text{times } P_{\min}(\phi, \phi + \delta) d\phi \right] \right\} h dh,$$

where the expectation is taken with respect to the joint disunity $p(x; \theta)p_\theta(\phi)$, $\mathcal{V}\{q(h)\} = \max_{r \geq 0} q(h + r)$ is the valley-filling function,

and $P_{\min}(\phi, \phi + \delta)$ is the minimal probability of error for the following binary hypothesis testing problem:

$$\begin{aligned} H_0 : \theta &= \phi; & x &\sim p(x; \phi) \\ H_1 : \theta &= \phi + \delta; & x &\sim p(x; \phi + \delta) \end{aligned}$$

with

$$\begin{aligned} \Pr(H_0) &= \frac{p_\theta(\phi)}{p_\theta(\phi) + p_\theta(\phi + \delta)} \\ \Pr(H_1) &= \frac{p_\theta(\phi + \delta)}{p_\theta(\phi) + p_\theta(\phi + \delta)}. \end{aligned}$$

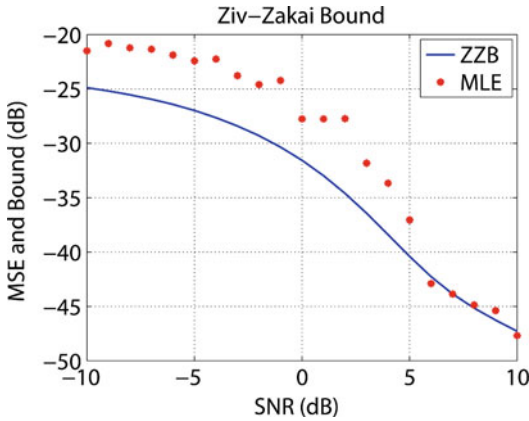
Example 4 For the ASGN model, we assume a uniform prior probability, i.e., $p_\theta(\phi) = 4, \phi \in [0, 1/4]$. The ZZB simplifies to

$$\begin{aligned} \mathbb{E} \left\{ \|\hat{\theta}(x) - \theta\|_2^2 \right\} &\geq \\ \frac{1}{2} \int_0^{\frac{1}{4}} \mathcal{V} \left\{ \left[\int_0^{\frac{1}{4}-h} 8 P_{\min}(\phi, \phi + h) d\phi \right] \right\} h dh. \end{aligned}$$

The binary hypothesis testing problem is to decide which one of two signals is buried in additive Gaussian noise. The optimal detector with minimal probability of error is the minimum distance receiver (Kay 2001b), and the associated probability of error is

$$\begin{aligned} P_{\min}(\phi, \phi + h) &= \\ &= Q \left(\frac{1}{2} \sqrt{\frac{\sum_{n=0}^{N-1} (s_n(\phi) - s_n(\phi + h))^2}{\sigma^2}} \right), \end{aligned}$$

where $Q(h) = \int_h^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$. For the frequency estimation problem, we numerically estimate the integral and plot the resulting ZZB in Fig. 3 together with the mean-squared error for the maximum likelihood estimator (MLE).



Bounds on Estimation, Fig. 3 Ziv-Zakai bound vs. maximum likelihood estimator for frequency estimation

Summary and Future Directions

We have reviewed several important performance bounds on statistical estimation problems, particularly, the Cramér-Rao bound, the Barankin-type bound, and the Ziv-Zakai bound. These bounds provide a universal way to quantify the performance of statistically modeled physical systems that is independent of any specific algorithm.

Future directions of performance bounds on estimation include deriving tighter bounds, developing computational schemes to approximate existing bounds in a tractable way, and applying them to practical problems.

Cross-References

- [Estimation, Survey on](#)
- [Particle Filters](#)

Recommended Reading

Kay SM (2001a), Chapter 2 and 3; Stoica P, Nehorai A (1989); Van Trees HL (2001), Chapter 2.7; Forster and Larzabal (2002); Bell et al. (1997).

Acknowledgments This work was supported in part by NSF Grants CCF-1014908 and CCF-0963742, ONR Grant N000141310050, AFOSR Grant FA9550-11-1-0210.

Bibliography

- Barankin EW (1949) Locally best unbiased estimates. *Ann Math Stat* 20(4):477–501
- Bell KL, Steinberg Y, Ephraim Y, Van Trees HL (1997) Extended Ziv-Zakai lower bound for vector parameter estimation. *IEEE Trans Inf Theory* 43(2):624–637
- D’Andrea AN, Mengali U, Reggiannini R (1994) The modified Cramér-Rao bound and its application to synchronization problems. *IEEE Trans Commun* 42(234):1391–1399
- Forster P, Larzabal P (2002) On lower bounds for deterministic parameter estimation. In: *IEEE international conference on acoustics, speech, and signal processing (ICASSP)*, 2002, Orlando, vol 2. IEEE, pp II–1141
- Gorman JD, Hero AO (1990) Lower bounds for parametric estimation with constraints. *IEEE Trans Inf Theory* 36(6):1285–1301
- Hochwald B, Nehorai A (1994) Concentrated Cramér-Rao bound expressions. *IEEE Trans Inf Theory* 40(2):363–371
- Hochwald B, Nehorai A (1997) On identifiability and information-regularity in parametrized normal distributions. *Circuits Syst Signal Process* 16(1):83–89
- Kay SM (2001a) Fundamentals of statistical signal processing, volume 1: estimation theory. Prentice Hall, Upper Saddle River, NJ
- Kay SM (2001b) Fundamentals of statistical signal processing, volume 2: detection theory. Prentice Hall, Upper Saddle River, NJ
- Marzetta TL (1993) A simple derivation of the constrained multiple parameter Cramér-Rao bound. *IEEE Trans Signal Process* 41(6):2247–2249
- Rockah Y, Schultheiss PM (1987) Array shape calibration using sources in unknown locations – part I: far-field sources. *IEEE Trans Acoust Speech Signal Process* 35(3):286–299
- Stoica P, Nehorai A (1989) MUSIC, maximum likelihood, and Cramér-Rao bound. *IEEE Trans Acoust Speech Signal Process* 37(5):720–741
- Stoica P, Ng BC (1998) On the Cramér-Rao bound under parametric constraints. *IEEE Signal Process Lett* 5(7):177–179
- Tichavsky P, Muravchik CH, Nehorai A (1998) Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Trans Signal Process* 46(5):1386–1396
- Van Trees HL (2001) Detection, estimation, and modulation theory: part 1, detection, estimation, and linear modulation theory. John Wiley & Sons, Hoboken, NJ

BSDE

► Backward Stochastic Differential Equations and Related Control Problems

Building Comfort and Environmental Control

John T. Wen

Electrical, Computer, and Systems Engineering (ECSE), Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract

Heating, ventilation, and air condition (HVAC) systems regulate building temperature and humidity distributions. Sound HVAC system operation is critical for maintaining the comfort level of building occupants, providing the proper operating condition for building equipments and contents, and in general achieving a healthy and safe building environment. HVAC is also a major energy consumer. Balancing comfort and environmental control versus energy consumption is a major theme in building HVAC control. This entry describes the modeling of the HVAC system dynamics in buildings, the HVAC control problem addressing human comfort and energy consumption, and the adaptation to variations in the ambient operating condition.

Keywords

Human comfort control · Temperature control · Humidity control · HVAC control · Intelligent building controls

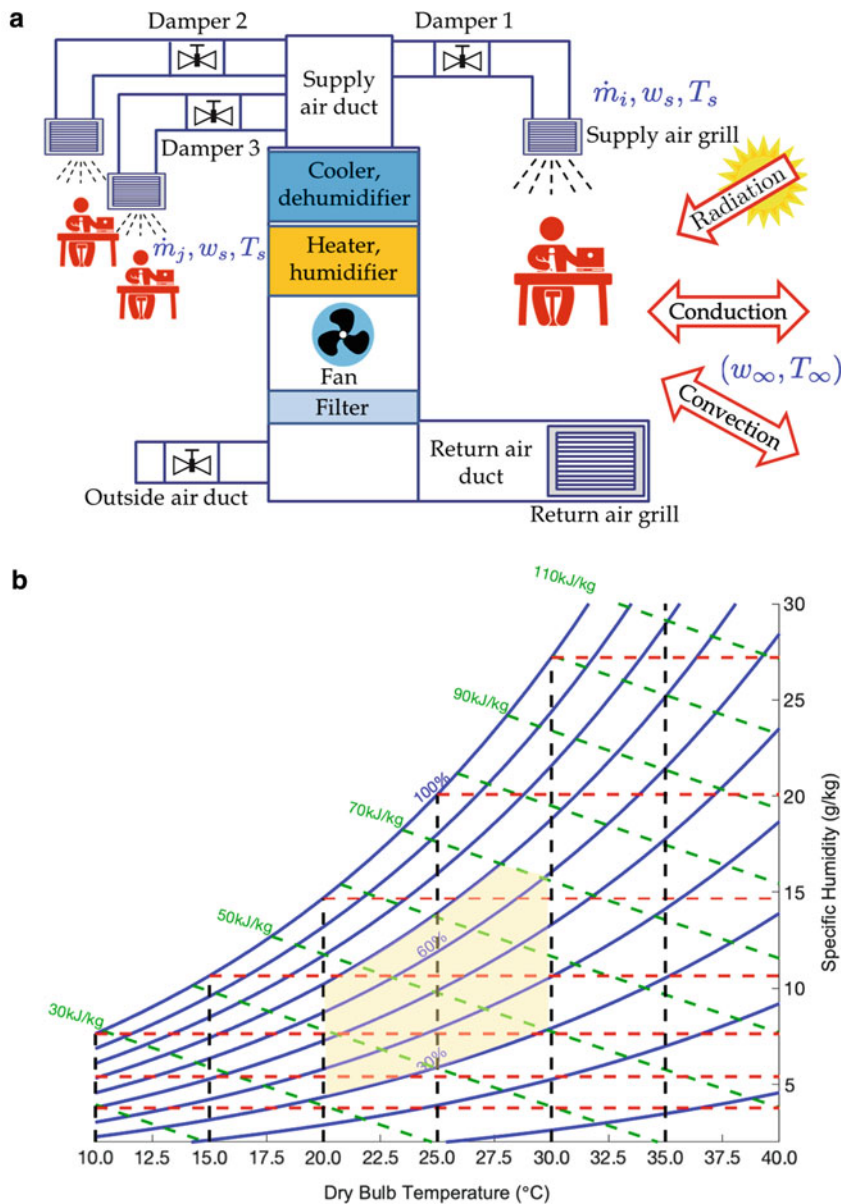
Introduction

The building environment affects its occupants through temperature, humidity, lighting, air quality, noise, etc. Temperature and humidity together are called the *thermohygroscopic* condition. The

building heating, ventilation, and air conditioning (HVAC) system regulates temperature and humidity by adjusting the supply air with given temperature and humidity to achieve the desired operating condition within the building. Residential and commercial buildings account for nearly 30% of the US energy consumption (U.S. Energy Information Administration 2019a), of which more than 50% is due to the HVAC operation (U.S. Energy Information Administration 2019b). Balancing human comfort and energy efficiency is a key building control objective.

A schematic of the HVAC system for a multi-zone building is depicted in Fig. 1a. A mixture of outside air and return air is heated or cooled and then supplied to each zone. The ratio between the outside air and return air is determined by the consideration of air quality and energy consumption. Thermostats and humidistats allow the setting of the target temperature and humidity and adjust the supplied air based on the measured temperature and humidity to achieve the specified conditions.

Moist air is characterized by its heat and moisture content. The psychrometric chart, shown in Fig. 1b, depicts the relationship of various thermodynamical properties, including the dry bulb temperature (air temperature), specific humidity (amount of moisture content in a unit volume of moist air), specific enthalpy (amount of heat in a unit volume of moist air), and relative humidity (amount of moisture in air expressed as a percentage of the amount needed for saturation) (Lide 2005; Urieli 2010). A typical subjective comfort zone is shown as the shaded region resulting from bounds on temperature, relative humidity, and specific enthalpy. Energy consumption of the HVAC system is determined by the cooling and heating of the supplied air, and the energy needed to deliver the supplied air into the zones by using blower fans or radiator pumps. The source of cooling is typically from an air conditioning system which operates on a vapor compression cycle (VCC). The design for such systems is usually for the steady state operation which is well understood. However, for systems with significant transients, e.g., short duration load demands, say during a heat wave, there may need to be



Building Comfort and Environmental Control, Fig. 1 HVAC system for building occupant comfort control. (a) Schematics of a building HVAC system. The state of the i th zone, given by (w_i, T_i) , is affected by the state of adjacent zones, (w_j, T_j) , the ambient condition (w_∞, T_∞) , the properties of the supply air (w_s, T_s) and flow rate

$\dot{m}_i$. (b) Psychrometric chart showing relationship between thermodynamical properties of moist air, including the dry bulb temperature (horizontal axis), specific humidity (vertical axis), specific enthalpy (green dash line), and relative humidity (blue curves). The shaded region indicates a typical comfort zone for human occupants

active control of these subsystems (Rasmussen et al. 2018).

HVAC Dynamical Model

Building dynamics as related to occupancy comfort and environmental condition is characterized by the heat and moisture transfer. The underlying physics is governed by the conservation of mass and energy. These are nonlinear partial differential equations that require the use numerical approximation for their solution (such as computation fluid dynamics) (U.S. Department of Energy 2019). Such models are useful for design and assessment, but are not suitable for control design due to their complexity and computation overhead. For the HVAC control system analysis and design, the building dynamics is typically modeled as a lumped parameter system. In this case, the building is divided into *zones* with the average temperature and humidity characterizing the state of the zone (Goyal and Barooah 2012; Okaeme et al. 2018).

The humidity dynamics in a zone is governed by the mass transfer between zones (including the ambient). Denote the mass of dry air in the i th zone by the constant M_i . Let the humidity ratio of the zone be w_i which is the amount of moisture in a unit volume of dry air. The mass balance is governed by moisture diffusion (Fick's Law):

$$M_i \dot{w}_i = - \sum_j G_{ij}(w_i - w_j) - G_{i0}(w_i - w_\infty) + v_{m_i} \quad (1)$$

where the sum is over all neighboring zones, G_{ij} is the mass conductance between zones i and j , G_{i0} is the mass conductance to the ambient, w_∞ is the humidity ratio of the ambient, and v_{m_i} is the moisture injected by the supply air into the i th zone: $v_{m,i} = \dot{m}_i \frac{(w_s - w_i)}{1 + w_s} + v_{m,d}$ with $\dot{m}_i$ as the mass flow rate of the supply air, w_s the humidity ratio of the supply air, and $v_{m,d}$ the exogenous moisture input, e.g., occupant body evaporation.

Similarly, temperature is governed by the heat and mass transfer between zones (including the ambient). For heat transfer, each zone is

considered as a thermal capacitance (sometimes called thermal mass or thermal inertia) connected to other thermal capacitances through thermal conductances. Let β be the latent heat of vaporization for water, the energy balance equation for the i th zone based on heat conduction is

$$C_i \dot{T}_i = - \sum_j K_{ij}(T_i - T_j) - K_{i0}(T_i - T_\infty) + v_{e_i} + \beta G_{i0}(w_i - w_\infty) + \beta \sum_j G_{ij}(w_i - w_j) \quad (2)$$

where $C_i = c_p M_i$ is the thermal capacitance of the i th zone (c_p is the specific heat), K_{ij} is the thermal conductance between zones i and j , j sums over all zones adjacent to zone i , and v_{e_i} is the heat input into the zone: $v_{e,i} = \dot{m}_i c_p (T_s - T_i) + v_{e,d}$ with T_s the supplied air temperature and $v_{e,d}$ the exogenous heat input, e.g., solar heat gain, heat from equipment and human occupants, etc. Sometimes thermal capacitances in walls are also modeled, e.g., the commonly used 3R2C model considers a wall as two thermal capacitors sandwiched between three thermal resistors (thermal resistance is the reciprocal to thermal conductance).

The humidity model of a multi-zone building may be represented as a moisture exchange graph with each zone as nodes and mass conductors as links between adjacent nodes that allow mass transfer. Similarly, the heat transfer is represented as a thermal graph with thermal capacitors as nodes and thermal conductors as links between nodes. The humidity and thermal equations for a multi-zone building may then be written in the following compact form:

$$M \dot{w} = -LGL^T w + B_{m0} w_\infty + v_m \quad (3a)$$

$$C \dot{T} = -DKD^T T + \beta LGL^T w - \beta B_{m0} w_\infty + B_{e0} T_\infty + v_e \quad (3b)$$

where (w, T) denote the vectors of humidity ratios and temperatures (including zone and wall temperatures), (v_m, v_e) are the moisture and heat input into the room, M and C are diagonal

positive definite matrices consisting of zone air masses and thermal capacitances, G and K are diagonal positive definite matrices consisting of link mass and thermal conductances, B_{m0} and $B_{e,0}$ are column vectors with mass and thermal conductances connecting nodes to the ambient. For the moisture exchange graph, L is the incidence matrix and LGL^T is the weight Laplacian of the graph. For the thermal graph, D is the incidence matrix and DKD^T is the weighted Laplacian (Deng et al. 2010; Mukherjee et al. 2012).

HVAC Control Systems

The HVAC system of a multi-zone building viewed in the control context may be depicted as in Fig. 2. As described in the modeling section above, the state of the building consists of temperatures in the thermal capacitors and humidity ratios in all zones. The control input may be the heat and moisture delivered or removed from each zone, or the mass flow rate of the supplied air. In some cases, temperature and humidity of the supplied air are independently controlled (e.g., with heater/air conditioner or humidifier/dehumidifier). Zone temperature and humidity are measured through thermometers and hygrometers. These measurements are used in thermostat and humidistat to maintain the specified temperature and humidity set points in all zones by regulating the heat and moisture injection/removal in the zone. The HVAC control

problem is to use the control input to achieve the target temperature and humidity profiles. The challenge of the control problem include:

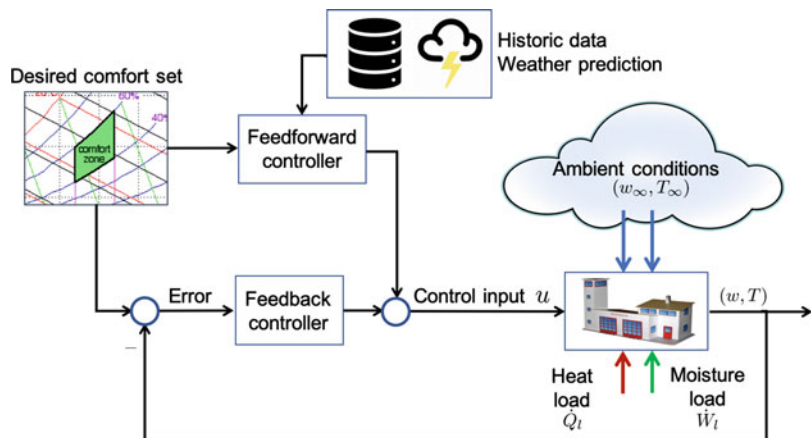
1. Coupled dynamics, i.e., each zone is affected by adjacent zones;
2. Unknown, uncertain, or time-varying model parameters (e.g., opening of windows or doors);
3. Time-varying exogenous inputs, e.g., ambient (outdoor) weather condition, heat and moisture from occupants and equipment within the zone, solar heat gain, etc. Some of the variation may be known ahead of time, e.g., through the weather forecast or room usage pattern. This knowledge may be incorporated into a predictive control strategy.

Existing commercial off-the-shelf building temperature and humidity control systems typically treat each zone individually, and apply an on-off control strategy based on the measured temperature and humidity in that zone. Hysteresis or dead zone is applied to the temperature and humidity error signal to avoid chattering in the control signal. Supervisory control schedules the temperature and humidity set points based on room usage to reduce energy consumption and enhance occupant comfort. For example, the temperature set point in an office space may be raised before the start of the work day and lowered during the off-hours.

Most of the building HVAC control literature deals with temperature control problem, with

Building Comfort and Environmental Control,

Fig. 2 HVAC control architecture of a multi-zone building. The humidity ratios and temperatures in all the zones are denoted by vectors (w, T) . The outside humidity ratio and temperature are denoted by (w_∞, T_∞)



the latent heat of vaporization treated as a disturbance in the temperature dynamics (3a). Numerous advanced control strategies have been proposed to address the multi-zone coupling in the dynamical model and exogenous heat input variation (Afram and Janabi-Sharifi 2014). They span the range of control methodologies, including classical linear control, advanced nonlinear control, and logic and neural network-based control. Robust, adaptive, or learning control has all been proposed to address the uncertain and possibly time-varying model parameters. Building temperature dynamics modeled with thermal resistance-capacitance (RC) elements possesses an inherent passivity property, as in passive electrical circuits (inductor-resistor-capacitor) or passive mechanical structures (mass-spring-damper). If the product of the input and output of a system is interpreted as the power delivered to the system, then for a passive system, the energy is always delivered to the system, meaning that the system either conserves or dissipates energy. For the building thermal model, the rate of the heat injection into the zones, v_e , and temperatures in the zones, $B^T T$, form a strictly passive input/output pairs (similar to voltage and current into a passive circuit, or collocated force and velocity in a mechanical structure) (Mukherjee et al. 2012). It follows from the passivity theorem that any passive output feedback would ensure stability. This means the existing decentralized control that ignores the multi-zone coupling is always stabilizing. Using this passivity framework, it may be shown that under constant but unknown exogenous input condition, high-gain feedback such as an on-off type of control, or integral feedback to adaptively remove any steady state bias, would asymptotically converge to a constant set point even in the complete absence of any model parameter information. When the exogenous input is time-varying, e.g., fluctuating occupancy and equipment usage or changing weather condition, feedback control alone could result in significant errors, possibly leading to occupant discomfort and energy inefficiency. If this variation is known beforehand (e.g., through short-term weather forecast or known

zone occupancy schedule), it may be used as a feedforward to preemptively adjust the zone temperature to improve the temperature tracking accuracy (utilizing the building thermal capacitances). If the model information is available, at least approximately, it may be used in a finite horizon optimization, as in model predictive control (MPC) (Ma et al. 2012; Mirakhorli and Dong 2016), to take the ambient variation into account to minimize a combination of tracking error and energy consumption. The optimization approach also addresses actuator limits such as the allowable range of fan speed. If the optimization variable is the heat into each zone, the optimization becomes a linear programming problem. If the optimization variable is the mass flow rate (controlled through the fan speed), the resulting optimization is bilinear, and the problem may become nonconvex. The optimization problem may be solved in a distributed manner based on schemes such as alternating direction method of multipliers (ADMM) (Boyd 2018). This is particularly useful to balance the comfort preference of multiple users and the desire for low energy cost of the building operator (Gupta et al. 2015). Voting and auction type of schemes have also been proposed in a multiuser environment (Erickson and Cerpa 2012). When the exogenous input exhibits a certain known structure (e.g., cyclic variation of occupancy, weather pattern similar to the past), the Lyapunov function motivated by the passivity property allows adaptation or learning of the feedforward control (Peng et al. 2016; Minakais et al. 2019).

The passivity approach also extends to the control of both temperature and humidity (Okaeme et al. 2018). If heat and humidity into each zone are independently controlled with the rate of heat and moisture injection as input, the extension of the passivity approach is straightforward with the corresponding output to be zone temperature and humidity. If the mass flow rate is the input, then the temperature and humidity measurements need to be judiciously combined to form a passive output.

Summary and Future Directions

HVAC is an important component of the overall building control system. It affects occupant comfort, cost of building operation, and interacts with other building control subsystems such as air quality, noise, and lighting. Current trend in HVAC system design is to incorporate better modeling and data analytics to improve the understanding and prediction of building usage, human occupant preferences, and variations in the building environment. Toward this direction of intelligent building control, more sensors than ever are deployed in buildings, including stationary sensors such temperature and humidity sensors and mobile sensors such as human wearable devices. These sensors are networked together using Internet of Things (IoT) technologies to facilitate data collection and analysis, which in turn are used to guide building operation and decision-making. Buildings are also becoming more interactive, with users able to provide individual feedback on comfort and preference levels. In this highly interconnected scenario, occupant privacy and building security are key concerns; any building monitoring and data gathering system must provide sufficient safeguard.

Cross-References

- [Building Control Systems](#)
- [Distributed Optimization](#)
- [Model Predictive Control for Power Networks](#)
- [Passivity-Based Control](#)

Bibliography

- Afram A, Janabi-Sharifi F (2014) Theory and applications of HVAC control systems – a review of model predictive control (MPC). *Build Environ* 72:343–355
- Boyd SP (2018) ADMM. <https://stanford.edu/~boyd/admm.html>
- Deng K, Barooah P, Mehta PG, Meyn SP (2010) Building thermal model reduction via aggregation of states. In: *Proceedings, 2010 American control conference*, pp 5118–5123
- Erickson VL, Cerpa AE (2012) Thermovote: participatory sensing for efficient building HVAC conditioning. In: *Proceedings of the Fourth ACM workshop on embedded sensing systems for energy-efficiency in buildings*, pp 9–16
- Goyal S, Barooah P (2012) A method for model-reduction of non-linear thermal dynamics of multi-zone buildings. *Energy Build* 47:332–340
- Gupta SK, Kar K, Mishra S, Wen JT (2015) Collaborative energy and thermal comfort management through distributed consensus algorithms. *IEEE Trans Autom Sci Eng* 12(4):1285–1296
- Lide DR (ed) (2005) *CRC handbook on chemistry and physics*, internet version 2005. CRC Press, Boca Raton
- Ma Y, Borrelli F, Hencsey B, Coffey B, Bengesa S, Haves P (2012) Model predictive control for the operation of building cooling systems. *IEEE Trans Control Syst Technol* 20(3):796–803
- Minakais M, Mishra S, Wen JT (2019) Database-driven iterative learning for building temperature control. *IEEE Trans Autom Sci Eng* 16(4):1896–1906
- Mirakhorli A, Dong B (2016) Occupancy behavior based model predictive control for building indoor climate a critical review. *Energy Build* 129:499–513
- Mukherjee S, Mishra S, Wen JT (2012) Building temperature control: a passivity-based approach. In: *Proceedings, 2012 IEEE conference on decision and control*, pp 6902–6907
- Okaeme CC, Mishra S, Wen JT (2018) Passivity-based thermohygrometric control in buildings. *IEEE Trans Control Syst Technol* 26(5):1661–1672
- Peng C, Zhang W, Tomizuka M (2016) Iterative design of feedback and feedforward controller with input saturation constraint for building temperature control. In: *2016 American control conference (ACC)*, pp 1241–1246
- Rasmussen B, Price C, Koeln J, Keating B, Alleyne A (2018) HVAC system modeling and control: vapor compression system modeling and control. In: *Intelligent building control systems*. Springer, Cham
- Urieli I (2010) Engineering thermodynamics- a graphical approach. In: *ASCE Annual conference & exposition*, Louisville
- US Department of Energy (2019) *EnergyPlus Version 9.1.0 documentation: engineering reference*
- US Energy Information Administration (2019a) *Annual energy outlook 2019, with projection at 2050*. Technical report. www.eia.gov/aeo
- US Energy Information Administration (2019b) *How much energy is consumed in U.S. residential and commercial buildings?* <https://www.eia.gov/tools/faqs/faq.php?id=86&t=1>

Building Control Systems

James E. Braun

Purdue University, West Lafayette, IN, USA

Abstract

This entry provides an overview of systems and issues related to providing optimized

controls for commercial buildings. It includes a description of the evolution of the control systems over time, typical equipment and control variables, typical two-level hierarchal structure for feedback and supervisory control, definition of the optimal supervisory control problem, references to typical heuristic control approaches, and a description of current and future developments.

Keywords

Building automation systems (BAS) · Cooling plant optimization · Energy management and controls systems (EMCS) · Intelligent building controls

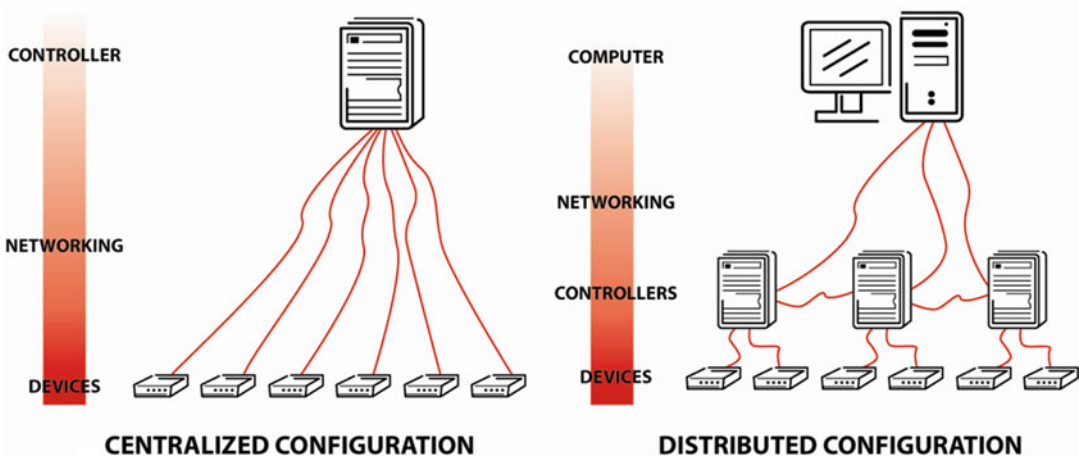
Introduction

Computerized control systems were developed in the 1980s for commercial buildings and are typically termed energy management and control systems (EMCS) or building automation systems (BAS). They have been most successfully applied to large commercial buildings that have hundreds of building zones and thousands of control points. Less than about 15 % of commercial buildings have EMCS, but they serve about 40 % of the floor area. Small commercial buildings tend not to have an EMCS, although there is a recent

trend towards the use of wireless thermostats with cloud-based energy management solutions.

EMCS architectures for buildings have evolved from centralized to highly distributed systems as depicted in Fig. 1 in order to reduce wiring costs and provide more modular solutions. The development of open communications protocols, such as BACNet, has enabled the use of distributed control devices from different vendors and improved the cost-effectiveness of ECMS. There has also been a recent trend towards the use of existing enterprise networks to reduce system installed costs and to more easily allow remote access and control from any Internet accessible device.

An EMCS for a large commercial building can automate the control of many of the building and system functions, including scheduling of lights and zone thermostat settings according to occupancy patterns. Security and fire safety systems tend to be managed using separate systems. In addition to scheduling, an EMCS manages the control of individual equipment and subsystems that provide heating, ventilation, and air conditioning of the building (HVAC). This control is achieved using a two-level hierarchical structure of local-loop and supervisory control. Local-loop control of individual set points is typically implemented using individual proportional-integral (PI) feedback algorithms that manipulate



Building Control Systems, Fig. 1 Evolution from centralized to distributed network architectures

individual actuators in response to deviations from the set points. For example, supply air temperature from a cooling coil is controlled by adjusting a valve opening that provides chilled water to the coil. The second level of supervisory control specifies the set points and other modes of operation that depend on time and external conditions.

Each local-loop feedback controller acts independently, but their performance can be coupled to other local-loop controllers if not tuned appropriately. Adaptive tuning algorithms have been developed in recent years to enable controllers to automatically adjust to changing weather and load conditions. There are typically a number of degrees of freedom in adjusting supervisory control set points over a wide range while still achieving adequate comfort conditions. Optimal control of supervisory set points involves minimizing a cost function with respect to the free variables and subject to constraints. Although model-based, control optimization approaches are not typically employed in buildings, they have been used to inform the development and assessment of some heuristic control strategies. Most commonly, strategies for adjusting supervisory control variables are established at the control design phase based on some limited analysis of the HVAC system and specified as a sequence of operations that is programmed into the EMCS.

Systems, Equipment, and Controls

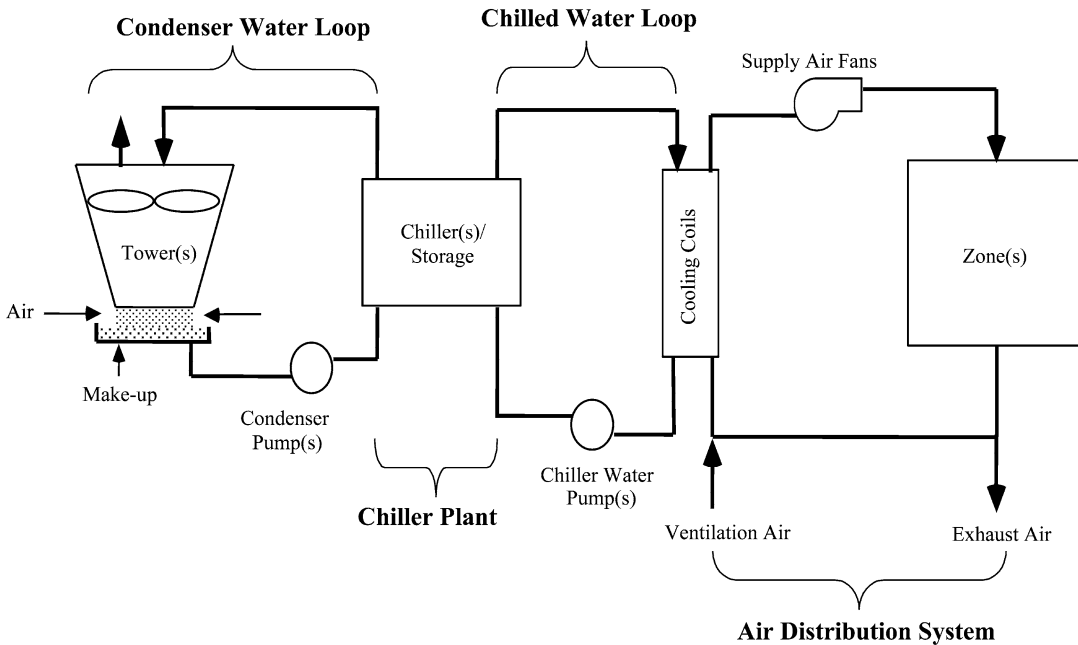
The greatest challenges and opportunities for optimizing supervisory control variables exist for centralized cooling systems that are employed in large commercial buildings because of the large number of control variables and degrees of freedom along with utility rate incentives. A simplified schematic of a typical centralized cooling plant is shown in Fig. 2 with components grouped under air distribution, chilled water loop, chiller plant, and condenser water loop.

Typical **air distribution systems** include VAV (variable-air volume) boxes within the zones, air-handling units, ducts, and controls.

An air-handling unit (AHU) provides the primary conditioning, ventilation, and flow of air and includes cooling and heating coils, dampers, fans, and controls. A single air handler typically serves many zones and several air handlers are utilized in a large commercial building. For each AHU, outdoor ventilation air is mixed with return air from the zones and fed to the cooling coil. Outdoor and return air dampers are typically controlled using an economizer control that selects between minimum and maximum ventilation air depending upon the condition of the outside air. The cooling coil provides both cooling and dehumidification of the process air. The air outlet temperature from the coil is controlled with a local feedback controller that adjusts the flow of water using a valve. A supply fan and return fan (not shown in Fig. 2) provide the necessary air-flow to and from the zones. With a VAV system, zone temperature set points are regulated using a feedback controller applied to dampers within the VAV boxes. The overall air flow provided by the AHU is typically controlled to maintain a duct static pressure set point within the supply duct.

The **chilled water loop** communicates between the cooling coils within the AHUs and chillers that provide the primary source for cooling. It consists of pumps, pipes, valves, and controls. Primary/secondary chilled water systems are commonly employed to accommodate variable-speed pumping. In the primary loop, fixed-speed pumps are used to provide relatively constant chiller flow rates to ensure good performance and reduce the risk of evaporator tube freezing. Individual pumps are typically cycled on and off with a chiller that it serves. The secondary loop incorporates one or more variable-speed pumps that are typically controlled to maintain a set point for chilled water loop differential pressure between the building supplies and returns.

The primary source of cooling for the system is typically provided by one or more **chillers** that are arranged in parallel and have dedicated pumps. Each chiller has an on-board local-loop feedback controller that adjusts its cooling capacity to maintain a specified set point for chilled water supply temperature. Additional chiller con-



Building Control Systems, Fig. 2 Schematic of a chilled water cooling system

Control variables include the number of chillers operating and the relative loading for each chiller. The relative loading can be controlled for a given total cooling requirement by utilizing different chilled water supply set points for constant individual flow or by adjusting individual flows for identical set points. Chillers can be augmented with thermal storage to reduce the amount of chiller power required during occupied periods in order to reduce on-peak energy and power demand costs. The thermal storage medium is cooled during the unoccupied, nighttime period using the chillers when electricity is less expensive. During occupied times, a combination of the chillers and storage are used to meet cooling requirements. Control of thermal storage is defined by the manner in which the storage medium is charged and discharged over time.

The **condenser water loop** includes cooling towers, pumps, piping, and controls. Cooling towers reject energy to the ambient air through heat transfer and possibly evaporation (for wet towers). Larger systems tend to have multiple cooling towers with each tower having multiple cells that share a common sump with individual

fans having two or more speed settings. The number of operating cells and tower fan speeds are often controlled using a local-loop feedback controller that maintains a set point for the water temperature leaving the cooling tower. Typically, condenser water pumps are dedicated to individual chillers (i.e., each pump is cycled on and off with a chiller that it serves).

In order to better understand building control variables, interactions, and opportunities, consider how controls change in response to increasing building cooling requirements for the system of Fig. 2. As energy gains to the zones increase, zone temperatures rise in the absence of any control changes. However, zone feedback controllers respond to higher temperatures by increasing VAV box airflow through increased damper openings. This leads to reduced static pressure in the primary supply duct, which causes the AHU supply fan controller to create additional airflow. The greater airflow causes an increase in supply air temperatures leaving the cooling coils in the absence of any additional control changes. However, the supply air temperature feedback controllers respond by opening the cooling coil

valves to increase water flow and the heat transfer to the chilled water (the cooling load). For variable-speed pumping, a feedback controller would respond to decreasing pressure differential by increasing the pump speed. The chillers would then experience increased loads due to higher return water temperature and/or flow rate that would lead to increases in chilled water supply temperatures. However, the chiller controllers would respond by increasing chiller cooling capacities in order to maintain the chilled water supply set points (and match the cooling coil loads). In turn, the heat rejection to the condenser water loop would increase to balance the increased energy removed by the chiller, which would increase the temperature of water leaving the condenser. The temperature of water leaving the cooling tower would then increase due to an increase in its energy water temperature. However, a feedback controller would respond to the higher condenser water supply temperature and increase the tower airflow. At some load, the current set of operating chillers would not be sufficient to meet the load (i.e., maintain the chilled water supply set points) and an additional chiller would need to be brought online.

This example illustrated how different local-loop controllers might respond to load changes in order to maintain individual set points. Supervisory control might change these set points and modes of operation. At any given time, it is possible to meet the cooling needs with any number of different modes of operation and set points leading to the potential for control optimization to minimize an objective function.

The system depicted in Fig. 2 and described in the preceding paragraphs represents one of many different types of systems employed in commercial buildings. Medium-sized commercial buildings often employ multiple direct expansion (DX) cooling systems where refrigerant flows between each AHU and an outdoor condensing unit that employs variable capacity compressors. The compressor capacity is typically controlled to maintain a supply air temperature set point, which is still available as a supervisory control variable. However, the other condensing unit controls (e.g., condensing fans, expansion valve)

are typically prepackaged with the unit and not available to the EMCS. For smaller commercial buildings, rooftop units (RTUs) are typically employed that contain a prepackaged AHU, refrigeration cycle, and controls. Each RTU directly cools the air in a portion of the building in response to an individual thermostat. The capacity control is typically on/off staging of the compressor and constant volume air flow is mostly commonly employed. In this case, the only free supervisory control variables are the thermostat set points. In general, the degrees of freedom for supervisory control decrease in going from chilled water cooling plants to DX system to RTUs. In addition, the utility rate incentives for taking advantage of thermal storage and advanced controls are greater for large commercial building applications.

Optimal Supervisory Control

In commercial buildings, it is common to have electric utility rates that have energy and demand charges that vary with time of use. The different rate periods can often include on-peak, off-peak, and mid-peak periods. For this type of rate structure, the time horizon necessary to truly minimize operating costs extends over the entire month. In order to better understand the control issues, consider the general optimal control problem for minimizing monthly electrical utility charges associated with operating an all-electric cooling system in the presence of time-of-use and demand charges. The dynamic optimization involves minimizing

$$J = \sum_{p=1}^{\text{rate periods}} J_p + R_{d,a} \max [P_k]_{k=1 \text{ to } N_{\text{month}}}$$

with respect to a trajectory of controls $\vec{u}_k, k = 1 \text{ to } N_{\text{month}}$, $\vec{M}_k, k = 1 \text{ to } N_{\text{month}}$ where

$$J_p = R_{e,p} \sum_{j=1}^{N_p} P_{p,j} \Delta t + R_{d,p} \max [P_{p,j}]_{j=1 \text{ to } N_p}$$

with the optimization subject to the following general constraints

$$\begin{aligned} P_k &= P(\vec{f}_k, \vec{u}_k, \vec{M}_k) \\ \vec{x}_k &= x(\vec{x}_{k-1}, \vec{f}_k, \vec{u}_k, \vec{M}_k) \\ \vec{u}_{k,\min} &\leq \vec{u}_k \leq \vec{u}_{k,\max} \\ \mathbf{x}_{k,\min} &\leq \mathbf{x}_k \leq \mathbf{x}_{k,\max} \\ \vec{y}_k(\vec{f}_k, \vec{u}_k, \vec{M}_k) &\leq \vec{y}_{k,\max}(\vec{f}_k, \vec{u}_k, \vec{M}_k) \end{aligned}$$

where J is the monthly electrical cost (\$), the subscript p denotes that a quantity is limited to a particular type of rate period p (e.g., on-peak, off-peak, mid-peak), $R_{d,p}$ is an anytime demand charge (\$/kW) that is applied to the maximum power consumption occurring over the month P_k is average building power (kW) for stage k within the month, N_{month} is the number of stages in the month, $R_{e,p}$ is the unit cost of electrical energy (\$/kWh) for rate period type p , Δt is the length of the stage (h), N_p is the number of stages within rate period type p in the month, $R_{d,p}$ is a rate period specific demand charge (\$/kW) that is applied to the maximum power consumption occurring during the month within rate period p , $\mathbf{f}_k$ is a vector of uncontrolled inputs that affect building power consumption (e.g., weather, internal gains), $\vec{u}_k$ is a vector of continuous supervisory control variables (e.g., supply air temperature set point), $\mathbf{M}_k$ is a vector of discrete supervisory control variables (chiller on/off controls), $\mathbf{x}_k$ is a vector of state variables, $\mathbf{y}_k$ is a vector of outputs, and subscripts min and max denote minimum and maximum allowable values.

The state variables could characterize the state of a storage device such as a chilled water or ice storage tank. In this case, the states would be constrained between limits associated with the device's practical storage potential. When variations in zone temperature set points are considered within an optimization, then state variables associated with the distributed nature of energy storage within the building structure are important to consider. The outputs are additional quantities of interest, such as equipment cooling capacities, occupant comfort conditions, etc., that often need to be constrained. In order to

implement a model-based predictive control scheme, it would be necessary to have models for the building power, state variables, and outputs in terms of the control and uncontrolled variables. The uncontrolled variables would generally include weather (temperature, humidity, solar radiation) and internal gains due to lights and occupants, etc., that would need to be forecasted over a prediction horizon.

It is not feasible to solve this type of monthly optimization problem for buildings for a variety of reasons, including that forecasting of uncontrolled inputs beyond a day is unreliable. Also, it is very costly to develop the models necessary to implement a model-based control approach of this scale for a particular building. However, it is instructive to consider some special cases that have led to some practical control approaches. First of all, consider the problem of optimizing only the cooling plant supervisory control variables when energy storage effects are not important. This is typically the case for typical systems that do not include ice or chilled water storage. For this scenario, the future does not matter and the problem can be reformulated as a static optimization problem, such that for each stage k the goal is to minimize the building power consumption, $J = P_k$, with respect to the current supervisory control variables, $\vec{u}_k$ and $\mathbf{M}_k$, and subject to constraints. ASHRAE (2011) presents a number of heuristic approaches for adjusting supervisory control variables that have been developed through consideration of this type of optimization problem. This includes algorithms for adjusting cooling tower fan settings, chilled water supply air set points, and chiller sequencing and loading.

Other heuristic approaches have been developed (e.g., Braun 2007; ASHRAE 2011) for controlling the charging and discharging of ice or chilled water storage that were derived from a daily optimization formulation. For the case of real-time pricing of energy, heuristic charging and discharging strategies were derived from minimizing a daily cost function

$$J_{\text{day}} = \sum_{k=1}^{N_{\text{day}}} R_{e,k} P_k \Delta t$$

with respect to a trajectory of charging and discharging rates, subject to a constraint of equal beginning and ending storage states along with other constraints previously described. For the case of typical time-of-use (e.g., on-peak, off-peak) or real-time pricing energy charges with demand charges, heuristic strategies have been developed based on the same form of the daily cost function above with an added demand cost constraint $R_{d,k}P_k \leq TDC$ where TDC is a target demand cost that is set heuristically at the beginning of each billing period and updated at each stage as $TDC_{k+1} = \max(TDC_k, R_{d,k}P_k)$. The heuristic storage control strategies can be readily combined with heuristic strategies for the cooling plant components.

There has been a lot of interest in developing practical methods for dynamic control of zone temperature set points within the bounds of comfort in order to minimize the utility costs. However, this is a very difficult problem and so this remains in the research realm for the time being with limited commercial success.

Summary and Future Directions

Although there is great opportunity for reducing energy use and operating costs in buildings through optimal supervisory control, it is rarely implemented in practice because of high costs associated with engineering site-specific solutions. Current efforts are underway to develop scalable approaches that utilize general methods for configuring and learning models needed to implement model-based predictive control (MPC). The current thinking is that solutions for optimal supervisory control will be implemented in the cloud and overlay on existing building automation systems (BMS) through the use of universal middleware. This will reduce the cost of implementation compared to programming within existing BMS. There is also a need to reduce the cost of the additional sensors needed to implement MPC. One approach involves the use of virtual sensors that employ models with low-cost sensor inputs to provide

higher value information that would normally require expensive sensors to obtain.

Cross-References

- [Building Energy Management System](#)
- [Model Predictive Control in Practice](#)
- [PID Control](#)

Bibliography

- ASHRAE (2011) Supervisory control strategies and optimization. In: 2011 ASHRAE handbook of HVAC applications, chap 42. ASHRAE, Atlanta, GA
- Braun JE (2007) A near-optimal control strategy for cool storage systems with dynamic electric rates. HVAC&R Res 13(4):557–580
- Li H, Yu D, Braun JE (2011) A review of virtual sensing technology and application in building systems. HVAC&R Res 17(5):619–645
- Mitchell JW, Braun JE (2013) Principles of heating ventilation and air conditioning in buildings. Wiley, Hoboken
- Roth KW, Westphalen D, Feng MY, Llana P, Quartararo L (2005) Energy impact of commercial building controls and performance diagnostics: market characterization, energy impact of building faults, and energy savings potential. TIA report no. D0180
- Wang S, Ma Z (2008) Supervisory and optimal control of building HVAC systems: a review. HVAC&R Res 14(1):3–32

Building Energy Management System

Prabir Barooah

MAE-B 324, University of Florida, Gainesville, FL, USA

Abstract

This entry provides an overview of building energy management systems (BEMS). It includes a description of the communication and control architectures typically used for energy management, definition of the optimal supervisory control problem, and a description

of current and future developments in optimal energy management.

Keywords

Building energy management · HVAC · Indoor climate control · Model predictive control · Inverse modeling · Predictive control · Optimization

Introduction

A building automation system (BAS) enables building operators to manage the indoor environment control system, along with fire and safety system and other auxiliary functions such as audio-visual systems in a building. The phrase building energy management system (BEMS) is sometimes used interchangeably with BAS, though energy management is only one aspect of a building's control system. Indoor environment control, which includes lighting and heating, ventilation, and air conditioning (HVAC), has a strong impact on energy use, and this may explain the meshing of the two terms. BEMS are typically used in large commercial buildings, though in recent times there is a trend toward smaller commercial buildings.

The number of possible types and configurations of HVAC systems in large buildings is enormous. In this section we will limit our discussion to single duct variable air volume (VAV), chilled water-based HVAC systems. Figure 1 shows a schematic of such a system. Constant air volume systems, in which the volume of air supplied to the building interior is constant over time, are gradually being phased out. The system shown is a single duct system, since the conditioned air is supplied to the zones through a single duct. Some buildings employ a dual duct system, in which outdoor air (OA) is supplied through a separate, dedicated OA duct. The system shown in the figure is a hydronic system since it uses water to transfer energy: chilled water produced in a chiller is supplied to one or more air handling units (AHUs) that cools and dehumidifies the air supplied to the interior of the building. Chilled

water-based systems are common in large buildings and even in some medium-sized buildings if they are part of a campus. In case of a campus, chilled water is produced in a chiller plant with multiple chillers. In cold and dry climates that do not require cooling and dehumidification, there is no cooling/dehumidification coil in the AHUs. Only heating coils are used, which may use heating hot water (HHW) or electric heating elements. Many buildings use packaged rooftop units (RTUs) that use a vapor compression refrigeration cycle to directly cool and dehumidify air. These systems are referred to as “DX” (Direct eXpansion) systems. DX systems are common in small and medium buildings that typically do not have BEMS.

Control Algorithms in Current BEMS

There are many BEMS vendors, such as Siemens, Johnson Controls, and Automated Logic. Almost all of these BEMS vendors also offer their HVAC equipment and controller hardware. The larger vendors offer their BEMS not simply as products but also solutions that can integrate HVAC equipment from other manufacturers. The BEMS from smaller vendors are usually used to integrate HVAC equipment from the same vendor.

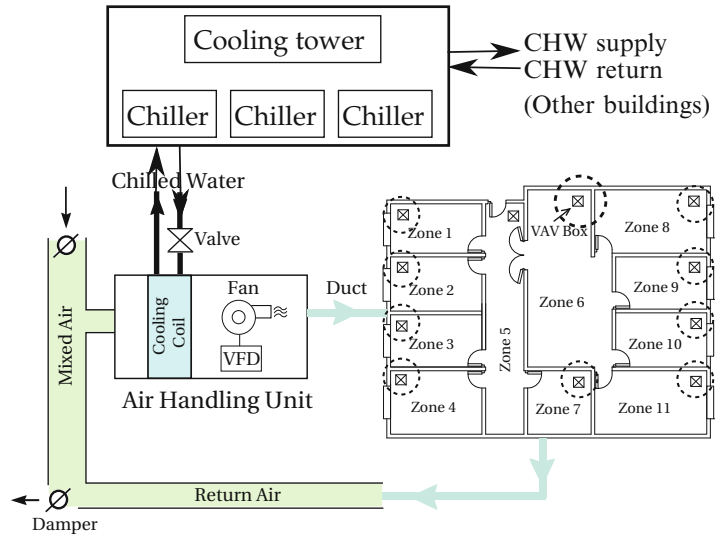
At present, commercially available BEMS are mostly used to perform set point tracking control. The set points are specified by human operators through a user interface. The default values of these set points are usually chosen during building commissioning. Some of these set points are informed by decades of engineering practice and field studies. For instance, the conditioned air temperature (downstream of the cooling coil) is frequently chosen to be 55°F in hot humid climates (Williams 2013).

Even the simplest set point control loops are hybrid controllers, employing a mix of logic loops and PI controllers. For example, a commonly used zone VAV box control algorithm is the so-called single maximum control. Depending on the current temperature of the zone and its past history, the controller switches to one of the three modes: cooling, heating, and deadband. In each mode, there is a distinct set point for the zone temperature, and a PI controller is used

Building Energy

Management System,

Fig. 1 A hydronic HVAC system with an air handling unit (AHU) and multiple zones. A central chiller plant supplies chilled water (CHW) to multiple buildings



to manipulate the flow rate and the amount of reheating (except in the cooling mode in which reheating is turned off) to maintain that set point. These set point tracking control algorithms typically come prepackaged in BEMS, and modifications to their programming are performed during installation and commissioning. A programming interface is offered as part of the BEMS that allows automatic changes to the set points. The degree of flexibility of these programming interfaces is typically limited, so usually only simple rule-based logics can be implemented. For instance, the Siemens offers a programming interface using a proprietary language called *ppc1*, which is reminiscent of BASIC, with features such as GOTO statements that are deprecated in modern programming languages.

Communication in BEMS

There are multiple levels of communication among devices and sub-networks in a BEMS. A simple classification employs three layers: floor level network layer, management layer, and enterprise layer. BACnet is a communication protocol that runs on top of other existing electrical standards like RS-485 (serial communication), Ethernet, and MS/TP (Master slave/token passing). LonWorks was engineered to be both a data protocol and an electrical standard for digital communications. In the USA, BACnet is more

widely used compared to Modbus and LON. In modern buildings, “BACnet/IP over Ethernet” is probably the most relevant. That essentially means that the BEMS uses BACnet/IP protocol for communication among devices and BACnet packets are carried over Ethernet cables.

Interoperability is still an issue in BEMS even though BACnet was devised to resolve interoperability. A reason for this issue is that BACnet is a standard and not all vendors implement it the same way. To ensure the quality of BACnet implementation, one can apply for a BTL license, but most vendors do not. In recent years the NiagaraTM framework has provided a way to integrate multitude of devices from multiple manufacturers using diverse protocols.

Optimization-Based Control of Building Energy Systems

Opportunities

There is a large untapped opportunity to improve energy efficiency and indoor climate through advanced decision-making. This opportunity comes from gap between what the current BEMS are capable of and what they are being used for. Modern buildings equipped with BEMS typically have sufficient actuation capabilities at many levels (chillers, AHUs, and zone VAV

boxes) that can be controlled to deliver high performance in terms of energy efficiency and indoor climate. Against this possibility, the reality at present is that BEMS are used to maintain constant set points which are designed based on steady-state considerations. Thus, the simplest way to improve HVAC operational performance is to change the set points in real time by solving an optimization problem that takes into account the difference between the design conditions and actual conditions. Lower-level control algorithms that BEMS are already equipped with can then be tasked with maintaining these set points. This optimization problem has to be revisited periodically as new information – measurements and forecasts – becomes available. For this reason, there is an emerging consensus that model predictive control (MPC) – that repeatedly solves an optimization problem posed over a receding horizon – is appropriate for achieving high performance from existing BEMS. Another option is a model-free approach such as reinforcement learning, though little exploration has been performed into that frontier.

As in any control systems, sensing and actuation play as big a role as control algorithms. The control problem – whether one employs MPC or some other approach – can be made easier by adding more sensing and actuation. Additional actuation is quite expensive since that requires changing the building's physical structure. Adding sensors to a BEMS after it has been installed and commissioned is far more expensive than the cost of the sensors themselves due to the cost of integration and testing. Still, adding sensing is far less expensive than adding actuators. Adding advanced decision-making is perhaps the least expensive, especially if the computations are performed in the cloud, while the BEMS serves only to provide the sensor data and execute the decisions computed by the cloud-based algorithm through the building's actuators. This requires a middleware. BACnet, for instance, allows multiple commands to be sent to a controller board with different priorities, and the equipment controller implements the one with the highest priority. This mechanism can be

used by the middleware to overwrite the default control commands computed by the BAS and replace them by the commands from the cloud-based algorithm.

We next discuss some of the opportunities of achieving high-performance building operation using MPC and the challenges therein. Although the right metric for performance will vary from one building to another depending on the preference of the building owner, operator, and its occupants, it is common in the research literature to take energy use (or energy cost) as the objective function to minimize, while indoor climate requirements are posed as constraints. The key energy consumers are the chillers that produce chilled water, reheat coils, and supply air fans in air handling units (AHUs) and finally reheat coils in zone-level VAV boxes (see Fig. 1). The control problem therefore has to consider decisions for these equipment and the downstream effect of these decisions on the indoor climate. The energy consumed by pumps for chilled and hot water are ignored here.

“Air-Side” Optimal Control

In the so-called air-side optimization problem, the control commands are set points of air handling units and perhaps zone-level VAV boxes. Chiller operation is outside the purview of the decision-making problem and will be discussed in section “[The “Water-Side” Optimal Control](#)”.

The optimization problem underlying an MPC controller will seek to minimize some objective function subject to the dynamic constraints and actuator limits. In a discrete-time setting, at time index k , the MPC controller computes the decisions $u_k, u_{k+1}, \dots, u_{k+N-1}$ over a planning horizon $\mathcal{K}_k = \{k, k+1, \dots, k+N-1\}$ of length N by solving an optimization problem of finite (N -length) horizon. The planning horizon depends on many factors. If the objective is to minimize energy cost, and monthly peak demand plays an important role in the energy cost, the planning horizon needs to span months. A day-long planning horizon seems to be the shortest possible while keeping the problem practically relevant.

The Single-Zone Case

We describe the problem in detail for a building in which an AHU is used to maintain the climate of one space, which we refer to as a “single-zone” building. In such a building, shown in Fig. 2, only four variables can be independently varied, which form the control command u : (i) $\dot{m}^{SA}$, the flow rate (kg/s) of *supply air*, (ii) r^{OA} , the *outdoor air ratio*, i.e., the ratio of outdoor air to supply air flow rates, $r^{OA} := \dot{m}^{OA} / (\dot{m}^{OA} + \dot{m}^{RA}) \in [0, 1]$, (iii) T^{CA} , the temperature of *conditioned air*, and (iv) q^{rh} , the rate of reheating (kW). Thus, $u_t = [\dot{m}^{SA}, r^{OA}, T^{CA}, q^{rh}]_t^T \in \mathbb{R}^4$. Each of these four control commands are in fact set points of lower-level control loops that are common in existing building control systems.

There are many choices of control that can lead to similar indoor climate but distinct energy consumption. A small T^{CA} with small $\dot{m}^{SA}$ can deliver the same “cooling” as a slightly larger T^{CA} and larger $\dot{m}^{SA}$. While lower $\dot{m}^{SA}$ uses less energy consumption, lower T^{CA} causes more energy consumption since it removes more moisture, which requires removing the large latent heat of evaporation. It is important to emphasize that the conditioned air temperature and humidity T^{CA}, W^{CA} cannot be decided independently, only T^{CA} is a part of the control command since it can be maintained by a chilled water control loop. The humidity of the conditioned air is indirectly decided by T^{CA} due to the dynamics

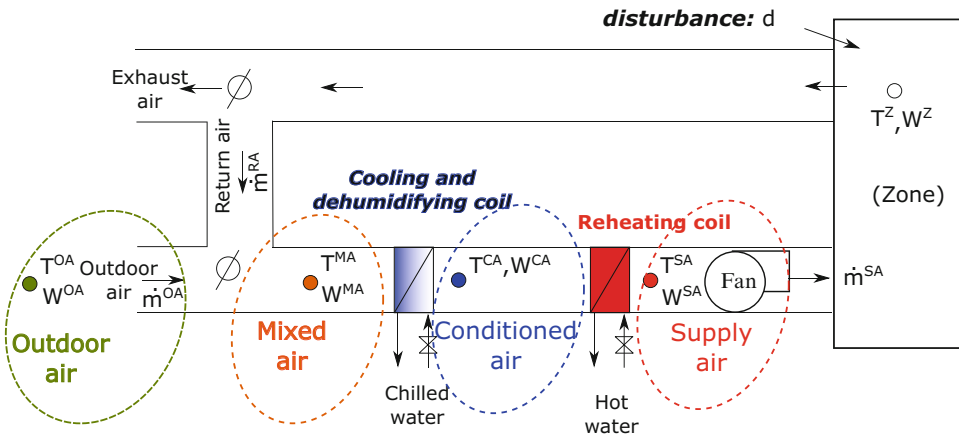
of the cooling coil. The relationship is highly complex and challenging to model for real-time optimization (Raman et al. 2019).

The relationship between the control command u and the disturbance d to indoor climate variables is best expressed as process (dynamic) models. They form the equality constraints in the optimization. Inequality constraints come from actuator limits and bounds on climate variables (state constraints). Apart from the dynamic constraints, there are two other types of constraints. The first set of constraints comes from thermal comfort and indoor air quality considerations. The second set comes from actuator limits.

For the purpose of exposition, we use the total HVAC energy (over a time interval N) as the objective function in the MPC optimizer: $J = E_{tot} = \sum_{t=k}^{k+N-1} (p_t^{cd} + p_t^{rh} + p_t^{fan}) \Delta t = \sum_{t \in \mathcal{T}_k} p_t^{tot} \Delta t$, though other choices are possible, such as the monthly energy cost that depends on total energy use and sometimes on a combination of energy use and peak demand during the month. In summary, the optimization problem within the MPC controller at time k is:

$$\begin{aligned} u_k^* &= \arg \min_{u_k, x_k} J(u_k, x_k, \hat{d}_k), \text{ s. t. } x_{k+1} \\ &= f(x_k, u_k, \hat{d}_k), x_k \in \mathcal{X}_k, x_k \in \mathcal{U}_k \end{aligned} \quad (1)$$

where $u_k := (u_k, \dots, u_{k+N-1})$ and $x_k := (x_k, \dots, x_{k+N-1})$ are the inputs and states, and



Building Energy Management System, Fig. 2 A single zone VAV HVAC system

$\hat{\mathbf{d}}_k = (\hat{d}_k, \dots, \hat{d}_{k+N-1})$, in which $\hat{d}$ is prediction of disturbance d , and $\mathcal{X}_k, \mathcal{U}_k$ are constraint sets for states and inputs.

Climate controllers currently used in buildings do not optimize; they err on the side of maintaining indoor climate since that is far more important than the energy bill (Tom 2008). Typically, T^{CA} is maintained by a PID loop at a set point that is decided based on decades of engineering experience. For instance, most hot and humid climates T^{CA} is set to 55°F (Williams 2013). The flow rate $\dot{m}^{\text{SA}}$ and reheating rate q^{rh} are decided by feedback controllers to ensure space temperature is within predetermined bounds; the bounds are again decided based on decades of research on human comfort (Baughman and Arens 1996; American Society of Heating, Refrigerating and Air-Conditioning Engineers 2010). Finally, the outdoor air ventilation is usually maintained at a set point based on design occupancy, which fixes the fourth component of $\mathbf{u}$, namely, r^{OA} .

The disturbance and its predictions play a crucial role in predictive control. The disturbance d consists of five components: (i) weather variables: solar heat gain η^{sun} , OA temperature T^{OA} , and OA humidity W^{OA} , and (ii) internal signals: sensible heat gain q^{int} (mostly from occupants and their actions, such as use of computers and lights), internal moisture generation rate $\dot{m}_{\text{H}_2\text{O}}$ (from occupants' bodies, decorative plants, coffee machines, etc.), and the number of occupants o . At every instant k , the optimizer needs predictions of all the exogenous signals for the planning horizon. Except for weather-related variables, obtaining forecasts of the remaining disturbance signals is a highly challenging problem.

Multi-zone Case

In most buildings an AHU is used to deliver air to multiple zones (see Fig. 1). The problem described above can be expanded to the multiple-zone case in a straightforward manner. The state dynamics will involve the states of thermal dynamics from each zone, and the control command will now include not simple AHU-level variables but also the set points for each of the VAVs. The problem is considerably more

challenging and not simply due to the higher computational complexity caused by the higher state dimension. Additional challenges come from the higher degree of uncertainty in models.

The “Water-Side” Optimal Control

The so-called water-side control problem is to make decisions about the chiller and cooling towers. Most chillers at present are run by constant-speed motors, so the key decision is turn a chiller in a bank of chillers either on or off. Even in constant speed chillers, the load on the chiller can be changed by actuating the inlet guide vanes. In some chiller plants, the supply water temperature can be manipulated and that becomes another decision variable. The water side problem is applicable only with buildings with chillers, which is more common in a campus or district setting (Patel et al. 2018).

Challenges

There are many challenges in successfully applying MPC to HVAC systems. One is the need for an accurate and yet computation-friendly dynamic model. Although there is a long history of modeling HVAC systems and equipment, there are still unresolved issues. The underlying physical processes involve conductive, convective, and radiative heat transfer as well as mixing and transport of several species such as moisture and CO_2 . Thus, first principles-based models can be arbitrarily complex. A popular class of models for modeling temperature evolution in a building is based on the resistance-capacitance networks, whose parameters are fitted to measured temperature data. There is a large unknown disturbance that comes from heat gains from occupants and their activities that makes system identification challenging (Coffman and Barooah 2018). Moreover, these models ignore humidity; work on identification of models that include both temperature and humidity is rare. Constructing control-oriented models for HVAC equipment, such as the cooling and dehumidification coil, is more challenging (Raman et al. 2019).

Another challenge for MPC-type control is the need for prediction of the disturbance signal over the planning horizon. Recall that there are many components in the disturbance signal, and while weather-related disturbances can be reasonably forecasted, signals such as number of occupants and the moisture generation are difficult to predict.

Yet another challenge is addressing the high computational complexity due to the large number of decision variables, especially where there are a large number of zones and the planning horizon is long.

Building-Specific Requirements

There are many types of buildings, and all have their own unique requirements on indoor climate which spill over to the constraints on the climate control system, MPC or not. Healthcare-related facilities in particular require special considerations and have distinct environmental constraints. The HVAC system described in detail in section “[“Air-Side” Optimal Control](#)” is a hydronic (chilled water) system. The details of the control problem will differ in case of a DX cooling system, which are used widely in packaged rooftop units used in small commercial buildings.

The air-side control problem discussed in section “[Optimization-Based Control of Building Energy Systems](#)” involved continuously variable actuators and set points, and the optimization becomes a nonlinear program (NLP). However, in case of a DX system, the decision variables may be integer valued if compressor stages need to be decided. In fact, many large HVAC equipment are ultimately on-off, such as compressors in chilled water plants, cooling towers with constant speed fans, etc. The optimization problem therefore becomes a mixed integer nonlinear programs (MINLP), which are considerably more challenging to solve than NLPs. Advances in solving MINLPs will thus benefit adoption of MPC in buildings.

Two other applications in which optimal control of BEMS can play an important role are demand side services and management of on-site renewable energy sources. More generally, in any building that moves away from the traditional

role of being purely a consumer of energy that is supplied by someone else (power grid, gas grid, etc.) to one of a *prosumer* that uses on-site generation, and perhaps energy storage, can benefit significantly from more intelligent real-time decision-making than the currently available rule-based control algorithms of existing BEMS.

Summary and Future Directions

BEMS augmented with advanced decision-making algorithms can improve both indoor climate and reduce energy use. In addition, they can help operate buildings with on-site generation and storage resources. There are many challenges in achieving this vision. Control-oriented models learned automatically from data, prediction of exogenous disturbances, and optimization algorithms all need advances. Another option is model-free (learning-based) control. No matter the approach, the resulting methods need to be inexpensive to deploy and maintain, which is challenging due to differences among buildings and changes that occur in a building over time. The systems and control community is well positioned to address many of these challenges.

Cross-References

- ▶ [Building Comfort and Environmental Control](#)
- ▶ [Building Control Systems](#)
- ▶ [Model Predictive Control in Practice](#)
- ▶ [Nominal Model-Predictive Control](#)

Bibliography

- American Society of Heating, Refrigerating and Air-Conditioning Engineers (2017) The ASHRAE handbook fundamentals (SI Edition)
- American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc (2010) ANSI/ASHRAE standard 55-2010, thermal environmental conditions for human occupancy. www.ashrae.org
- Baughman A, Arens EA (1996) Indoor humidity and human health—Part I: literature review of health effects of humidity-influenced indoor pollutants. ASHRAE Trans 102 Part1:192–211
- Braun J (1990) Reducing energy costs and peak electrical demand through optimal control of building thermal storage. ASHRAE Trans 96:876–888

- Coffman A, Barooah P (2018) Simultaneous identification of dynamic model and occupant-induced disturbance for commercial buildings. *Build Environ* 128:153–160
- Patel NR, Risbeck MJ, Rawlings JB, Maravelias CT, Wenzel MJ, Turney RD (2018) A case study of economic optimization of HVAC systems based on the Stanford University campus airside and waterside systems. In: 5th international high performance buildings conference
- Raman NS, Devaprasad K, Barooah P (2019) MPC-based building climate controller incorporating humidity. In: American control conference, pp 253–260
- Tom S (2008) Managing energy and comfort: don't sacrifice comfort when managing energy. *ASHRAE J* 50(6):18–26
- Williams J (2013) Why is the supply air temperature 55F? <http://www.8760engineering.com/blog/why-is-the-supply-air-temperature-55f/>. Last Accessed 02 Oct 2017

Building Fault Detection and Diagnostics

Jin Wen¹, Yimin Chen², and Adam Regnier³

¹Department of Civil, Architectural, and Environmental Engineering, Drexel University, Philadelphia, PA, USA

²Building Technology and Urban Systems Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA

³Kinetic Buildings, Philadelphia, PA, USA

Abstract

Malfunctioning control, operation, and building equipment, such as unstable control loop, biased sensor, and stuck outdoor air damper, are considered as the top cause for “deficient” building systems. In this article, types of faults that are commonly observed in a building system are introduced firstly. Literature-reported fault detection and diagnosis methods for building systems are then briefly summarized, which is followed by a discussion on the technical challenges in building system fault detection and diagnosis areas. Other topics, such as impacts that a fault could cause to a building, how to evaluate a fault detection and diagnosis method, and issues related to

implementing a fault testing, are reviewed next. Future directions in this area are presented at the end of this article.

Keywords

FDD · Fault impact · FDD evaluation · FDD challenges

Introduction

Malfunctioning control, operation, and building equipment, such as unstable control loop, biased sensor, and stuck outdoor air damper, are considered as the top cause for “deficient” building systems. These dramatically increase the building energy consumption (estimated that between 0.35 and 17 quads of additional energy consumption caused by key faults at a national level (Roth et al. 2004), and significantly impact the health/productivity of occupants. Field studies have shown that an energy saving of 5–30% and improved indoor air quality can be achieved by simply applying automated fault detection and diagnosis (AFDD) followed by corrections even if these are not done in real time (Katipamula and Brambley 2005a).

In Haves (1999), fault detection is defined as “determination that the operation of the building is incorrect or unacceptable in some respect” and fault diagnosis is defined as “identification or localization of the cause of faulty operation.” In order to perform fault detection, a “correct” or “acceptable” operation and system status typically needs to be established first. This status is referred to as “baseline” and/or “fault free” status, although a true “fault-free” status does not really exist in real building systems. For example, a common economizer fault is that an outdoor air damper is stuck at a fixed position. To detect this fault, the “correct” outdoor air damper position, i.e., the baseline or fault-free status, needs to be established first. A fault detection could be performed at a component (such as a damper) level, subsystem (such as an economizer) level, system (such as an air handling unit) level, or whole building level. The output of a fault detection is an alarm or report that indicates the monitored

object (component, subsystem, system, or whole building) is “faulty,” i.e., performs abnormally from its baseline/fault-free status.

Compared to fault detection, fault diagnosis is much more difficult because more knowledge, information, and analysis are needed to isolate the root causes that impact the system performance. For example, a fault detection process might detect an air handling unit to be “faulty.” However, what exactly causes this system to be faulty needs to be identified in a fault diagnosis process.

It is noticed that existing literature does not clearly separate the concepts of AFDD and fault detection and diagnosis (FDD). A reason for this could be due to the challenge to define the concept of “automated,” as many AFDD strategies may still engage some manual efforts. Hence these two concepts are not strictly separated in this article.

Types of Faults

There are many ways to categorize faults in a building. Faults can be categorized by the location of the malfunctioned (faulty) component. Such as envelope fault, air handling unit fault, chiller fault, etc. Faults can also be categorized by the types of component that a fault affects, such as hardware fault and software fault. Hardware faults can further be divided into sensor fault, controlled device (damper, valve, etc.) fault, and equipment (fan and coil, etc.) fault. Software faults can further be divided into operator fault (i.e., forgot to remove a software override command) and control fault (unstable control, sluggish control, wrongly programmed control strategy, etc.). Faults can be categorized by how a fault occurs and progresses, such as abrupt faults (occur suddenly and do not progress further) and degrading faults (occur slowly and progress overtime). Examples of abrupt faults include damper or valve stuck fault, fan failure fault, software override fault, etc. Examples of degrading faults include sensor bias fault, in which the bias often changes over time, coil fouling fault, and fan efficiency degradation

fault. Depending on how many subsystem a fault impacts, faults can be categorized as component (or local) faults or whole building faults. The latter refers to those faults that occur in one subsystem but affect more than one subsystem. For example, a chilled water supply temperature sensor bias fault affects operation of both the chiller and the air handling units that the chiller serves.

FDD Methods

Many methods have been developed for component level and whole building level AFDD for building systems. In Katipamula and Brambley (2005a, b) and Kim and Katipamula (2018), Katipamula, Brambley, and Kim provided a comprehensive classification for methods used for HVAC system AFDD as shown in Fig. 1.

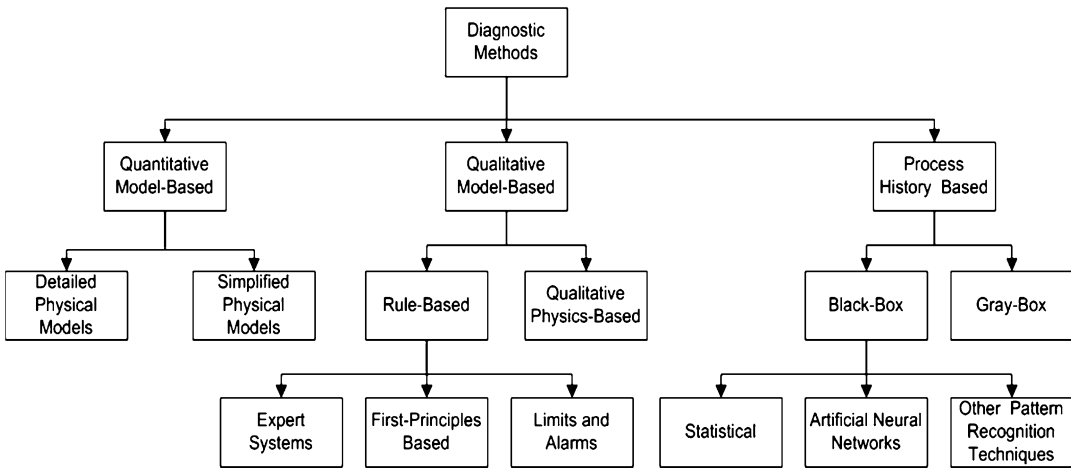
These methods have been reported to be used for fault detection alone without isolating the root cause or for a combination of both fault detection and fault isolation.

Some new studies that are not discussed in Katipamula and Brambley (2005a, b) and Kim and Katipamula (2018) have reported successful application of Bayesian network (BN) for fault diagnosis, including its application for chiller fault diagnosis (Zhao et al. 2013), air handling unit fault diagnosis (Zhao et al. 2015), and whole building fault diagnosis. The use of Bayesian networks (BN) seeks to combine the aspects of both rule-based and data-driven approaches with many of the benefits of a probabilistic data-driven approach combined with the benefits of a rule-based approach in a single algorithm.

Technical Challenges

Many technical challenges still exist when developing AFDD tools for building systems, which often make reducing the product cost a difficulty:

1. Challenge of limited measurements and poor measurement quality: Measurements used for



Building Fault Detection and Diagnostics, Fig. 1 Classification scheme for FDD methods (Katipamula and Brambley 2005a)

- building AFDD tools are most commonly obtained from a building automation system (BAS). Yet the sensor system in a BAS is designed for building control purpose, not for AFDD purpose. There is a lack of redundant measurements to increase measurement reliability. As a result, sensor faults have strong impact on overall AFDD accuracy and false alarm rate.
2. Challenge of building measurement taxonomy: A lack of standard measurement and data taxonomy has been a well-recognized challenge for building AFDD tools. Engineers and researchers often have to spend significant time to understand the physical meaning of point (measurement) names and their hierarchy and to correlate these measurements with AFDD strategies.
 3. Challenge of qualitative and quantitative model-based methods: Developing and calibrating physics-based models to reach an accuracy level that is sufficient for building AFDD, especially for whole building AFDD, remains to be challenging and requires expensive manual inputs. At the same time, qualitative and quantitatively model-based tools often suffer from low scalability when being applied for different buildings, due to the needs to customize physics-based models, rules, and thresholds
 4. Challenge of process history-based method: Process history-based methods have good scalability and low implementation cost. However, these methods rely on data. Hence data quality strongly affects the performance of these AFDD tools. As mentioned in Challenge 1, building data quality is much worse than those from other applications, such as process line. In a real building, missing data and a lack of trended data are common, which further complicate the application of process history-based methods. Moreover, for whole building AFDD tools that need to utilize data from the entire building, which often has hundreds, if not thousands of data points, how to handle such an increased data dimensionality in a computationally efficient way, also referred to as the curse of the data dimensionality, is a challenge.
 5. Challenge of generating baseline data: For building systems, it is often challenging to differentiate weather and/or internal load triggered system operational changes from fault triggered abnormalities. Difficulties exist as to how to collect historical baseline data, and how to efficiently select among these collected historical baseline data, those baseline data that are under similar weather and internal load conditions as the current data from system operation. Notice that a real building sys-

tem is often impossible to be at a true “fault-free” status. How to define baseline status and how to define threshold (difference between fault-free and faulty statuses) are key challenges for building AFDD tools.

6. Challenge of fault diagnosis and multiple faults: Many methods have been reported for fault detection, i.e., identifying abnormalities. Yet there is a lack of studies that focus on locating and identifying root causes, especially for those faults that cause abnormalities in multiple subsystems/components, i.e., have coupled abnormalities. For example, if the supply air temperature sensor in an air handling unit has a positive bias, i.e., screen reading is higher than its real value, the real supply air temperature would be lower than its set point. Hence the abnormalities could include increased cooling coil valve, abnormal chilled water loop pressure/flow rate, and reduced downstream variable air volume unit damper positions. Moreover, in a real building system, it is common that multiple faults exist. How to isolate multiple faults (Li and Braun 2007) has been rarely reported in the open literature.
7. Challenge of FDD method evaluation: Literature-reported FDD methods are mostly developed and evaluated using simulated system data. This is due to the difficulties of obtaining and analyzing real building data. Implementing faults and obtaining data that contain fault impacts in real buildings are already challenging. Yet cleaning and analyzing building data to obtain “ground truth” is even more arduous since unexpected naturally occurring faults could exist in the system and cause abnormalities or complicate (sometimes even eliminate) the fault impacts expected from the artificially implemented faults.

Fault Impacts

A fault in building systems affect the building’s energy consumption, indoor environmental quality, and systems’ lifespan. Notice that not all fault impacts are adverse. For example, if the

outdoor damper is stuck at a 100% open position, higher-than-normal ventilation air will be brought into the building. Hence this fault might have a positive impact on the indoor environmental quality. Of course, meanwhile, excessive energy might need to be spent to condition this additional ventilation air. Hence this fault has an adverse impact on building energy consumption. Vice versa, there are faults that have an adverse impact on indoor environment yet might save energy. Overall speaking, there is very limited information in the literature relating to the overall energy impacts and other impacts of faults on HVAC systems. Impacts of undetected faults are a function of multiple factors, such as:

- How to define “fault-free” baseline operation
- The probability of the fault occurring
- The probability of the fault going undetected without an effective AFDD procedure in place

Kim and Katipamula (2018) provide a summary of studies that report on fault impacts.

FDD Method Evaluation

Many market-available AFDD tools and literature-reported AFDD methods have been developed. However, there is no standard method to evaluate these tools/products and to report their performance, such as the diagnostic accuracy and/or cost effectiveness. The following gaps exist when reporting an AFDD tool/method:

1. Whether the tool requires the installation of additional sensors that are not typically found in existing systems. The installation of additional expensive sensors, like BTU/flow meters on the coil piping or additional air flow stations, is often prohibitive in the widespread adoption of a proposed AFDD tool.
2. Whether the tool requires costly engineering effort to be implemented in the field. Common causes for such additional engineering hours include fault data generation/collection, strategy and fault threshold customization for each installation, and modeling of the physical systems/components.

3. Whether the strategy is effective at maintaining its accuracy through multiple operational modes and transient states typically experienced during a system's normal operation. Many strategies that have been demonstrated to be effective in specific operational conditions are not adaptive for other operating conditions.
4. False alarm rates are rarely reported in the literature. Yet this has been widely identified by the industry as a key factor delaying widespread commercial adoption of AFDD tools for building systems.
5. Whether the strategy is able to handle the typical real-world BAS data quality. Strategies that are only tested using simulated data might experience difficulties when applied to real buildings due to problems with typical BAS data (missing data, noise, sensor faults, sensor accuracy, etc.).
6. Whether the strategy can easily be applied in different buildings, i.e., scalability. Strategies that are developed for a specific building or a specific type of system might experience difficulties when applied to other buildings/systems.

Without a standardized evaluation method, it is very hard for building owners/operators and building service companies to select a proper AFDD strategy/product that suits their needs.

Several studies (such as in Yuill and Braun 2013) have reported on methodologies for comparing AFDD tools for some components such as rooftop unit (RTU). Currently, a standardized method of testing is being developed for RTU FDD methods at the American Society of Heating, Refrigerating, and Air-conditioning Engineers (SPC 207p). An ongoing project (Granderson 2018) being performed by the Lawrence Berkeley National Laboratory (LBNL) aims at developing more evaluation methodologies for other HVAC component AFDD.

Implementing Fault Testing

In order to collect data for AFDD method evaluation, faults often need to be artificially injected to

a real system. There are generally two methods to artificially implement a fault: via the BAS or via the hardware. For example, a damper stuck fault can be implemented via the hardware by disconnecting the motor from the existing BAS and connecting it with a constant input signal (e.g., by a voltage meter) to maintain the damper position. It can also be implemented via the BAS by software-overriding the control signal at a desired constant. However, when a fault is implemented via the BAS, collected BAS measurements need to be post-processed to imitate those from a real fault event. Using the same damper stuck example discussed above, if the fault is implemented by software-overriding the damper control signal, the real control signal generated from the control algorithm for outdoor air damper needs to be recorded and need to replace the software-overridden damper signal because if a real damper stuck fault occurs, the damper control signal would not be the constant of the stuck position. Understanding how a fault would affect the trended measurements is critical for a successful fault implementation. When implementing faults in a real building or real system, data need to be carefully and manually analyzed to identify the ground truth. In general, several scenarios could happen from the collected data: (1) no artificial fault is implemented but naturally occurring fault exist; (2) artificially implemented fault has caused system abnormality; (3) artificially implemented fault has not caused any measurable system abnormality under the weather and operation conditions tested; and (4) both artificially implemented fault and naturally occurring faults exist and have caused abnormality.

Limited data and test bed, such as that reported by Wen and Li 2012, exist in the open literature that can be used to test AFDD tools. The LBNL project mentioned above (Granderson 2018) is aiming at developing more data and test bed.

Summary and Future Directions

Faults in building systems strongly affect a building's energy consumption, indoor environment quality, and overall system lifespan. Many studies exist in the literature that focus

on AFDD method development for both HVAC component or whole building faults. Qualitative, quantitative, and process history-based methods have all been reported. However, many challenges have prevented a wide adoption of AFDD tools in the field. The primary drivers for growth of adoption of AFDD solutions include (1) reduction of implementation costs; (2) education of customers; and (3) industry standards, including standardized terminology and benchmarks or ratings that can be used for the evaluation of competing technologies. Innovations in machine learning and artificial intelligence have provided potential techniques to overcome these challenges. There is generally a lack of fault diagnosis method reported in the literature, especially for multiple simultaneous faults. Fault impact analysis, fault method evaluation, and real building fault testing are also lacking in the literature and need further development.

Cross-References

- [Building Comfort and Environmental Control](#)
- [Building Control Systems](#)
- [Building Energy Management System](#)
- [Fault Detection and Diagnosis](#)

Bibliography

- Granderson J (2018) Automated Fault Detection and Diagnostics (AFDD) performance testing. Available from: https://www.energy.gov/sites/prod/files/2018/05/f52/32601_Granderson_050218-835.pdf
- Haves P (1999) Overview of diagnostic methods. In: Proceedings of the workshop on diagnostics for commercial buildings: from research to practice
- Katipamula S, Brambley MR (2005a) Review article: methods for fault detection, diagnostics, and prognostics for building systems—a review, Part I. HVAC&R Res 11(1):3–25
- Katipamula S, Brambley MR (2005b) Review article: methods for fault detection, diagnostics, and prognostics for building systems—a review, part II. HVAC&R Res 11(2):169–187
- Kim W, Katipamula S (2018) A review of fault detection and diagnostics methods for building systems. Sci Technol Built Environ 24(1):3–21
- Li H, Braun JE (2007) A methodology for diagnosing multiple simultaneous faults in vapor-compression air conditioners. HVAC&R Res 13(2):369–395
- Roth KW et al (2004) The energy impact of faults in US commercial buildings
- Wen J, Li S (2012) RP-1312 – tools for evaluating fault detection and diagnostic methods for air-handling units ashrae Research Final Report (RP 1312), Atlanta, GA, USA
- Yuill DP, Braun JE (2013) Evaluating the performance of fault detection and diagnostics protocols applied to air-cooled unitary air-conditioning equipment. HVAC&R Res 19(7):882–891
- Zhao Y, Xiao F, Wang S (2013) An intelligent chiller fault detection and diagnosis methodology using Bayesian belief network. Energy Build. 57:278–288
- Zhao Y, Wen J, Wang S (2015) Diagnostic Bayesian networks for diagnosing air handling units faults – Part II: faults in coils and sensors. Appl Therm Eng 90: 145–157

Building Lighting Systems

Sandipan Mishra

Department of Mechanical, Aerospace, and Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract

Feedback control of lighting systems ranges from rule-driven on-off control to optimization based control algorithms. Simple lighting control algorithms use illumination and occupancy sensor measurements to dim lighting and harvest ambient lighting, whenever possible. For networks of lighting fixtures, illumination balancing algorithms use local information from sensors to achieve uniform lighting in a decentralized manner. Optimization-based algorithms are particularly useful for control of lighting fixtures with multi-spectral light sources. Such schemes are typically model based, i.e., they require knowledge of the light transport inside the illuminated space and rely on color quality indices and light source efficiencies to balance occupant comfort with energy consumption.

Keywords

Optimization-based control · Predictive control · Light transport · Distributed control · LED lighting

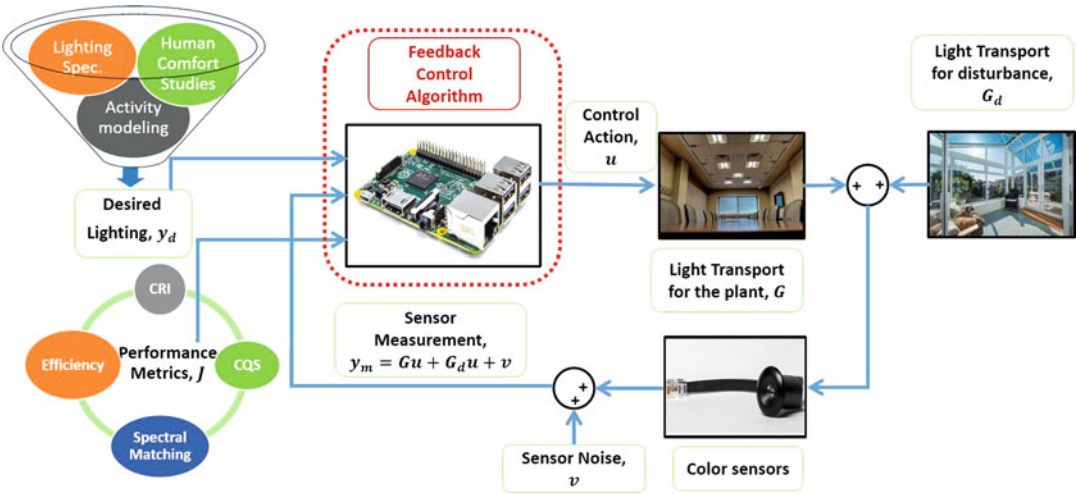
Introduction

Indoor illumination is a key component of a comfortable building interior and significantly affects health, productivity, mood, comfort, security, and safety of the occupants. In 2015, artificial indoor lighting in the residential and commercial sectors consumed nearly 404 billion kilowatt-hours (kWh) of electricity in the United States (Conti 2013). While traditional lighting systems were mainly designed to provide illumination alone, the new generation of lighting systems takes advantage of recent developments in solid state lighting and the advances in the state-of-the-art spectral and occupancy sensing technology to deliver functionality beyond just illumination. These lighting fixtures mix light from different light emitting diodes (LEDs) to create energy-efficient, high-quality, and healthy illumination. Lighting control systems use occupancy information and illumination measurements information to adjust LED intensities to produce comfortable illumination in an energy efficient way.

Although LEDs are more energy-efficient as compared to incandescent and fluorescent bulbs, feedback control of lights can enable further energy savings in two ways (1) occupancy-based lighting and (2) daylight harvesting. The former refers to the dimming of the light fixtures in unoccupied zones of the building, while the latter is based on harvesting ambient light from windows (or skylights) so that fixtures can be dimmed to save energy. Feedback control is particularly important for lighting systems because of unpredictable and time-varying nature of occupancy, naturally changing daylight conditions, and changes in light transport inside the illuminate space. In addition to energy savings, human comfort, mood, productivity, and health are also critical from a lighting standpoint. Most light control algorithms typically consider three lighting quality metrics: uniformity, brightness, and color quality. In general, lighting comfort depends on several parameters such as illumination, correlated color temperature (CCT), perception of color, and glare.

A schematic of a closed loop lighting control system is shown in Fig. 1. The “plant” is the indoor space being illuminated; the controlled inputs are the (multi- or single-channel) LED fixture commands; disturbances include the ambient daylight, while typical feedback is obtained

B



Building Lighting Systems, Fig. 1 Schematic of a typical lighting control system. (Figure taken from Imam et al. 2016)

from ambient illumination measurements or color sensor readings. Occupancy sensors are often used to detect the number and/or location of the occupants, and the color sensors measure the intensity and spectral content of the incident light. Occupancy information is used to determine the desired lighting for the space, while the color (or ambient brightness) measurements are used to maintain the desired illumination condition (y_d). The controller uses the lighting set-point values and the sensor measurements and generates the required input for each channel of the LED fixture(s) to achieve the control objective, which is based on a set of parameters determined by the building manager or user (e.g., energy cost, uniformity requirement, etc.). Performance metrics for the lighting system typically include energy consumption, uniformity, brightness, and color quality indices such as Color Rendering Index (CRI) or Color Quality Scale (CQS).

Light Transport Modeling

While high-fidelity ray tracing software such as Zemax may be used to directly simulate illumination of the indoor space, these models are not suitable for design of feedback control. We now present a brief description of the typical model for light transport in an indoor space used for designing feedback controllers.

Consider a space with n light fixtures, each of which contains p adjustable intensity channels (e.g., different color LEDs). Assume that the input to each fixture is represented as a vector, $u^i \in \mathbb{R}^p$, $i = 1, \dots, n$, and each entry in u^i is normalized to the range $[0, 1]$. If there are m locations of interest for the overall light field in the space, and there is a sensor with q channels of spectral sensing at each of these locations, then the output vector is a vector with qm elements, consisting of the different $y^j \in \mathbb{R}^q$, $j = 1, \dots, m$. The input-output model for this space can be given by:

$$y = Gu + w + v \quad (1)$$

where $y = [y^1, y^2, \dots, y^m]^T \in \mathbb{R}^{3m}$ is the output light measurement vector, $u = [u^1, u^2, \dots, u^n]^T \in \mathbb{R}^{pn}$ is the input light intensity control vector, $G \in \mathbb{R}^{3m \times pn}$ is the **Light Transport Matrix** (Foley et al. 1995) (LTM), $w \in \mathbb{R}^{3m}$ is the effect of ambient light on the sensors, and $v \in \mathbb{R}^{3m}$ is the measurement noise. The LTM is used to characterize the input-output relationship for the lighting system.

Feedback Control of Lighting

Lighting control methodologies can be broadly classified into three categories (1) logic-based controllers, (2) regulation-based controllers, and (3) optimization-based controllers. In logic-based methods, discrete logical statements and conditionals are used to determine the correct action for the lighting system in different situations based sensor measurements and a set of rules (Wen and Agogino 2010). Regulation-based control algorithms used feedback measurements to track predetermined reference value (Rubinstein 1984). Optimization-based algorithms are an alternative approach to controller design for closed loop lighting systems where the control problem is posed as an optimization problem and solved in real-time using optimization techniques (Afshari et al. 2014; Caicedo and Pandharipande 2016).

Optimization-based algorithms can be further classified into two sub-categories. In one-shot optimization algorithms, the control law is obtained by choosing the optimal solution at each time-step of the control loop. In iterative optimization algorithms, gradient-based methods are used to obtain the optimal direction for the evolution of the system at each time-step. The input for the next time-step is obtained by making an incremental change in the current input in the gradient direction. This process is repeated until the system converges to the global optimum of the cost function. While the one-shot optimization algorithms are deadbeat, they are typically more sensitive to noise and can significantly degrade in performance in the

presence of model uncertainty. We now describe five lighting control algorithms in greater detail.

Decentralized Integral Control

The simplest lighting control methodology that guarantees zero steady state error in the presence of sufficiently slowly varying disturbances is pure integral control:

$$u(k) = u(k-1) + \alpha (y_d(k-1) - y(k-1)) \quad (2)$$

where α is the controller gain, (y_d) is the desired, and (y) is the measured sensor measurements at k -th time-step. For decentralized implementation of this integral controller, the control law for i -th location is:

$$u_i(k) = u_i(k-1) + \alpha (y_d(k-1) - y_i(k-1)) \quad (3)$$

where α is a design parameter that controls the speed of convergence and $y_d(k-1)$ and $y_i(k-1)$ are, respectively, the desired and measured light levels at i -th location at the $(k-1)$ -th time-step. This control law is termed Decentralized Integral Controller (DIC).

Illumination Balancing Algorithm

For lighting systems with collocated light sources and sensors, cross-illumination between the different fixtures can cause nonuniform illumination if a simple DIC control law is used. In Koroglu and Passino (2014), Koroglu and Passino addressed this issue by proposing a distributed control algorithm called Illumination Balancing Algorithm (IBA), which uses the illumination levels of the neighboring fixtures to generate the control input for each fixture.

The control objective of IBA is Koroglu and Passino (2014) to track a desired light level while ensuring uniform illumination across all the zones of the work space, with communication among the light fixtures being restricted within a neighborhood. The work space is divided into pre-defined zones, where each zone consists of a monochromatic (white channel only) light source and a Light Dependent Resistor (LDR) is the light sensor. An arbitrary zone l is chosen as the

“leader.” This leader is the only fixture that has knowledge of the *desired* illumination level. For this fixture, illumination balancing and integral control are simultaneously used, as described in the first equation of (4). The rest of the zones only balance their illumination levels with their pre-defined neighbors (described in the second equation of (4)). Without explicitly knowing the target illumination value, these zones eventually track the desired illumination level by following the leader zone l . The control law is given by:

$$\begin{aligned} u_l(k) &= u_l(k-1) + \alpha e_l(k-1) \\ &\quad - \sum_{j \in N(l)} \gamma (y_l(k-1) - y_j(k-1)) \\ u_i(k) &= u_i(k-1) \\ &\quad - \sum_{j \in N(i)} \gamma (y_i(k-1) - y_j(k-1)) \quad i \neq l. \end{aligned} \quad (4)$$

where $N(i)$ represents the set of neighboring zones adjacent to the i -th zone and γ is a scaling factor that determines the stability margin and convergence speed. We refer the reader to Koroglu and Passino (2014) for a detailed development.

Daylight and Occupancy Adaptive Lighting Control

Unlike the distributed approach taken in IBA, a centralized control approach can be adopted for lighting control, as in Caicedo et al. (2010). In this algorithm, a centralized control law was designed to optimize power efficiency through daylight harvesting and occupancy adaptation. The objective was to minimize the total power consumption of the system while ensuring a variable level of minimum illuminance at each sensor and keeping the input intensities of the neighboring light fixtures close together. The desired minimum level of illuminance in a zone depends on its occupancy.

Light fixtures typically use pulse width modulation (PWM) for dimming, for which the power consumption by each light channel is proportional to its input intensity. Based on this fact,

the total power consumption of the system as the summation of the input intensities of the light fixtures at any given time k is $\sum_{i=1}^n \frac{u_i(k)}{\eta_i}$, where η_i is the efficiency of the i th fixture (channel). The overall optimization problem can be formulated as follows:

$$\Delta \mathbf{u}(k) = \underset{\Delta \mathbf{u}}{\operatorname{argmin}} \sum_{i=1}^n \frac{u_i(k-1) + \Delta u_i(k)}{\eta_i} \quad (5)$$

subject to the following constraints:

$$\begin{aligned} \mathbf{G} \Delta \mathbf{u}(k) &\geq \mathbf{y}_d^x(k) - \mathbf{y}(k-1), \\ -\Delta \mathbf{u}(k-1) &\leq \Delta \mathbf{u}(k) \leq \mathbf{1} - \Delta \mathbf{u}(k-1), \\ |\Delta u_i(k) - \Delta u_j(k)| &\leq \delta u - \Delta u_i(k-1) \\ &\quad + \Delta u_j(k-1), \end{aligned}$$

$$\text{where } i=1, \dots, n \text{ and } \forall j \in N(i)$$

where $\Delta \mathbf{u}(k) = [\Delta u_1(k), \dots, \Delta u_n(k)]^T$ is the incremental step vector for input intensity, $\mathbf{y}(k-1)$ is the sensor measurement, $\mathbf{y}_d^x(k)$ is the desired illuminance based on the occupancy state, x represents the occupancy state (0 and 1 mean unoccupied and occupied, respectively), δu is maximum allowed increment, $N(i)$ is the set of neighboring fixtures for the i -th fixture, and $\mathbf{G} \in \mathbb{R}^{qm \times pn}$ is the LTM.

The optimization problem with the (linear) objective function and constraints in (5) is solved using the interior point method in an iterative approach (Caicedo et al. 2010). The control update for time k is:

$$\mathbf{u}(k) = \mathbf{u}(k-1) + \alpha \Delta \mathbf{u}(k) \quad (6)$$

At each time instant k , the centralized controller solves the linear optimization problem in (5) to obtain the incremental step vector for the input intensity $\Delta \mathbf{u}(k)$ and uses the update equation (6) to obtain the complete input intensity vector $\mathbf{u}(k)$ for the next time-step. This algorithm assumes co-located light fixtures and sensors and an occupancy sensor for each zone. The design parameter α determines the speed of adaptation and can be adjusted to achieve smoother transition of the intensity levels.

Spectral Optimization for Polychromatic Lighting

The control strategies discussed in the previous sections were designed for single-color lighting systems. Aldrich et al. (2010) studied the problem of controlling color-tunable LED fixtures, which can generate a wide range of colors. As in Caicedo et al. (2010), the control design is posed as a constrained optimization problem. The objective is to minimize energy consumption or maximize the color rendering index (CRI) for a given user-specified color temperature (T_d).

The piecewise cost function includes three terms, as shown in (7), namely, (1) the 2-norm of the difference between the desired and measured color temperatures (T_d and T), (2) a nonlinear term ($\Delta uv(T)$) that quantifies the distance between the generated color and the black body curve, (The black body curve characterizes the set of all color points of the light radiated from a black body at different temperatures.) and (3) an additional term that is either the normalized power consumption or the normalized CRI of the generated light.

$$\min_u \quad \mathbf{J}(u) = \begin{cases} \left(\frac{T_d - T(u)}{T_d} \right)^2 + \alpha_1 \Delta uv(T(u))^2 + \alpha_2 \left(\frac{\Gamma_E u}{P_t} \right)^2, & \text{Power minimization} \\ \left(\frac{T_d - T(u)}{T_d} \right)^2 + \alpha_1 \Delta uv(T(u))^2 - \alpha_2 \left(\frac{R_d(u)}{100} \right)^2, & \text{CRI maximization} \end{cases} \quad (7)$$

$$\text{subject to} \quad y_d - y_{\text{dark}} \leq \mathbf{G} \mathbf{u} \leq (1 + \beta)(y_d - y_{\text{dark}}) \quad \text{and} \quad \mathbf{0} \leq \mathbf{u} \leq \mathbf{1}.$$

where $\mathbf{u} = [u_1, \dots, u_p]^T$ is the input intensity vector for the p LED channels, Γ_E is a row vector consisting of the power consumption of

each LED channel in the full-on state, P_t is the maximum power consumption for an individual light fixture, α_1 and α_2 are weights, while y_d

and y_{dark} are respectively the desired minimum illumination and ambient illumination measured at the sensor node position and $\mathbf{G} \in \mathbb{R}^{qm \times pn}$ is the LTM. The linear constraints ensure that the achieved illumination level on the work space falls within a certain boundary (defined by the parameter β) around the desired minimum illumination level y_d . An off-the-shelf solver was used to determine the optimum input intensity $\mathbf{u}^{opt}$ satisfying all the constraints specified in (7). This algorithm does not take the spatial variation of the light field into account and is suitable for a single light source and sensor.

Hierarchical Optimization for Multichannel LEDs

In many applications, the number of source channels is larger than the number of sensor channels (which are typically limited to RGB). Afshari and Mishra (2015) proposed an optimization-based framework for control of such systems, which exploits the inherent control redundancy due to the color metamerisms. The optimization problem is posed as determination of the appropriate light input to minimize the weighted sum of illumination quality and energy consumption.

Given the LTM, $\mathbf{G} \in \mathbb{R}^{qm \times pn}$, as described in section “[Light Transport Modeling](#)”, and a vector $y_d \in \mathbb{R}^{qm}$ consisting of the desired RGB values for each of the sensors, the light quality metric was formulated as the Euclidean norm of the difference between the desired and measured sensor values (y_d and y). On the other hand, the performance metric was formulated as a function of the input intensities (u) that might optimize different performance parameters, such as power efficiency, CRI, or CCT. The controller design problem is formulated as:

$$\begin{aligned} \mathbf{u}^{opt} = \underset{\mathbf{u}}{\operatorname{argmin}} \quad & \|y_d - y_k\|_2^2 + \alpha_f f(\mathbf{u}_k) \\ \text{subject to} \quad & 0 \leq u \leq 1 \end{aligned} \quad (8)$$

where $\mathbf{u} = [u_1, u_2, \dots, u_{pn}]^T$ consists of the input intensities of the light fixtures, $f(\mathbf{u})$ denotes a user-defined performance metric, and α_f is a weighting coefficient. u_k and y_k denote

the input intensity and RGB sensor measurement at k -th time-step, respectively.

The optimization problem in (8) can be solved using a *two-step hierarchical approach*. In the first step, a candidate solution ($\bar{u}_k$) to drive the output of the system toward the desired light field is calculated using a gradient based method. The update equation for this step is:

$$\bar{u}_k = u_k + \epsilon G^T (G G^T)^{-1} (y_d - y_k), \quad (9)$$

where ϵ is the step size. Since the system has more source channels than sensor channels, the redundancy in the generation of color results in the existence of an infinite number of candidate solutions, all resulting in the same sensor reading. Each of these solutions may successfully achieve the desired color set point, as long as y_d is a feasible target. This creates the opportunity for further optimization of the system performance within the set of all the candidate solutions.

In the next step, another optimization problem is solved to find the input intensity vector from the set of all candidate solutions that optimizes the user-defined performance metric. This optimal solution, u , is obtained using the following equation:

$$u_{k+1} = \bar{u}_k - G_N w \quad (10)$$

where G_N is an orthogonal matrix the columns of which span the null space of G and the w is obtained from solving the following optimization problem:

$$\min_w f(u - G_N w)$$

$$\text{subject to} \quad 0 \leq (u - G_N w)(i) \leq 1, \quad i=1, \dots, pn. \quad (11)$$

As an example of the user-defined performance metric $f(u)$, we can choose to optimize the total power consumption of the system. Assuming that the power consumption of individual LED channels of the fixtures can be approximated as a linear function of the channel's input intensity, $f(u) = \Gamma_E u$, where

Γ_E represents the power consumption of the individual LED channels in full-on state. The hierarchical approach results in convergence to the power-efficient solution that achieves y_d . Note that this approach ensures zero steady state error as well as power minimization, as opposed to trading off light quality for power saving.

Summary and Conclusion

This chapter provides an overview of various existing feedback control algorithms for LED-based lighting systems. In order to choose a particular feedback control strategy for smart lighting systems, some of the questions that need to be addressed are whether the lighting system is single-color or color-tunable, whether the multichannel color sensing and occupancy sensing technologies are available, whether the sensors are at the ceiling or at the work space in the illuminated space, what is the accuracy and speed of the sensors, etc. With advances in computation, sensing, and communication technologies, lighting systems may be flexibly adapted and reconfigured to specific requirements by switching between various methodologies. These advanced smart lighting systems, interconnected with other building systems, can in fact be used to provide new services for an office building beyond illumination.

Cross-References

- [Building Comfort and Environmental Control](#)
- [Building Control Systems](#)

- [Control of Circadian Rhythms and Related Processes](#)
- [Human-Building Interaction \(HBI\)](#)

Bibliography

- Afshari S, Mishra S (2015) An optimization framework for control of non-square smart lighting systems with saturation constraints. In: *Proceedings of American Control Conference*, pp 1665–1670
- Afshari S, Mishra S, Julius A, Lizarralde F, Wason J, Wen J (2014) Modeling and control of color tunable lighting systems. *Energy Build* 68:242–253
- Aldrich M, Zhao N, Paradiso J (2010) Energy efficient control of polychromatic solid state lighting using a sensor network. In: *Proceedings of SPIE optical engineering + applications conference*, pp 778408
- Caicedo D, Pandharipande A (2016) Daylight and occupancy adaptive lighting control system: an iterative optimization approach. *Light Res Technol* 48(6): 661–675
- Caicedo D, Pandharipande A, Leus G (2010) Occupancy-based illumination control of LED lighting systems. *Light Res Technol* 43(2):217–234
- Conti J (2013) Annual energy outlook 2013 with projections to 2040. U.S. Energy Information Administration, Washington, DC, Technical Report DOE/EIA-0383
- Foley J, van Dam A, Feiner S, Hughes J (1995) *Computer graphics: principles and practice in C*, 2nd edn. Addison-Wesley Professional, Boston
- Imam MT, Afshari S, Mishra S (2016) An experimental survey of feedback control methodologies for advanced lighting systems. *Energy Build* 130:600–612
- Koroglu M, Passino K (2014) Illumination balancing algorithm for smart lights. *IEEE Trans Control Syst Technol* 22(2):557–567
- Rubinstein F (1984) Photoelectric control of equi-illumination lighting systems. *Energy Build* 6(2): 141–150
- Wen Y, Agogino A (2010) Control of wireless-networked lighting in open-plan offices. *Light Res Technol* 43(2):235–248

Cascading Network Failure in Power Grid Blackouts

Ian Dobson
Iowa State University, Ames, IA, USA

Abstract

Cascading failure consists of complicated sequences of dependent failures and can cause large blackouts. The emerging risk analysis, simulation, and modeling of cascading blackouts are briefly surveyed, and key references are suggested.

Keywords

Outage · Branching process · Simulation · Dependent failures · Risk · Power law

Introduction

The main mechanism for the rare and costly widespread blackouts of bulk power transmission systems is cascading failure. Cascading failure can be defined as a sequence of dependent events that successively weaken the power system (IEEE PES CAMS Task Force on Cascading Failure 2008). The events and their dependencies are very varied and include outages or failures of many different parts of the power system and a whole range of possible physical, cyber, and

human interactions. The events and dependencies tend to be rare or complicated, since the common and straightforward failures tend to be already mitigated by engineering design and operating practice.

Examples of a small initial outage cascading into a complicated sequence of dependent outages are the August 10, 1996, blackout of the Northwest United States that disconnected power to about 7.5 million customers (Kosterev et al. 1999), the August 2003 blackout of about 50 million customers in Northeastern United States and Canada (US-Canada Power System Outage Task Force 2004), and the September 2011 San Diego blackout (Federal Energy Regulatory Commission 2012). Although such extreme events are rare, the direct costs can run to billions of dollars and the disruption to society is substantial. Large blackouts also have a strong effect on shaping the way power systems are regulated and the reputation of the power industry. Moreover, some blackouts involve social disruptions that can multiply the economic damage. The hardship to people and possible deaths underscore the engineer's responsibility to work to avoid blackouts.

It is useful when analyzing cascading failure to consider cascading events of all sizes, including the short cascades that do not lead to interruption of power to customers and cascades that involve events in other infrastructures, especially since loss of electricity can significantly impair other essential or economically important infrastructures. Note that in the context of

interacting infrastructures, the term “cascading failure” sometimes has the more restrictive definition of events cascading *between* infrastructures (Rinaldi et al. 2001).

Blackout Risk

Cascading failure is a sequence of dependent events that successively weaken the power system. At a given stage in the cascade, the previous events have weakened the power system so that further events are more likely. It is this dependence that makes the long series of cascading events that cause large blackouts likely enough to pose a substantial risk. (If the events were independent, then the probability of a large number of events would be the product of the small probabilities of individual events and would be vanishingly small.) The statistics for the distribution of sizes of blackouts have correspondingly “heavy tails” indicating that blackouts of all sizes, including large blackouts, can occur. Large blackouts are rare, but they are expected to happen occasionally, and they are not “perfect storms.”

In particular, it has been observed in several developed countries that the probability distribution of blackout size has an approximate power law dependence (Carreras et al. 2004b, 2016; Dobson et al. 2007; Hines et al. 2009). (The power law is of course limited in extent because every grid has a largest possible blackout in which the entire grid blacks out.) The power law region can be explained using ideas from complex systems theory. The main idea is that over the long term, the power grid reliability is shaped by the engineering responses to blackouts and the slow load growth and tends to evolve toward the power law distribution of blackout size (Dobson et al. 2007; Ren et al. 2008).

Blackout risk can be defined as the product of blackout probability and blackout cost. One simple assumption is that blackout cost is roughly proportional to blackout size, although larger blackouts seem to have costs (especially indirect costs) that increase faster than linearly. In the case of the power law dependence, the larger blackouts can become rarer at a similar

rate as costs increase, and then the risk of large blackouts is comparable to or even exceeding the risk of small blackouts (Carreras et al. 2016). Mitigation of blackout risk should consider both small and large blackouts, because mitigating the small blackouts that are easiest to analyze may inadvertently increase the risk of large blackouts (Newman et al. 2011).

Approaches to quantify blackout risk are challenging and emerging, but there are also valuable approaches to mitigating blackout risk that do not quantify the blackout risk. The n-1 criterion that requires the power system to survive any single component failure has the effect of reducing cascading failures. The individual mechanisms of dependence in cascades (overloads, protection failures, voltage collapse, transient stability, lack of situational awareness, human error, etc.) can be addressed individually by specialized analyses or simulations or by training and procedures. Credible initiating outages can be sampled and simulated, and those resulting in cascading can be mitigated (Hardiman et al. 2004). This can be thought of as a “defense in depth” approach in which mitigating a subset of credible contingencies is likely to mitigate other possible contingencies not studied. Moreover, when blackouts occur, a postmortem analysis of that particular sequence of events leads to lessons learned that can be implemented to mitigate the risk of some similar blackouts (US-Canada Power System Outage Task Force 2004).

Simulation and Models

There are many simulations of cascading blackouts using Monte Carlo and other methods, for example, Hardiman et al. (2004), Carreras et al. (2004a), Chen et al. (2005), Kirschen et al. (2004), Anghel et al. (2007), and Bienstock and Mattia (2007). All these simulations select and approximate a modest subset of the many physical and engineering mechanisms of cascading failure, such as line overloads, voltage collapse, and protection failures. In addition, operator actions or the effects of engineering the network may also be crudely represented. Some

of the simulations give a set of credible cascades, and others approximately estimate blackout risk.

Except for describing the initial outages, where there is a wealth of useful knowledge, much of standard risk analysis and modeling does not easily apply to cascading failure in power systems because of the variety of dependencies and mechanisms, the combinatorial explosion of rare possibilities, and the heavy-tailed probability distributions. However, progress has been made in probabilistic models of cascading (Chen et al. 2006; Dobson 2012; Rahnamay-Naeini et al. 2012).

The goal of high-level probabilistic models is to capture salient features of the cascade without detailed models of the interactions and dependencies. They provide insight into cascading failure data and simulations, and parameters of the high-level models can serve as metrics of cascading.

Branching process models are transient Markov probabilistic models in which, after some initial outages, the outages are produced in successive generations. Each outage in each generation (a “parent” outage) produces a probabilistic number of outages (“children” outages) in the next stage according to an offspring probability distribution. The children failures then become parents to produce the next generation and so on, until there is a generation with zero children and the cascade stops. As might be expected, a key parameter describing the cascading is its average propagation, which is the average number of children outages per parent outage. Branching processes have traditionally been applied to many cascading processes outside of risk analysis (Harris) and have been applied to estimate the distribution of the total number of outages from utility outage data (Dobson 2012). A probabilistic model that tracks the cascade as it progresses in time through lumped grid states is presented in Rahnamay-Naeini et al. (2012). The interactions between line outages are described in a probabilistic influence network different than the grid network in Hines et al. (2017).

There is an extensive complex networks literature on cascading in abstract networks that is largely motivated by idealized models of propagation of failures in the Internet. The way that

failures propagate only along the network links is not realistic for power systems, which satisfy Kirchhoff’s laws so that many types of failures propagate differently. For example, line overloads tend to propagate along cutsets of the network, and there are both short-range and long-range interactions between outages (Dobson et al. 2016; Hines et al. 2017). However, the high-level qualitative results of phase transitions in the complex networks have provided inspiration for similar effects to be discovered in power system models (Dobson et al. 2007).

Summary and Future Directions

One challenge for simulation is what selection of phenomena to model and in how much detail in order to get useful and validated engineering results (IEEE Working Group on Cascading Failures 2016). Faster simulations would help to ease the requirements of sampling appropriately from all the sources of uncertainty. Better metrics of cascading in addition to average propagation need to be developed and extracted from real and simulated data in order to better quantify and understand blackout risk. There are many new ideas emerging to analyze and simulate cascading failure, and the next step is to validate and improve these new approaches by comparing them with observed blackout data. Overall, there is an exciting challenge to build on the more deterministic approaches to mitigate cascading failure and find ways to more directly quantify and mitigate cascading blackout risk by coordinated analysis of real data, simulation, and probabilistic models.

Cross-References

- Hybrid Dynamical Systems, Feedback Control of
- Lyapunov Methods in Power System Stability
- Power System Voltage Stability
- Small Signal Stability in Electric Power Systems

Bibliography

- Anghel M, Werley KA, Motter AE (2007) Stochastic model for power grid dynamics. In: 40th Hawaii international conference on system sciences, Jan 2007
- Bienstock D, Mattia S (2007) Using mixed-integer programming to solve power grid blackout problems. *Discret Optim* 4(1):115–141
- Carreras BA, Lynch VE, Dobson I, Newman DE (2004a) Complex dynamics of blackouts in power transmission systems. *Chaos* 14(3):643–652
- Carreras BA, Newman DE, Dobson I, Poole AB (2004b) Evidence for self-organized criticality in a time series of electric power system blackouts. *IEEE Trans Circuits Syst Part I* 51(9):1733–1740
- Carreras BA, Newman DE, Dobson I (2016) North American blackout time series statistics and implications for blackout risk. *IEEE Trans Power Syst* 31(6):4406–4414
- Chen J, Thorp JS, Dobson I (2005) Cascading dynamics and mitigation assessment in power system disturbances via a hidden failure model. *Int J Elect Power Energy Syst* 27(4):318–326
- Chen Q, Jiang C, Qiu W, McCalley JD (2006) Probability models for estimating the probabilities of cascading outages in high-voltage transmission network. *IEEE Trans Power Syst* 21(3):1423–1431
- Dobson I (2012) Estimating the propagation and extent of cascading line outages from utility data with a branching process. *IEEE Trans Power Syst* 27(4):2146–2155
- Dobson I, Carreras BA, Newman DE (2005) A loading-dependent model of probabilistic cascading failure. *Probab Eng Inf Sci* 19(1):15–32
- Dobson I, Carreras BA, Lynch VE, Newman DE (2007) Complex systems analysis of series of blackouts: cascading failure, critical points, and self-organization. *Chaos* 17:026103
- Dobson I, Carreras BA, Newman DE, Reynolds-Barredo JM (2016) Obtaining statistics of cascading line outages spreading in an electric transmission network from standard utility data. *IEEE Trans Power Syst* 31(6):4831–4841
- Federal Energy Regulatory Commission and the North American Electric Reliability Corporation (2012) Arizona-Southern California outages on September 8, 2011: causes and recommendations
- Hardiman RC, Kumbale MT, Makarov YV (2004) An advanced tool for analyzing multiple cascading failures. In: Eighth international conference on probability methods applied to power systems, Ames, Sept 2004
- Harris TE (1989) *Theory of branching processes*. Dover, New York
- Hines P, Apt J, Talukdar S (2009) Large blackouts in North America: historical trends and policy implications. *Energy Policy* 37(12):5249–5259
- Hines PDH, Dobson I, Rezaei P (2017) Cascading power outages propagate locally in an influence graph that is not the actual grid topology. *IEEE Trans Power Syst* 32(2):958–967
- IEEE PES CAMS Task Force on Cascading Failure (2008) Initial review of methods for cascading failure analysis in electric power transmission systems. In: IEEE power and energy society general meeting, Pittsburgh, July 2008
- IEEE Working Group on Cascading Failures (2016) Benchmarking and validation of cascading failure analysis tools. *IEEE Trans Power Syst* 31(6):4887–4900
- Kirschen DS, Strbac G (2004) Why investments do not prevent blackouts. *Electr J* 17(2):29–36
- Kirschen DS, Jawayeera D, Nedec DP, Allan RN (2004) A probabilistic indicator of system stress. *IEEE Trans Power Syst* 19(3):1650–1657
- Kosterev D, Taylor C, Mittelstadt W (1999) Model validation for the August 10, 1996 WSCC system outage. *IEEE Trans Power Syst* 14:967–979
- Newman DE, Carreras BA, Lynch VE, Dobson I (2011) Exploring complex systems aspects of blackout risk and mitigation. *IEEE Trans Reliab* 60(1):134–143
- Rahnamay-Naeini M, Wang Z, Mammoli A, Hayat MMA (2012) Probabilistic model for the dynamics of cascading failures and blackouts in power grids. In: IEEE power and energy society general meeting, July 2012
- Ren H, Dobson I, Carreras BA (2008) Long-term effect of the n-1 criterion on cascading line outages in an evolving power transmission grid. *IEEE Trans Power Syst* 23(3):1217–1225
- Rinaldi SM, Peerenboom JP, Kelly TK (2001) Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Syst Mag* 21: 11–25
- US-Canada Power System Outage Task Force (2004) Final report on the August 14, 2003 Blackout in the United States and Canada

Cash Management

Abel Cadenillas

University of Alberta, Edmonton, AB, Canada

Abstract

Cash on hand (or cash held in highly liquid form in a bank account) is needed for routine business and personal transactions. The problem of determining the right amount of cash to hold involves balancing liquidity against investment opportunity costs. This entry traces solutions using both discrete-time and continuous-time stochastic models.

Keywords

Brownian motion · Inventory theory ·
Stochastic impulse control

Definition

A firm needs to keep cash, either in the form of cash on hand or as a bank deposit, to meet its daily transaction requirements. Daily inflows and outflows of cash are random. There is a finite target for the cash balance, which could be zero in some cases. The firm wants to select a policy that minimizes the expected total discounted cost for being far away from the target during some time horizon. This time horizon is usually infinity. The firm has an incentive to keep the cash level low, because each unit of positive cash leads to a holding cost since cash has alternative uses like dividends or investments in earning assets. The firm has an incentive to keep the cash level high, because penalty costs are generated as a result of delays in meeting demands for cash. The firm can increase its cash balance by raising new capital or by selling some earnings assets, and it can reduce its cash balance by paying dividends or investing in earning assets. This control of the cash balance generates fixed and proportional transaction costs. Thus, there is a cost when the cash balance is different from its target, and there is also a cost for increasing or reducing the cash reserve. The objective of the manager is to minimize the expected total discounted cost.

Hasbrouck (2007), Madhavan and Smidt (1993), and Manaster and Mann (1996) study inventories of stocks that are similar to the cash management problem.

The Solution

The qualitative form of optimal policies of the cash management problem in discrete time was studied by Eppen and Fama (1968, 1969), Girgis (1968), and Neave (1970). However, their solutions were incomplete.

Many of the difficulties that they and other researchers encountered in a discrete-time framework disappeared when it was assumed that decisions were made continuously in time and that demand is generated by a Brownian motion with drift. Vial (1972) formulated the cash management problem in continuous time with fixed and proportional transaction costs, linear holding and penalty costs, and demand for cash generated by a Brownian motion with drift. Under very strong assumptions, Vial (1972) proved that if an optimal policy exists, then it is of a simple form (a, α, β, b) .

This means that the cash balance should be increased to level α when it reaches level a and should be reduced to level β when it reaches level b . Constantinides (1976) assumed that an optimal policy exists and it is of a simple form and determined the above levels and discussed the properties of the optimal solution. Constantinides and Richard (1978) proved the main assumptions of Vial (1972) and therefore obtained rigorously a solution for the cash management problem.

Constantinides and Richard (1978) applied the theory of stochastic impulse control developed by Bensoussan and Lions (1973, 1975, 1982). They used a Brownian motion W to model the uncertainty in the inventory. Formally, they considered a probability space $(\Omega, \mathcal{F}, P)$ together with a filtration $(\mathcal{F}_t)$ generated by a one-dimensional Brownian motion W . They considered

X_t := inventory level at time t and assumed that X is an adapted stochastic process given by

$$X_t = x - \int_0^t \mu ds - \int_0^t \sigma dW_s + \sum_{i=1}^{\infty} I_{\{\tau_i < t\}} \xi_i.$$

Here, $\mu > 0$ is the drift of the demand and $\sigma > 0$ is the volatility of the demand. Furthermore, τ_i is the time of the i -th intervention and ξ_i is the intensity of the i -th intervention.

A stochastic impulse control is a pair

$((\tau_n); (\xi_n))$

$= (\tau_0, \tau_1, \tau_2, \dots, \tau_n, \dots; \xi_0, \xi_1, \xi_2, \dots, \xi_n, \dots),$

where

$$\tau_0 = 0 < \tau_1 < \tau_2 < \dots < \tau_n < \dots$$

is an increasing sequence of stopping times and (ξ_n) is a sequence of random variables such that each $\xi_n : \Omega \mapsto \mathbf{R}$ is measurable with respect to $\mathcal{F}_{\tau_n}$. We assume $\xi_0 = 0$. The management (the controller) decides to act at time

$$X_{\tau_i^+} = X_{\tau_i} + \xi_i.$$

We note that ξ_i and X can also take negative values. The management wants to select the pair

$$((\tau_n); (\xi_n))$$

that minimizes the functional J defined by

$$J(x; ((\tau_n); (\xi_n))) := E \left[\int_0^\infty e^{-\lambda t} f(X_t) dt + \sum_{n=1}^\infty e^{-\lambda \tau_n} g(\xi_n) I_{\{\tau_n < \infty\}} \right],$$

where

$$f(x) = \max(hx, -px)$$

and

$$g(\xi) = \begin{cases} C + c\xi & \text{if } \xi > 0 \\ \min(C, D) & \text{if } \xi = 0 \\ D - d\xi & \text{if } \xi < 0 \end{cases}$$

Furthermore, $\lambda > 0, C, c, D, d \in (0, \infty)$, and $h, p \in (0, \infty)$. Here, f represents the running cost incurred by deviating from the aimed cash level 0, C represents the fixed cost per intervention when the management pushes the cash level upwards, D represents the fixed cost per intervention when the management pushes the cash level downwards, c represents the proportional cost per intervention when the management pushes the cash level upwards, d represents the proportional cost per intervention when the management pushes the cash level downwards, and λ is the discount rate.

The results of Constantinides and Richard (1978) were complemented, extended, or improved by Baccarin (2009), Bar-Ilan et al.

(2004), Cadenillas et al. (2010), Cadenillas and Zapatero (1999), Feng and Muthuraman (2010), Harrison et al. (1983), and Ormeci et al. (2008).

Cross-References

- [Financial Markets Modeling](#)
- [Inventory Theory](#)

Bibliography

- Baccarin S (2009) Optimal impulse control for a multidimensional cash management system with generalized cost functions. *Eur J Oper Res* 196:198–206
- Bar-Ilan A, Perry D, Stadje W (2004) A generalized impulse control model of cash management. *J Econ Dyn Control* 28:1013–1033
- Bensoussan A, Lions JL (1973) Nouvelle formulation de problemes de controle impulsif et applications. *C R Acad Sci (Paris) Ser A* 276:1189–1192
- Bensoussan A, Lions JL (1975) Nouvelles methodes en controle impulsif. *Appl Math Opt* 1:289–312
- Bensoussan A, Lions JL (1982) Controle impulsif et inequations quasi variationnelles. Bordas, Paris
- Cadenillas A, Zapatero F (1999) Optimal Central Bank intervention in the foreign exchange market. *J Econ Theory* 87:218–242
- Cadenillas A, Lakner P, Pinedo M (2010) Optimal control of a mean-reverting inventory. *Oper Res* 58:1697–1710
- Constantinides GM (1976) Stochastic cash management with fixed and proportional transaction costs. *Manag Sci* 22:1320–1331
- Constantinides GM, Richard SF (1978) Existence of optimal simple policies for discounted-cost inventory and cash management in continuous time. *Oper Res* 26:620–636
- Eppen GD, Fama EF (1968) Solutions for cash balance and simple dynamic portfolio problems. *J Bus* 41:94–112
- Eppen GD, Fama EF (1969) Cash balance and simple dynamic portfolio problems with proportional costs. *Int Econ Rev* 10:119–133
- Feng H, Muthuraman K (2010) A computational method for stochastic impulse control problems. *Math Oper Res* 35:830–850
- Girgis NM (1968) Optimal cash balance level. *Manag Sci* 15:130–140
- Harrison JM, Sellke TM, Taylor AJ (1983) Impulse control of Brownian motion. *Math Oper Res* 8:454–466
- Hasbrouck J (2007) Empirical market microstructure. Oxford University Press, New York
- Madhavan A, Smidt S (1993) An analysis of changes in specialist inventories and quotations. *J Financ* 48:1595–1628

- Manaster S, Mann SC (1996) Life in the pits: competitive market making and inventory control. *Rev Financ Stud* 9:953–975
- Neave EH (1970) The stochastic cash-balance problem with fixed costs for increases and decreases. *Manag Sci* 16:472–490
- Ormeci M, Dai JG, Vande Vate J (2008) Impulse control of Brownian motion: the constrained average cost case. *Oper Res* 56:618–629
- Vial JP (1972) A continuous time model for the cash balance problem. In: Szego GP, Shell C (eds) *Mathematical methods in investment and finance*. North Holland, Amsterdam

Characteristics in Optimal Control Computation

Ivan Yegorov¹ and Peter M. Dower²

¹Department of Mathematics, North Dakota State University, Fargo, ND, USA

²Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC, Australia

Abstract

A review of characteristics-based approaches to optimal control computation is presented for nonlinear deterministic systems described by ordinary differential equations. We recall Pontryagin's principle which gives necessary optimality conditions for open-loop control strategies. The related framework of generalized characteristics for first-order Hamilton–Jacobi–Bellman equations is discussed as a theoretical tool that bridges the gap between the necessary and sufficient optimality conditions. We point out widely used numerical techniques for obtaining optimal open-loop control strategies (in particular for state-constrained problems) and how indirect (characteristics based) and direct methods may reasonably be combined. A possible transition from open-loop to feedback constructions is also described. Moreover, we discuss approaches for attenuating the curse of dimensionality for certain classes of Hamilton–Jacobi–Bellman equations.

Keywords

Optimal control · Pontryagin's principle · Hamilton–Jacobi–Bellman equations · Method of characteristics · Indirect and direct numerical approaches · State constraints · Curse of dimensionality

Introduction

Optimal control problems for dynamical systems described by ordinary differential equations have various applications in science and engineering (Afanas'ev et al. 1996; Pesch 1994; Trélat 2012; Schättler and Ledzewicz 2015). One has to distinguish between the classes of open-loop and feedback (closed-loop) control strategies; the former are functions of time, while the latter are functions of time and state. In comparison with optimal open-loop control strategies, optimal feedback control strategies are typically more advantageous in practice, since they are applicable to all considered initial states and more robust with respect to uncertainties.

It is common to distinguish between two groups of numerical approaches to constructing optimal open-loop control strategies: direct and indirect methods (Grüne 2014; Pesch 1994; Trélat 2012; Benson et al. 2006; Rao 2010; Cristiani and Martinon 2010). Direct methods involve the direct transcriptions of infinite-dimensional optimal open-loop control problems to finite-dimensional nonlinear programming problems via discretizations in time applied to state and control variables, as well as to dynamical state equations. Indirect methods employ Pontryagin's principle (Pontryagin et al. 1964; Cesari 1983; Yong and Zhou 1999; Schättler 2014). Comparatively, direct methods are in principle less precise and less justified from a theoretical perspective than indirect methods but are often more robust with respect to initialization and more straightforward to use.

For describing optimal feedback control strategies, Hamilton–Jacobi–Bellman partial differential equations (HJB PDEs) and sufficient optimality conditions constitute a fundamental

theoretical framework (Fleming and Soner 2006; Bardi and Capuzzo-Dolcetta 2008; Subbotin 1995; Melikyan 1998; Yong and Zhou 1999). Exact solutions of boundary value, initial value, and mixed problems for HJB PDEs are known only in very special cases. Many widely used numerical approaches to solving these problems, including semi-Lagrangian schemes (Bardi and Capuzzo-Dolcetta 2008; Bokanowski et al. 2017), finite-difference schemes (Fleming and Soner 2006; Kushner and Dupuis 2001; Bokanowski et al. 2017), finite element methods (Jensen and Smears 2013), and level set methods (Osher and Fedkiw 2003), typically rely on dense state space discretizations. Where a problem for an HJB PDE is stated in an unbounded domain of the state space, one has to select an appropriate bounded subdomain for discretization and computation (which is often a heuristic choice); see, e.g., (Fleming and Soner 2006; Kushner and Dupuis 2001; Bardi and Capuzzo-Dolcetta 2008). A more critical aspect is that the computational cost of such grid-based numerical methods grows exponentially with the increase of the state space dimension, so that their feasible implementation is in general extremely limited (even on supercomputers) if the state space dimension is greater than 3. This leads to what R. Bellman called the curse of dimensionality (Bellman 1957). Attenuating the curse of dimensionality for various classes of HJB PDEs and also more general Hamilton–Jacobi (HJ) PDEs, such as Hamilton–Jacobi–Isaacs (HJI) PDEs for zero-sum two-player differential games, has therefore become an important research area. A number of related approaches have been developed for particular classes of problems; see, e.g., the corresponding overview in Yegorov and Dower (2018). It is emphasized that, even if the curse of dimensionality is mitigated, the so-called curse of complexity may still cause significant issues in numerical implementation (McEneaney 2006; McEneaney and Kluberg 2009; Kang and Wilcox 2017).

A relevant direction for attenuating the curse of dimensionality for certain classes of first-order HJ equations is to reduce the

evaluation of their solutions at any selected states to finite-dimensional optimization (nonlinear programming) problems (Darbon and Osher 2016; Chow et al. 2017; Kang and Wilcox 2017). In contrast with many grid-based methods, this leads to the following advantages:

- the solutions can be evaluated independently at different states, which allows for attenuating the curse of dimensionality;
- since different states are separately treated, one can choose arbitrary bounded regions and grids for computations and arrange parallelization;
- when obtaining the value function, i.e., the solution of an HJB or HJI equation, at selected states by solving the related finite-dimensional optimization problems, one can usually retrieve the corresponding control actions without requiring possibly unstable approximations of the partial derivatives of the value function.

However, the curse of complexity can ensue if the considered nonlinear programming problem is essentially multi-extremal or if one wants to construct a global or semi-global solution approximation in a high-dimensional region. Sparse grid frameworks can be helpful for the latter purpose in some cases (Kang and Wilcox 2017). Other approaches to mitigating the curse of dimensionality and the curse of complexity for certain classes of nonlinear optimal control problems include methods based on the max- and min-plus algebras and corresponding idempotent analysis (McEneaney 2006; McEneaney and Kluberg 2009).

The finite-dimensional optimization problems that describe the values at arbitrary isolated states of the solutions of first-order HJB equations can build on the aforementioned direct and indirect approaches to approximating optimal open-loop control strategies. Note also that the indirect approaches using Pontryagin’s principle relate to the (generalized) method of characteristics for such nonlinear first-order PDEs and can hence be called characteristics based. The

method of characteristics often serves as a theoretical tool that bridges the gap between the necessary and sufficient optimality conditions (Subbotin 1995; Melikyan 1998; Subbotina 2006). The purpose of this entry is to review some existing and recent developments regarding the characteristics-based approaches to optimal control computation.

Pontryagin's Principle and Characteristics

We start with a discussion of Pontryagin's principle (Pontryagin et al. 1964; Cesari 1983; Yong and Zhou 1999; Hartl et al. 1995). For simplicity, consider a deterministic time-invariant system given by

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)), & u(t) \in U, & t \in I, \\ I = [0, T] \text{ in case of fixed terminal time } T, \text{ and } I = [0, +\infty) \text{ if } T \text{ is free,} \\ x(0) = x_0 \in G \text{ is fixed,} \\ \mathcal{U} \text{ is the class of measurable open-loop control functions } u: I \rightarrow U, \end{cases} \quad (1)$$

with the terminal constraint

$$x(T) \in \Omega \quad (2)$$

and with the objective of minimizing a Bolza functional over admissible open-loop control strategies:

$$\varphi(x(T)) + \int_0^T \ell(x(t), u(t)) dt \longrightarrow \min. \quad (3)$$

In (1), (2), and (3), $t \geq 0$, $x(t) \in \mathbb{R}^{n \times 1}$, and $u(t) \in \mathbb{R}^{m \times 1}$ are the time, state, and control variables, respectively; both cases of fixed and free terminal time T are incorporated, and the following are assumed:

- $U \subset \mathbb{R}^m$ is a closed convex set that specifies pointwise control constraints, and $G \subseteq \mathbb{R}^n$ is an open domain in the state space;
- the functions $f: \bar{G} \times U \rightarrow \mathbb{R}^n$, $\varphi: \bar{G} \rightarrow \mathbb{R}$, and $\ell: \bar{G} \times U \rightarrow \mathbb{R}$ are continuous ($\bar{G}$ denotes the closure of G), and they are also continuously differentiable with respect to the state variable in G ;
- G is a strongly invariant domain in the sense that any state trajectory of (1) with $x_0 \in G$ and $u(\cdot) \in \mathcal{U}$ does not leave G ($G = \mathbb{R}^n$ is a trivial example of a strongly invariant domain);

- for any $x_0 \in G$ and $u(\cdot) \in \mathcal{U}$, there exists a unique solution $x: I \rightarrow \mathbb{R}^n$ of (1);
- $\Omega \subseteq \mathbb{R}^n$ is a closed terminal set in the state space.

We refer to Cesari (1983) for a survey of sufficient conditions for the existence of optimal open-loop control strategies in various classes of problems.

The Hamiltonian is defined by

$$\begin{aligned} H(x, u, p, \tilde{p}) &= \langle p, f(x, u) \rangle + \tilde{p} \ell(x, u) \\ \forall (x, u, p, \tilde{p}) &\in G \times U \times \mathbb{R}^n \times \mathbb{R}, \end{aligned} \quad (4)$$

where $p \in \mathbb{R}^{n \times 1}$ is the so-called costate or adjoint variable. Pontryagin's (minimum) principle states that, if $(u^*(\cdot), x^*(\cdot), T^*)$ is an optimal process for the problem (1), (2), and (3), there exist a function $p^*: [0, T^*] \rightarrow \mathbb{R}^{n \times 1}$ and a constant $\tilde{p}^* \geq 0$ such that the following properties hold:

- $(p^*(t), \tilde{p}^*)$ is nonzero for every $t \in [0, T^*]$;
- $(x^*(\cdot), p^*(\cdot))$ is an absolutely continuous solution of the characteristic system

$$\begin{aligned} \dot{x}^*(t) &= H_p(x^*(t), u^*(t), p^*(t), \tilde{p}^*) \\ &= f(x^*(t), u^*(t)), \end{aligned}$$

$$\dot{p}^*(t) = -H_x(x^*(t), u^*(t), p^*(t), \tilde{p}^*) \quad (5)$$

on $[0, T^*]$ with the boundary conditions

$$\begin{aligned} x^*(0) &= x_0, \quad x^*(T^*) \in \Omega, \\ p^*(T^*) &= \tilde{p}^* \varphi_x(x^*(T^*)) \\ &\in N(x^*(T^*); \Omega), \end{aligned} \quad (6)$$

where $N(x^*(T^*); \Omega)$ denotes the normal cone to Ω at $x^*(T^*)$, which is polar to the related tangent cone (Clarke et al. 1998, §2.5);

- the Hamiltonian minimum condition

$$\begin{aligned} H(x^*(t), u^*(t), p^*(t), \tilde{p}^*) \\ = \min_{w \in U} H(x^*(t), w, p^*(t), \tilde{p}^*) \end{aligned} \quad (7)$$

is satisfied for almost all $t \in [0, T^*]$ (with respect to the Lebesgue measure);

- if the terminal time is not fixed in the problem formulation, the Hamiltonian vanishes along the optimal process, i.e., $H(x^*(t), u^*(t), p^*(t), \tilde{p}^*) = 0$ for almost all $t \in [0, T^*]$.

For example, the terminal costate $p^*(T^*)$ is free if Ω is a singleton, while $p^*(T^*) = \tilde{p}^* \varphi_x(x^*(T^*))$ if $\Omega = \mathbb{R}^n$. A solution of the system (1) satisfying the Hamiltonian minimum condition (7) and such that $(p^*(t), \tilde{p}^*)$ does not vanish is called a characteristic (regardless of whether the boundary conditions (6) are fulfilled or not), in line with the generalized method of characteristics for first-order HJB

PDEs (Subbotin 1995; Melikyan 1998; Subbotina 2006). A characteristic fully satisfying Pontryagin's principle is called extremal. We say that a characteristic with $\tilde{p}^* = 0$ is abnormal, while $\tilde{p}^* > 0$ indicates a normal characteristic. A singular arc of a characteristic occurs when the set of minimizers in (7) is not a singleton on a time subinterval.

The value function in the optimal control problem (1), (2), and (3) maps the initial state x_0 to the infimum of the cost functional. This particular problem is formulated with zero initial instant $t_0 = 0$, but, in general, the value function depends on both t_0 and $x_0 = x(t_0)$ (and the integral in (3) is taken over $[t_0, T]$). Value functions satisfy HJB equations in the viscosity sense and may be nonsmooth (Fleming and Soner 2006; Bardi and Capuzzo-Dolcetta 2008; Yong and Zhou 1999). A number of verification theorems as well as relations between the costate components of extremal characteristics and the super- and subdifferentials of value functions have been established; see, e.g., Yong and Zhou (1999, Chapter 5). Such results in fact serve as a theoretical connection between Pontryagin's principle (necessary optimality conditions) and HJB equations (used in sufficient optimality conditions). Note that the mentioned relations are markedly simplified where value functions are differentiable (so that the super- and subdifferentials are reduced to the gradients).

In the specific case of fixed terminal time T and free terminal state ($\Omega = \mathbb{R}^n$), the Cauchy problem for the HJB equation in the problem (1), (2), and (3) is formally written as follows:

$$\begin{cases} -\frac{\partial V(x, t)}{\partial t} + \sup_{w \in U} \left\{ -\left\langle \frac{\partial V(x, t)}{\partial x}, f(x, w) \right\rangle - \ell(x, w) \right\} = 0, & (x, t) \in G \times [0, T), \\ V(x, T) = \varphi(x), & x \in G. \end{cases}$$

Pontryagin's principle has been extended to certain classes of optimal control problems with state constraints, leading to a number of new possible features of related characteristics, in particular junctions and contacts with

state-constraint boundaries, boundary arcs, as well as costate jumps (Schättler 2014; Hartl et al. 1995; Maurer 1977; Bonnard et al. 2003; Pesch 1994; Cristiani and Martinon 2010; Basco et al. 2018).

Characteristics-Based Approaches to Approximating Optimal Open-Loop Control Strategies

For simplicity, consider an optimal open-loop control problem as formulated in the previous section, or, in order to address a state-constrained case as well, its modification to include a scalar pure state inequality constraint of finite order and under the additional assumptions of scalar control input and control-affine dynamical equations (Maurer 1977; Bonnard et al. 2003; Pesch 1994; Cristiani and Martinon 2010). Pontryagin's principle allows for reducing the studied problem to a characteristic boundary value problem with accompanying conditions. It is typical to suppose that the Hamiltonian minimum condition determines the extremal control action as a single-valued function of the state x and extended costate $(p, \tilde{p})$ almost everywhere outside singular and constrained boundary arcs which have to be separately investigated. The analysis of the singular and boundary arcs can also provide the related representations of the control u and constraint-based Lagrange multiplier function η in terms of $x, p, \tilde{p}$ (Maurer 1977; Bonnard et al. 2003; Pesch 1994; Cristiani and Martinon 2010), and this is assumed here.

For computational purposes, the characteristic boundary value problem can be stated with a finite number of unknowns, even though the original control problem is infinite-dimensional. The first unknown in the characteristic boundary value problem is the initial costate $p(0)$. Due to the homogeneity of the Hamiltonian with respect to $(p, \tilde{p}, \eta)$, the normal case $\tilde{p} > 0$ can be reduced to $\tilde{p} = 1$, while the abnormal case $\tilde{p} = 0$ must be separately treated. The abnormal case is sometimes excluded a priori from practical consideration, since the Hamiltonian (4) does not depend on the running cost $\ell(x, u)$ for $\tilde{p} = 0$. Alternatively, one can consider $(p(0), \tilde{p})$ as an unknown vector on the unit sphere in $\mathbb{R}^{n+1}$. Bearing in mind the Hamiltonian conservation property for time-invariant control systems, the Hamiltonian vanishing condition in case of free terminal time T can be imposed just at $t = T$.

An unconstrained singular arc adds unknown entry and exit times together with the corresponding junction conditions to the characteristic boundary value problem. Note that, for a system with scalar control input and control-affine dynamical equations, the coefficient for the control in the Hamiltonian is called the switching function. Moreover, if a singular arc is of first order, i.e., if the control variable explicitly appears in the second total time derivative of the switching function (such an appearance may in general take place first for an even derivative), then the junction conditions can be specified at the entry time as vanishing of the switching function together with its first total time derivative (Pesch 1994; Cristiani and Martinon 2010; Bonnard et al. 2003).

Note also that, for certain classes of problems, singular regimes can be excluded for normal characteristics if one adds a small regularizing term such as a control-dependent positive definite quadratic form to the running cost, so that the Hamiltonian minimum condition leads to a single-valued extremal control map in the normal case.

A boundary arc adds unknown entry and exit times together with the junction conditions, Hamiltonian continuity, and costate jump conditions at the entry and exit times, and the costate jump parameters are other unknowns which may nevertheless be described by analytical representations in certain cases (Maurer 1977; Bonnard et al. 2003). Expected contacts with the constraint boundary can be treated similarly.

Thus, solving the characteristic boundary value problem corresponds to finding the unknown parameters leading to an extremal characteristic that satisfies all terminal and path constraints. This is usually done via the numerical approach of multiple shooting (Grüne 2014; Rao 2010; Bonnard et al. 2003; Trélat 2012; Pesch 1994; Cristiani and Martinon 2010). It in particular requires a priori guesses for the numbers of unconstrained singular arcs, constrained boundary arcs, and contacts with the constraint boundary on a sought-after characteristic. The numerical efficiency can in principle be improved

if one uses an indirect collocation method so that the state and costate trajectories are parametrized via specific piecewise polynomial functions (Rao 2010; Kang and Wilcox 2017). Such methods include local collocation and global collocation. For local collocation, the degrees of local polynomials are fixed and the number of meshes is varied to improve accuracy. On the other hand, global collocation fixes the number of meshes and varies the degree of a global polynomial to improve accuracy.

The characteristic boundary value problem may have multiple solutions for the initial states at which the state components of some characteristics with different terminal states intersect each other and the value function is therefore expected to be nonsmooth (Yong and Zhou 1999, Chapters 4, 5). This may cause a practical dilemma of how to retrieve optimal characteristics for some initial states in small neighborhoods of switching manifolds and near constraint boundaries. A number of studies hence proposed to replace the terminal costate condition with the cost optimization criterion, leading to constrained nonlinear programming instead of constrained root-finding (Yegorov and Dower 2018; Darbon and Osher 2016). Figure 1 is taken from Yegorov and Dower (2018, Example 5.1) and compares the numerical results of solving the optimization and boundary value problems for the characteristics of a particular Eikonal type HJB PDE.

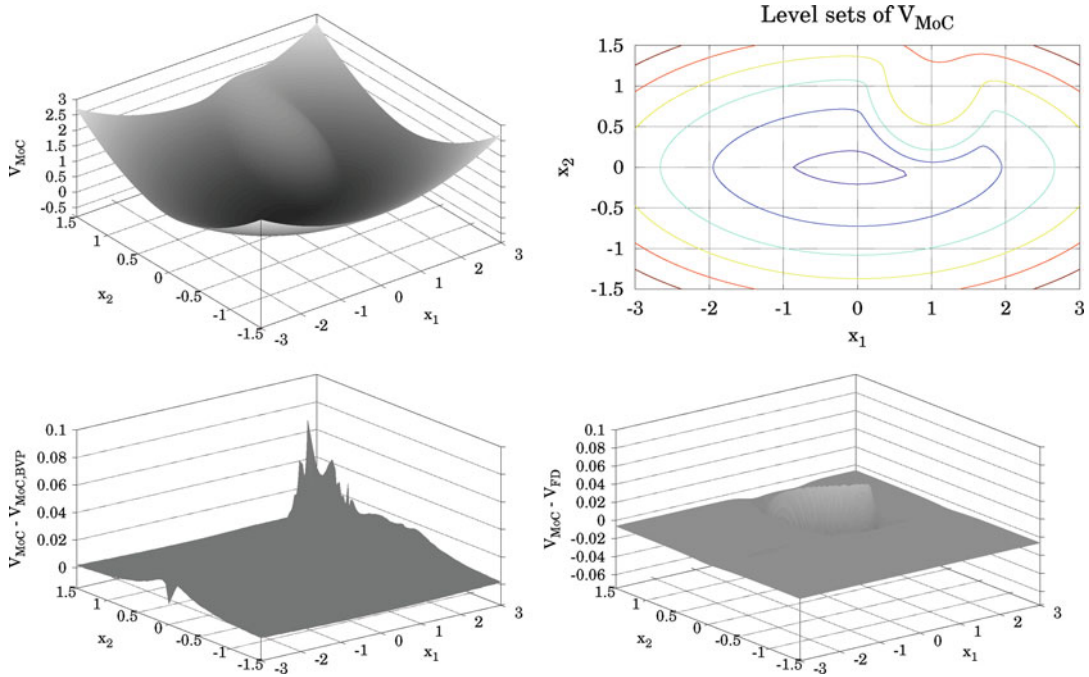
Since indirect (characteristics based) methods for approximating optimal open-loop control strategies are often rather sensitive with respect to initialization, more robust direct methods may serve as an auxiliary tool for finding appropriate initial conditions (Pesch 1994; Cristiani and Martinon 2010; Trélat 2012).

Related efforts in the development of spectral methods for fluid dynamics have led to a class of direct numerical approaches referred to as pseudospectral methods that allow for costate estimation and find application in a range of practical problems (Benson et al. 2006; Huntington et al. 2007; Rao 2010). A pseudospectral method involves global orthogonal collocation. That is, the state and control variables are parametrized

via a basis of global polynomials, the dynamics are collocated at a specific set of points on the observed time interval, and the collocation points are chosen as the roots of an orthogonal polynomial or linear combinations of such polynomials and their derivatives. For smooth problems, these methods exhibit the so-called spectral accuracy that can be a precursor to fast convergence. Non-smooth problems can be divided into phases so that orthogonal collocation is applied globally within each phase. For example, the Gauss pseudospectral method uses global Lagrange polynomials and Legendre–Gauss points for collocation, and it differs from several other pseudospectral methods in that the dynamics are not collocated at the boundary points. This choice of collocation, together with an appropriate Lagrange polynomial approximation of the costate variable, leads to the Karush–Kuhn–Tucker conditions that are identical to the discretized form of the first-order necessary optimality conditions (Pontryagin’s principle) at the Legendre–Gauss points. Thus, the Karush–Kuhn–Tucker multipliers of the direct nonlinear programming problems can be used to estimate the costate components of the extremal characteristics at the collocation points as well as at the boundary points. Figure 2 is a schematic illustration of this costate estimation technique. More details can be found in Benson et al. (2006), Huntington et al. (2007), and Rao (2010).

Transition from Open-Loop to Feedback Constructions

As was discussed in the introduction, reducing the computation of value functions and optimal control actions at selected states to finite-dimensional nonlinear programming problems, e.g., by solving optimal open-loop control problems via indirect (characteristics based) or direct methods, helps to attenuate the curse of dimensionality for HJB PDEs. It was however noted that the curse of complexity can represent a formidable barrier to global or semi-global approximation of sought-after functions in high-dimensional regions. Moreover, the curse of



Characteristics in Optimal Control Computation,

Fig. 1 These illustrations are reproduced from Yegorov and Dower (2018, Example 5.1). (Permission granted by Springer Nature under Copyright Clearance Center's RightsLink® license number 4641070769945). They correspond to a certain nonlinear optimal control problem whose value function is a unique viscosity solution of a Cauchy problem for an Eikonal type first-order HJB PDE. The state space is two-dimensional, and the goal is to maximize a Mayer cost functional. The value function was computed via the following three different approaches: (i) optimization over the characteristic Cauchy problems, (ii) solving the characteristic boundary value problems, and (iii) solving the Cauchy problem for the HJB PDE via a finite-difference scheme. The related value function approximations are denoted by V_{MoC} , $V_{MoC,BVP}$, and V_{FD} , respectively (the abbreviations “MoC”, “BVP”, and “FD” stand for “method of characteristics”, “boundary

value problem”, and “finite differences”). The upper two subfigures illustrate a section of V_{MoC} for a fixed time with some of its level sets. The lower left and lower right subfigures show, respectively, the differences $V_{MoC} - V_{MoC,BVP}$ and $V_{MoC} - V_{FD}$ at the same fixed time (in order to see these graphs clearer, essentially larger scales for the vertical axis are used). One has $V_{MoC} \geq V_{MoC,BVP}$ due to the maximization criterion and how the approaches are stated. Since optimization over the characteristic Cauchy problems is more justified from a theoretical perspective, one can interpret the indicated differences as error estimates for the other two approaches. It is seen that, near some of the points for which the value function is nonsmooth and the characteristic boundary value problem therefore admits multiple solutions, the values of $V_{MoC} - V_{MoC,BVP}$ are not negligible and are noticeably greater than the corresponding values of $|V_{MoC} - V_{FD}|$.

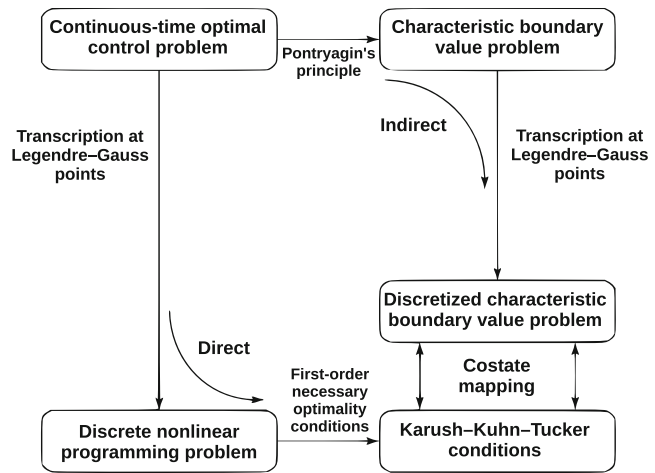
complexity has a strong impact on the practical implementation of related model predictive control schemes, where the optimal control action has to be recomputed at sampling instants using the available data on the current state (see, e.g., Wang 2009; Kang and Wilcox 2017) and the intervals between consecutive sampling instants are not large enough in comparison with the average time required for one control evaluation.

For certain classes of problems, sparse grid frameworks may help to attenuate the curse of

complexity if the state space dimension is moderate (typically not greater than 6) and if sparse grid interpolation is applied to functions with sufficiently smooth behavior (Kang and Wilcox 2017). In this case, the data on a sparse grid is generated offline (the possibility to independently treat different nodes on the sparse grid via a curse-of-dimensionality-free method allows the offline computation to be parallelized), and the online control evaluation for an arbitrary state in a certain bounded region is performed by

Characteristics in Optimal Control Computation, Fig. 2

Costate mapping between the direct Gauss pseudospectral discretization and Pontryagin's principle. This is a slight modification of Benson et al. (2006, Fig. 1)



using sparse grid interpolation. Note that interpolation of costates is preferable to that of feedback control strategies if the latter are expected to have a sharper behavior. Although the theoretical estimates for sparse grid interpolation errors are rather conservative and the actual errors may be very large, this approach could provide acceptable control performance for some practical models, as was demonstrated by Kang and Wilcox (2017).

Summary and Future Directions

In this entry, we briefly discussed some important characteristics-based approaches to approximating optimal open-loop control strategies and in particular how such indirect approaches may reasonably be combined with direct methods, as well as a possible transition to feedback control approximations via sparse grid techniques. To conclude, we point out some relevant directions of future research in related fields.

Kang and Wilcox (2017) used Smolyak's sparse grid constructions with hierarchical polynomial interpolation. Other methods of scattered data interpolation (e.g., Kriging) may be tested as well. In general, the range of applicability of such sparse grid frameworks to solving various feedback control problems is worth investigating.

It is also reasonable to use the discussed approaches for solving the exit-time optimal control problems that describe asymptotically stabilizing strategies (Michalska and Mayne 1993). Since the indirect (characteristics based) and direct collocation techniques enable costate estimation, it is relevant to design methods for constructing piecewise affine control Lyapunov functions in relatively high dimensions.

Another challenging direction is the development of curse-of-dimensionality-free approaches to solving HJI PDEs that arise in particular classes of zero-sum closed-loop two-player differential games; see, e.g., the related discussions in Yegorov and Dower (2018), Darbon and Osher (2016), and Chow et al. (2017). For general classes of nonlinear systems, this appears to be very difficult, due to potential singularities in the behavior of characteristics for zero-sum two-player differential games that include equivocal, envelope, and focal manifolds (Bernhard 1977; Melikyan 1998).

Cross-References

- Numerical Methods for Nonlinear Optimal Control Problems
- Optimal Control and Pontryagin's Maximum Principle

- [Optimal Control and the Dynamic Programming Principle](#)
- [Optimal Control with State Space Constraints](#)

Bibliography

- Afanas'ev VN, Kolmanovskii VB, Nosov VR (1996) Mathematical theory of control systems design. Springer Science + Business Media, Dordrecht
- Bardi M, Capuzzo-Dolcetta I (2008) Optimal control and viscosity solutions of Hamilton–Jacobi–Bellman equations. Birkhäuser, Boston
- Basco V, Cannarsa P, Frankowska H (2018) Necessary conditions for infinite horizon optimal control problems with state constraints. *Math Control Relat F* 8(3&4):535–555
- Bellman R (1957) Dynamic programming. Princeton University Press, Princeton
- Benson DA, Huntington GT, Thorvaldsen TP, Rao AV (2006) Direct trajectory optimization and costate estimation via an orthogonal collocation method. *J Guid Control Dyn* 29(6):1435–1440
- Bernhard P (1977) Singular surfaces in differential games: an introduction. In: Hagedorn P, Knobloch HW, Olsder GJ (eds) Differential games and applications. Lecture notes in control and information sciences, vol 3, pp 1–33. Springer, Berlin, Heidelberg
- Bokanowski O, Desilles A, Zidani H, Zhao J (2017) User's guide for the ROC-HJ solver: reachability, optimal control, and Hamilton–Jacobi equations. Version 2.3. <http://uma.ensta-paristech.fr/soft/ROC-HJ/>
- Bonnard B, Faubourg L, Launay G, Trélat E (2003) Optimal control with state constraints and the space shuttle re-entry problem. *J Dyn Control Syst* 9(2): 155–199
- Cesari L (1983) Optimization—theory and applications, problems with ordinary differential equations. Applications of mathematics, vol 17. Springer, New York
- Chow YT, Darbon J, Osher S, Yin W (2017) Algorithm for overcoming the curse of dimensionality for time-dependent non-convex Hamilton–Jacobi equations arising from optimal control and differential games problems. *J Sci Comput* 73(2–3):617–643
- Clarke FH, Ledyaev YS, Stern RJ, Wolenski PR (1998) Nonsmooth analysis and control theory. Springer, New York
- Cristiani E, Martinon P (2010) Initialization of the shooting method via the Hamilton–Jacobi–Bellman approach. *J Optim Theory Appl* 146(2):321–346
- Darbon J, Osher S (2016) Algorithms for overcoming the curse of dimensionality for certain Hamilton–Jacobi equations arising in control theory and elsewhere. *Res Math Sci* 3:19
- Fleming WH, Soner HM (2006) Controlled Markov processes and viscosity solutions. Springer, New York
- Grüne L (2014) Numerical methods for nonlinear optimal control problems. In: Baillieul J, Samad T (eds) Encyclopedia of systems and control. Springer, London
- Hartl RF, Sethi SP, Vickson RG (1995) A survey of the maximum principles for optimal control problems with state constraints. *SIAM Rev* 37(2):181–218
- Huntington GT, Benson DA, Rao AV (2007) Optimal configuration of tetrahedral spacecraft formations. *J Aeronaut Sci* 55(2):141–169
- Jensen M, Smears I (2013) On the convergence of finite element methods for Hamilton–Jacobi–Bellman equations. *SIAM J Numer Anal* 51(1):137–162
- Kang W, Wilcox LC (2017) Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and HJB equations. *Comput Optim Appl* 68(2):289–315
- Kushner HJ, Dupuis P (2001) Numerical methods for stochastic control problems in continuous time. Springer, New York
- Maurer H (1977) On optimal control problems with bounded state variables and control appearing linearly. *SIAM J Control Optim* 15(3):345–362
- McEneaney WM (2006) Max-plus methods in nonlinear control and estimation. Birkhäuser, Boston
- McEneaney WM, Kluberg J (2009) Convergence rate for a curse-of-dimensionality-free method for a class of HJB PDEs. *SIAM J Control Optim* 48(5):3052–3079
- Melikyan AA (1998) Generalized characteristics of first order PDEs: application in optimal control and differential games. Birkhäuser, Boston
- Michalska H, Mayne DQ (1993) Robust receding horizon control of constrained nonlinear systems. *IEEE Trans Automat Contr* 38(11):1623–1633
- Osher S, Fedkiw R (2003) Level set methods and dynamic implicit surfaces. Springer, New York
- Pesch HJ (1994) A practical guide to the solution of real-life optimal control problems. *Control Cybern* 23(1–2): 7–60
- Pontryagin LS, Boltyansky VG, Gamkrelidze RV, Mishchenko EF (1964) The mathematical theory of optimal processes. Macmillan, New York
- Rao AV (2010) A survey of numerical methods for optimal control. *Adv Astron Sci* 135(1):497–528
- Schättler H (2014) Optimal control with state space constraints. In: Baillieul J, Samad T (eds) Encyclopedia of systems and control. Springer, London
- Schättler H, Ledzewicz U (2015) Optimal control for mathematical models of cancer therapies: an application of geometric methods. Interdisciplinary applied mathematics, vol 42. Springer, New York
- Subbotin AI (1995) Generalized solutions of first-order PDEs: the dynamical optimization perspective. Birkhäuser, Boston
- Subbotina NN (2006) The method of characteristics for Hamilton–Jacobi equations and applications to dynamical optimization. *J Math Sci* 135(3):2955–3091
- Trélat E (2012) Optimal control and applications to aerospace: some results and challenges. *J Optim Theory Appl* 154(3):713–758

- Wang L (2009) Model predictive control system design and implementation using MATLAB. Springer, London
- Yegorov I, Dower P (2018) Perspectives on characteristics based curse-of-dimensionality-free numerical approaches for solving Hamilton–Jacobi equations. Appl Math Opt. To appear. <https://doi.org/10.1007/s00245-018-9509-6>
- Yong J, Zhou XY (1999) Stochastic controls: Hamiltonian systems and HJB equations. Springer, New York

Circadian Rhythms Entrainment

► Control of Circadian Rhythms and Related Processes

Classical Frequency-Domain Design Methods

J. David Powell¹, Christina M. Ivler², and Abbas Emami-Naeini³

¹Aero/Astro Department, Stanford University, Stanford, CA, USA

²Shiley School of Engineering, University of Portland, Portland, OR, USA

³Electrical Engineering Department, Stanford University, Stanford, CA, USA

Abstract

The design of feedback control systems in industry is often carried out using frequency-response (FR) methods. Frequency-response design is used primarily because it provides good designs in the face of uncertainty in the plant model. For example, for systems with poorly known or changing high-frequency resonances, we can temper the feedback design to alleviate the effects of those uncertainties. This tempering can be carried out more easily using FR design without extensive modeling of the uncertainties. The method is most effective for systems that are stable in open loop; however, it can also be applied to systems

with instabilities. This article will introduce the reader to methods of design using lead and lag compensation. It will also cover the use of FR design to reduce steady-state errors and to improve robustness to uncertainties in high-frequency dynamics.

Keywords

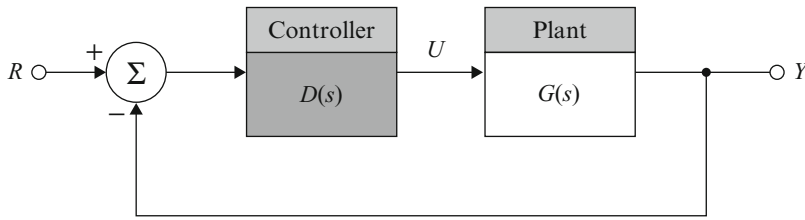
Frequency response · Magnitude · Phase · Bode plot · Stability · Bandwidth · Disturbance rejection bandwidth · Phase margin · Gain margin · Crossover frequency · Notch filter · Gain stabilization · Amplitude stabilization

Introduction

Finding an appropriate compensation ($D(s)$ in Fig. 1) using frequency response is probably the easiest of all feedback control design methods. Designs are achievable starting with the frequency response (FR) plots of both magnitude and phase of $G(s)$, then selecting $D(s)$ to achieve certain values of the gain and/or phase margins and system bandwidth or error characteristics. This article will cover the design process for finding an appropriate $D(s)$.

Design Specifications

As discussed in the ► “Frequency-Response and Frequency-Domain Models”, the **gain margin (GM)** is the factor by which the gain can be raised before instability results. The **phase margin (PM)** is the amount by which the phase of $D(j\omega)G(j\omega)$ exceeds -180° when $|D(j\omega)G(j\omega)| = 1$, the **Crossover Frequency**. Performance requirements for control systems are often partially specified in terms of PM and/or GM. For example, a typical specification might include the requirement that $PM > 50^\circ$ and $GM > 5$. It can be shown that the PM tends to correlate well with the damping ratio, ζ , of the closed-loop roots. In fact, it is shown in Franklin et al. (2019) that



Classical Frequency-Domain Design Methods, Fig. 1 Feedback system showing compensation, $D(s)$, where R is the command input, U is the control, and Y is the plant output. (Source: Franklin, Powell, Emami-Naeini,

Feedback Control of Dynamic Systems, 8th Ed., © 2019, p-280, Reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)

$$\zeta \cong \frac{PM}{100}$$

for many systems; however, the actual resulting damping and/or response overshoot of the final closed-loop system will need to be verified if they are specified as well as the PM. A PM of 50° would tend to yield a ζ of 0.5 for the closed-loop roots, which is a modestly damped system. The GM does not generally correlate directly with the damping ratio but is a measure of the degree of stability and is a useful secondary specification to ensure stability robustness.

Another design specification is the **bandwidth**, which is defined in the **Cross-Reference** section [1]. The bandwidth is a direct measure of the frequency at which the closed-loop system starts to fail in following the input command, responding with reduced magnitude and increasing phase lag. It is also a measure of the speed of response of a closed-loop system. Generally speaking, it correlates well with the step response and rise time of the system. A related design specification, the **disturbance rejection bandwidth**, also defined in the [► Frequency-Response and Frequency-Domain Models](#), evaluates the ability of the control system to reject disturbances. It signifies the upper limit of the disturbance frequency where the control system will reject its effect.

In some cases, the **steady-state error** must be less than a certain amount. As discussed in Franklin et al. (2019), the steady-state error is a direct function of the low-frequency gain of the FR magnitude plot. However, increasing the low-frequency gain typically will raise the entire mag-

nitude plot upward, thus increasing the magnitude 1 (0 db) crossover frequency and, therefore, increasing the speed of response, bandwidth, and disturbance rejection bandwidth of the system.

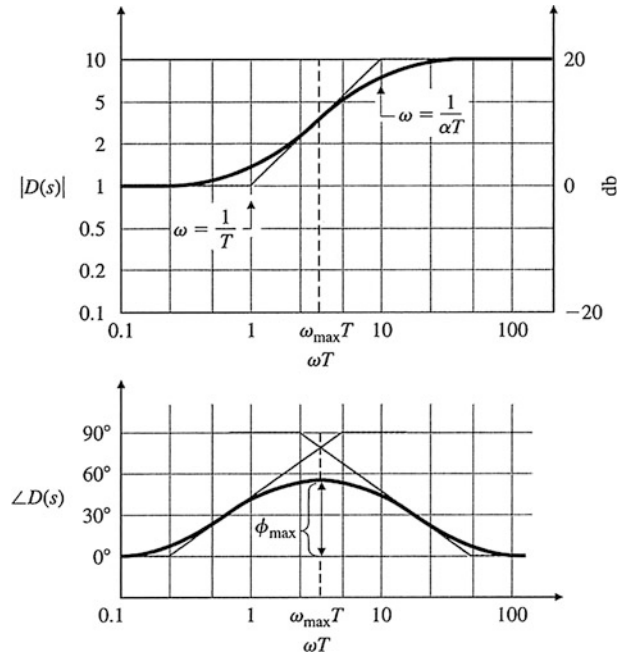
Compensation Design

In some cases, the design of a feedback compensation can be accomplished by using proportional control only, i.e., setting $D(s) = K$ (see Fig. 1) and selecting a suitable value for K . This can be accomplished by plotting the magnitude and phase of $G(s)$, looking at $|G(j\omega)|$ at the frequency where $\angle G(j\omega) = -180^\circ$ and then selecting K so that $|KG(j\omega)|$ yields the desired GM. Similarly, if a particular value of PM is desired, one can find the frequency where $\angle G(j\omega) = -180 + \text{the desired PM}$. The value of $|KG(j\omega)|$ at that frequency must equal 1; therefore, the value of $|G(j\omega)|$ must equal $1/K$. Note that the $|KG(j\omega)|$ curve moves vertically based on the value of K ; however the $\angle KG(j\omega)$ curve is not affected by the value of K . This characteristic simplifies the design process.

In more typical cases, proportional feedback alone is not sufficient. There is a need for a certain damping (i.e., PM) and/or speed of response (i.e., bandwidth), and there is no value of K that will meet the specifications. Therefore, some increased damping from the compensation is required. Likewise, a certain steady-state error requirement and its resulting low-frequency gain will cause the $|D(j\omega)G(j\omega)|$ to be greater than desired for an acceptable PM, so more phase lead is required from the compensation. This is

Classical Frequency-Domain Design Methods,

Fig. 2 Lead-compensation frequency response with $1/\alpha = 10$, $K = 1$. (Source: Franklin, Powell, Emami-Naeini, Feedback Control of Dynamic Systems, 8th Ed., © 2019, p-388 Reprinted by permission of Pearson Education, Inc.)



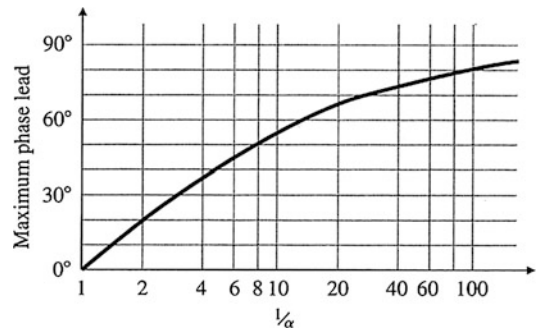
typically accomplished by **lead compensation**. A phase increase (or lead) is accomplished by placing a zero in $D(s)$. However, that alone would cause an undesirable high-frequency gain which would amplify noise; therefore, a first-order pole is added in the denominator at frequencies substantially higher than the zero break point of the compensator. Thus the phase lead still occurs, but the amplification at high frequencies is limited. The resulting lead compensation has a transfer function of the form

$$D(s) = K \frac{Ts + 1}{\alpha Ts + 1}, \quad \alpha < 1, \quad (1)$$

where $1/\alpha$ is the ratio between the pole/zero break-point frequencies. Figure 2 shows the frequency response of this lead compensation. The maximum amount of phase lead supplied is dependent on the ratio of the pole to zero and is shown in Fig. 3 as a function of that ratio.

For example, a lead compensator with a zero at $s = -2$ ($T = 0.5$) and a pole at $s = -10$ ($\alpha T = 0.1$) (and thus $\alpha = \frac{1}{5}$) would yield the maximum phase lead of $\phi_{\max} = 40^\circ$, where

$$\sin(\phi_{\max}) = \frac{1 - \alpha}{1 + \alpha}.$$



Classical Frequency-Domain Design Methods,

Fig. 3 Maximum phase increase for lead compensation. (Source: Franklin, Powell, Emami-Naeini, Feedback Control of Dynamic Systems, 8th Ed., © 2019, p-390, Reprinted by permission of Pearson Education, Inc.)

Note from the figure that we could increase the phase lead almost up to 90° using higher values of the **lead ratio**, $1/\alpha$; however, Fig. 2 shows that increasing values of $1/\alpha$ also produces higher amplifications at higher frequencies. Thus our task is to select a value of $1/\alpha$ that is a good compromise between an acceptable PM and acceptable noise sensitivity at high frequencies. Usually the compromise suggests that a lead compensation should contribute a maximum of 70° to the

phase. If a greater phase lead is needed, then a double-lead compensation would be suggested, where

$$D(s) = K \left(\frac{Ts + 1}{\alpha Ts + 1} \right)^2.$$

Even if a system had negligible amounts of noise present, the pole must exist at some point because of the impossibility of building a pure differentiator. No physical system—mechanical or electrical or digital—responds with infinite amplitude at infinite frequencies, so there will be a limit in the frequency range (or bandwidth) for which derivative information (or phase lead) can be provided.

As an example of designing a lead compensator, let us design compensation for a DC motor with the transfer function

$$G(s) = \frac{1}{s(s + 1)}.$$

We wish to obtain a steady-state error (e_{ss}) of less than 0.1 for a unit-ramp input, and we desire a system bandwidth greater than 3 rad/sec. Furthermore, we desire a PM of 45° . To accomplish the error requirement, Franklin et al., show that

$$e_{ss} = \lim_{s \rightarrow 0} s \left[\frac{1}{1 + D(s)G(s)} \right] R(s), \quad (2)$$

and if $R(s) = 1/s^2$ for a unit ramp, Eq. (2) reduces to

$$e_{ss} = \lim_{s \rightarrow 0} \left\{ \frac{1}{s + D(s)[1/(s + 1)]} \right\} = \frac{1}{D(0)}.$$

Therefore, we find that $D(0)$, the steady-state gain of the compensation, cannot be less than 10 if it is to meet the error criterion, so we pick $K = 10$. The frequency response of $KG(s)$ in Fig. 4 shows that the PM = 20° if no phase lead is added by compensation. If it were possible to simply add phase without affecting the magnitude, we would need an additional phase of only 25° at the $KG(s)$ crossover frequency of $\omega = 3$ rad/sec. However, maintaining the same low-frequency gain and adding a compensator zero will increase the crossover frequency;

hence more than a 25° phase contribution will be required from the lead compensation. To be safe, we will design the lead compensator so that it supplies a maximum phase lead of 40° . Figure 3 shows that $1/\alpha = 5$ will accomplish that goal. We will derive the greatest benefit from the compensation if the maximum phase lead from the compensator occurs at the crossover frequency. With some trial and error, we determine that placing the zero at $\omega = 2$ rad/sec and the pole at $\omega = 10$ rad/sec causes the maximum phase lead to be at the crossover frequency. The compensation, therefore, is

$$D(s) = 10 \frac{s/2 + 1}{s/10 + 1}.$$

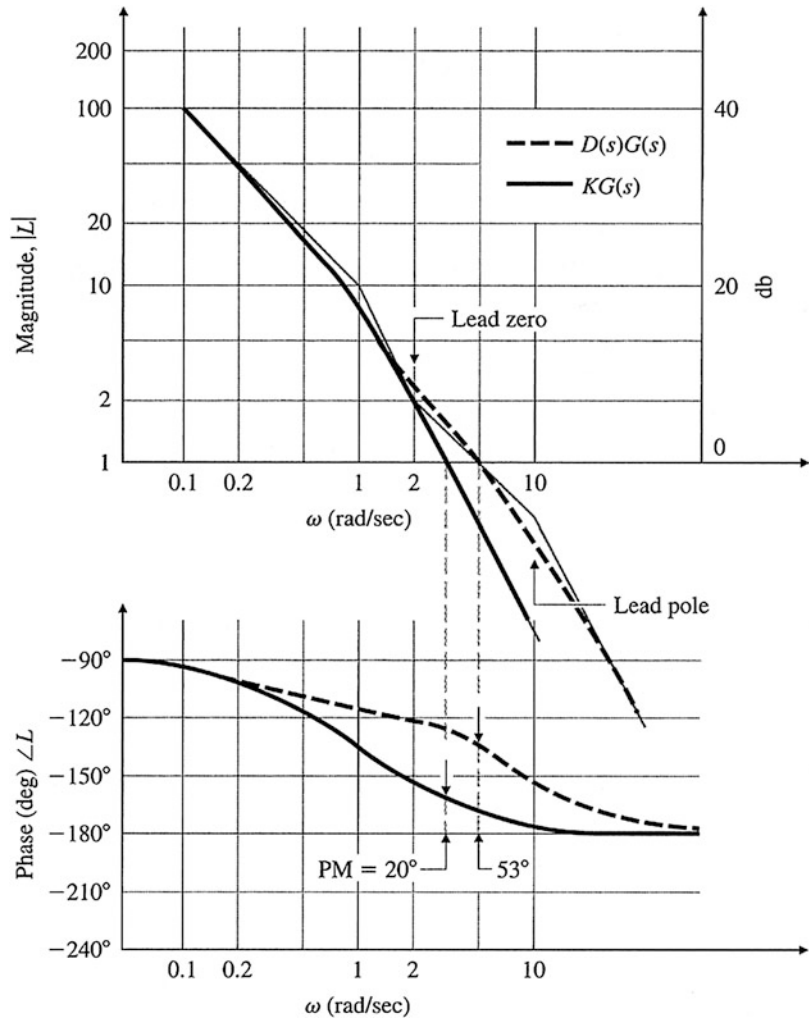
The frequency-response characteristics of $L(s) = D(s)G(s)$ in Fig. 4 can be seen to yield a PM of 53° , which satisfies the PM and steady-state error design goals. In addition, the crossover frequency of 5 rad/sec will also yield a bandwidth greater than 3 rad/sec, as desired.

Lag compensation is the same form as the lead compensation in Eq. (1) except that $\alpha > 1$. Therefore, the pole is at a lower frequency than the zero, and it produces higher gain at lower frequencies. The compensation is used primarily to reduce steady-state errors by raising the low-frequency gain, but without increasing the crossover frequency and speed of response. This can be accomplished by placing the pole and zero of the lag compensation well below the crossover frequency. Alternatively, lag compensation can also be used to improve the PM by keeping the low frequency gain the same but reducing the gain near crossover, thus reducing the crossover frequency. That will usually improve the PM since the phase of the uncompensated system typically is higher at lower frequencies.

Systems being controlled often have high-frequency dynamic phenomena, such as mechanical resonances, that could have an impact on the stability of a system. In very-high-performance designs, these high-frequency dynamics are included in the plant model, and a compensator is designed with a specific knowledge of those dynamics. However, a more **robust** approach

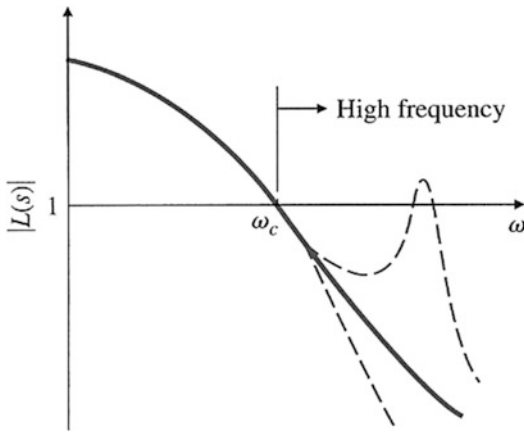
Classical Frequency-Domain Design Methods,

Fig. 4 Frequency response for lead-compensation design. (Source: Franklin, Powell, Emami-Naeini, *Feedback Control of Dynamic Systems*, 8th Ed., © 2019, p-391, Reprinted by permission of Pearson Education, Inc.)



for designing with uncertain high-frequency dynamics is to keep the high-frequency gain low, just as we did for sensor-noise reduction. The reason for this can be seen from the gain-frequency relationship of a typical system, shown in Fig. 5. The only way instability can result from high-frequency dynamics is if an unknown high-frequency resonance causes the magnitude to rise above 1 (0 db). Conversely, if all unknown high-frequency phenomena are guaranteed to remain below a magnitude of 1 (0 db), stability can be guaranteed. The likelihood of an unknown resonance in the plant G rising above 1 (0 db) can be reduced if the nominal high-frequency loop gain (L) is lowered by the addition of extra poles

in $D(s)$, such as in a low-pass filter. When the stability of a system with resonances is assured by tailoring the high-frequency magnitude never to exceed 1 (0 db), we refer to this process as **amplitude** or **gain stabilization**. Of course, if the resonance characteristics are known exactly and remain the same under all conditions, a specially tailored compensation, such as a **notch filter** at the resonant frequency, can be used to tailor the phase for stability even though the amplitude does exceed magnitude 1 (0 db) as explained in Franklin et al. (2019). Design of a notch filter is more easily carried out using **root locus** or **state space** design methods, all of which are discussed in Franklin et al. (2019). This



Classical Frequency-Domain Design Methods,
Fig. 5 Effect of high-frequency plant uncertainty.
 (Source: Franklin, Powell, Emami-Naeini, Feedback
 Control of Dynamic Systems, 8th Ed., © 2019, p-412,
 Reprinted by permission of Pearson Education, Inc.)

method of stabilization is referred to as **phase stabilization**. A drawback to phase stabilization is that the resonance information is often not available with adequate precision or varies with time; therefore, the method is more susceptible to errors in the plant model used in the design. Thus, we see that sensitivity to plant uncertainty and sensor noise are both reduced by sufficiently low gain at high frequency.

Summary and Future Directions

Before the common use of computers in design, frequency-response design was the most widely used method. While it is still the most widely used method for routine designs, complex systems, and their designs are being carried out using a multitude of methods. This section has introduced the classical methods used before and after the widespread use of computers in design. Future designs in the frequency domain will most likely capitalize on the capability of computers to balance many objectives in design by the optimization of various cost functions. This approach is discussed in ► [Control System Optimization Methods in the Frequency Domain](#)

Cross-References

- [Control System Optimization Methods in the Frequency Domain](#)
- [Frequency-Response and Frequency-Domain Models](#)
- [Polynomial/Algebraic Design Methods](#)
- [Quantitative Feedback Theory](#)
- [Spectral Factorization](#)

Bibliography

- Franklin GF, Powell JD, Workman M (1998) Digital control of dynamic systems, 3rd edn. Ellis-Kagle Press, Half Moon Bay
- Franklin GF, Powell JD, Emami-Naeini A (2019) Feedback control of dynamic systems, 8th edn. Pearson, Upper Saddle River

Computational Complexity in Robustness Analysis and Design

Onur Toker

Electrical and Computer Engineering, Florida Polytechnic University, Lakeland, FL, USA

Abstract

For many engineering analysis and design problems, one can think about various different types of algorithms with quite different features. Depending on the specific application, “general”, “practical”, and even “optimal” may have different meanings. In this chapter, we will be following the standards defined in the theory of computation, (Garey, Johnson (1979) Computers and intractability, a guide to the theory of NP-completeness. W. H. Freeman, San Francisco; Papadimitriou (1995) Computational complexity. Addison-Wesley/Longman, Reading), but we also would like to present an engineering interpretation of these concepts. First of all, an algorithm is considered as

“general”, if it really provides a correct answer for all cases of the problem. Although this makes perfect sense from a scientific perspective, it may be too restrictive from an engineering/economical viewpoint. For example, a technique which can be used for most cases may also be considered as “general” enough. An algorithm is considered as “practical” if its execution time is reasonable. From a scientific perspective, the worst case execution time looks more appealing, but for many engineering projects average execution time may be more meaningful. Similarly, there is no universal definition for reasonable execution time, but most scientists and engineers agree that polynomial execution time is reasonable, and anything beyond that is not. In this entry, we will look at some of the classical robust control problems, and try to present an engineering interpretation of the relevant computational complexity results.

Keywords

Approximation complexity · Computational complexity · *NP*-complete · *NP*-hard · Unsolvability

Introduction

Engineering literature is full of complex and challenging analysis and design problems, and we are usually interested in finding general, practical, and optimal solution techniques. Some of these are decision problems where the answer is either YES or NO, but there are also more involved design problems where the output of the procedure is expected to be something more complicated than a simple Boolean value. Therefore, we need to clarify what is an algorithm and provide a rigorous definition consistent with our intuitive understanding. Although the theory of computation provides mathematically rigorous definitions for a computer, and an algorithm, we may still have disagreements on certain details, for example, whether probabilistic methods should

be allowed or not. To find a primitive root in $\mathbb{Z}_p$ (Knuth 1997), there are highly effective techniques which involve random number generation. One may ask whether a random number generation-based set of rules should also be considered as an algorithm, provided that it will terminate almost surely after a few number of steps for all instances of the problem. In a similar fashion, one may ask whether any real number should be allowed as an input to an algorithm. Furthermore, one may also ask whether all calculations should be limited to algebraic functions only or should exact calculation of a limited set of non-algebraic functions, e.g., trigonometric functions, the gamma function, etc. should be considered as acceptable in an algorithm. Although all of these look reasonable options, from a rigorous point of view, they are different notions. In the following, we will be adopting the standard definitions of the theory of computation.

Turing Machines

Alan M. Turing (Turing 1936) defined a “hypothetical computation machine” to formally define the notions of algorithm and computability. A Turing machine is, in principle, quite similar to today’s digital computers. An algorithm is a Turing machine with a program, which is guaranteed to terminate after finitely many steps. These are basically standard definitions in the theory of computation, but alternative more relaxed definitions are possible too.

Depending on new scientific, engineering, and technological developments, superior computation machines may be constructed. For example, there is no guarantee that quantum computing research will not lead to superior computation machines (Chen et al. 2006; Kaye et al. 2007). In the future, the engineering community may feel the need to revise formal definitions of computer, algorithm, computability, tractability, etc., if such superior computation machines can be constructed and used for scientific/engineering applications.

Mathematical definition and basic properties of Turing machines are discussed in Garey and Johnson (1979), Papadimitriou (1995),

Hopcroft et al. (2001), Lewis and Papadimitriou (1998) and Sipser (2006). Although the original mathematical definition refers to a quite simple and low-performance “hardware” compared to today’s much higher-performance devices, the following two observations justify their use in the study of computational complexity of engineering problems. Anything which can be solved on today’s current digital computers can be solved on a Turing machine. Furthermore, a polynomial-time algorithm on today’s digital computers will correspond to again a polynomial-time algorithm on the original Turing machine, and vice versa.

When we use the term polynomial-time, we indeed mean polynomial-time in problem size. The notion of “problem size” is an important detail. For example, consider the problem of deciding whether an $n \times n$ matrix is singular or not. What is the most natural or meaningful definition for the problem size? If we decide to use IEEE floating formats, and simply ignore round of errors, problem size can be defined as n . But from a scientific perspective, ignoring these round of errors may not be acceptable, and we may be forced to define the problem size as the total number of bits required to represent the input. No matter which encoding is used, there will be always numbers which cannot be represented precisely, and in that case, because of being unable to present the input to the computer, we may not even talk about solution algorithms. A more interesting case is a 2×2 matrix with entries requiring very large number of bits to encode. In that case, the theory of computation suggests a large value for the problem size, which may or may not be acceptable from an engineering perspective. Although the whole theory is based on rigorous mathematical definitions and theories, sometimes certain definitions may look a bit counterintuitive. Knowing all of these details is essential for proper interpretation of the computational complexity results.

Unsolvability

For certain engineering problems, it can be shown that there can be no algorithm which can provide the correct answer for all possible cases. Such

problems are called unsolvable. The condition “all cases” may be considered as too restrictive from an engineering perspective, and one may argue that such negative results have only theoretical importance with limited or no practical implications. But such results do imply that we should concentrate our efforts on alternative research directions, for example, development of algorithms which are guaranteed to work for cases which appear more frequently in real scientific/engineering applications.

Hilbert’s tenth problem is basically about finding an algorithm for testing whether a Diophantine equation has an integer solution. In 1970, Matijasevich showed that there can be no algorithm (Matijasevich 1993) which can handle all possible cases. Therefore, we say that the problem of checking whether a Diophantine equation has an integer solution is unsolvable. For certain special cases, solution algorithms do exist, but unfortunately none of these can be extended to a general algorithm.

In robust control theory, there are certain unsolvable decision problems for systems with time-varying parametric uncertainty. Most of these can be reduced to what is known as the Post correspondence problem (Davis 1985) and the embedding of free semigroups into matrices. In (Blondel and Tsitsiklis 2000a), it was proved that the problem of checking the boundedness of switching systems of type

$$x(k+1) = A_{f(k)}x(k),$$

where f is an arbitrary unknown function from $\mathbb{N}$ into $\{0, 1\}$, is unsolvable. This is related to a time-varying parametric uncertainty problem with system dynamics described by

$$x(k+1) = (c_k A_0 + (1 - c_k) A_1)x(k), \quad (1)$$

with c_k being the time-varying uncertain parameter in $[0, 1]$, and we are only interested in boundedness of the state vector. Stated in a more formal way:

Bounded state for time-varying parametric uncertainty problem (BSTVP) Consider the

dynamical system defined in (1) with the time-varying uncertain parameter c_k in $[0, 1]$. The problem is to test whether the state vector x remains bounded.

Results proved in Blondel and Tsitsiklis (2000a) imply that the BSTVP is an unsolvable problem. A related problem is asymptotic stability against time-varying parametric uncertainty, more precisely whether the state vector converges to zero.

Asymptotic stability for time-varying parametric uncertainty problem (ASTVP) Consider the dynamical system defined in (1) with the time-varying uncertain parameter c_k in $[0, 1]$. The problem is to test whether the state vector x converges to zero, i.e., checking whether $\lim_{k \rightarrow \infty} \|x(k)\| = 0$.

This is known to be related to the joint spectral radius (JSR) (Rota and Strang 1960), indeed equivalent to whether the JSR of $\{A_0, A_1\}$ is less than one. In other words, checking whether JSR is less than one is a kind of robust stability test against time-varying parametric uncertainty. For a long period of time, a simple procedure based on what is known as the finiteness conjecture (FC) (Lagarias and Wang 1995) was considered as an algorithm candidate for testing robust stability. Till the 2000s, there was no known counter example for the finiteness conjecture, and a significant number of scientists and engineers considered it as true. FC may be interpreted as “For asymptotic stability of $x(k+1) = A_{f(k)}x(k)$ type switching systems, it is enough to consider periodic switchings.” The truth of this FC conjecture would imply existence of an algorithm for the ASTVP problem. However, it was shown in Bousch and Mairesse (2002) that FC is not true (see Blondel et al. (2003) for a significantly simplified proof). To the best of the author’s knowledge, whether ASTVP is solvable or not is an open problem. However, there are several very useful results about computing and/or approximating the joint spectral radius. See Blondel and Nesterov (2005, 2009), Protasov et al. (2010), and Chang and Blondel (2013) and references therein for details. Despite all of these strong results, existence of an algorithm to test whether

JSR is less than one remains as an open problem. For further results on unsolvability in robust control, see Blondel et al. (1999) and Blondel and Megretski (2004) and references therein.

NP-Hardness and NP-Completeness

Solvability of an engineering problem, more precisely a decision problem, is related to whether there exists an algorithm which is guaranteed to terminate in a finite amount of time. But the execution time of the algorithm is as important as its existence. Consider a graph theoretic problem, let n be the number of nodes, and consider an algorithm which requires 2^n steps on a Turing machine or a digital computer. Such algorithms with execution times growing faster than polynomial are considered as inefficient. On the other hand, algorithms with worst-case execution times bounded by a polynomial of the problem size are considered as efficient.

One may argue that bounding the worst-case execution time by a polynomial of the problem size may be too restrictive for certain applications. Indeed, depending on the nature of the specific application, being polynomial-time on the average may be good enough. The same may be true for algorithms which are polynomial-time almost surely, or for “most” of the cases. However, the standard computational complexity theory is developed around this idea of being polynomial-time even in the worst case (Garey and Johnson 1979; Papadimitriou 1995). Understanding this detail is also essential for proper interpretation of complexity results.

To explain NP-hardness, NP-completeness, and tractability, we need to define certain sets. The class P corresponds to decision problems which can be solved on a Turing machine with execution time bounded by a polynomial of the problem size (Garey and Johnson 1979). In short, such problems are categorized as decision problems which have polynomial-time solution algorithms. The definition of the class NP is more technical and involves nondeterministic Turing machines (Garey and Johnson 1979). Nondeterministic Turing machines can be considered as computers executing one of the

finitely many instructions on a random basis. The class NP may also be interpreted as the class of decision problems for which the truth of the problem can be verified in polynomial-time. Nondeterministic Turing machines may not have a major practical use, but they are quite important from a theoretical perspective.

It is currently unknown whether P and NP are equal or not. This is a major open problem, and the importance of it in the theory of computation is comparable to the importance of the Riemann hypothesis in number theory. Indeed both are among unsolved Millennium Prize Problems.

A problem is NP -complete if it is NP and every NP problem polynomially reduces to it (Garey and Johnson 1979). When a problem is proved to be NP -complete, it means that finding a polynomial-time solution algorithm is not possible, provided $P \neq NP$. There are several NP -complete problems for which there is no known polynomial-time algorithm. Despite significant amount of research, there is no known polynomial-time algorithm for any of the NP -complete problems. This *may* be considered as indication of $NP \not\subseteq P$. Indeed, finding a polynomial-time solution algorithm for any of the NP -complete problems will automatically imply existence of polynomial-time solution algorithms for all NP -complete problems, and this will be a major breakthrough if it is indeed true. Although current evidence is more toward $P \neq NP$, this does not constitute a formal proof.

Being NP -complete is considered as being intractable, and this whole argument may collapse if P and NP turns out to be equal. However, under the assumption of $NP \not\subseteq P$, a problem being NP -complete does imply impossibility of finding polynomial-time solution algorithms. Again, the standard theory of computation requires algorithms to handle *all* cases properly, execution time to be *always* bounded by a polynomial of the problem size, and the definition of problem size may look a bit counterintuitive in certain cases. These details and assumptions are important for proper interpretation of the complexity results.

A problem is called NP -hard if and only if there is an NP -complete problem which is

polynomial-time reducible to it (Garey and Johnson 1979). Being NP -hard is also considered as being intractable, because unless $P = NP$, no polynomial-time solution algorithm can be found. All NP -complete problems are also NP -hard.

Approximation-Based Techniques

NP -hardness and NP -completeness are originally defined to express the inherent difficulty of decision-type problems. But there are several very important engineering problems which are of approximation type; see Papadimitriou (1995). Most of the robust stability tests can be formulated as “Check whether $\gamma < 1$,” which is basically a decision-type problem. An approximate value of γ may not be always good enough to “solve” the problem. For a robust stability analysis problem, a result like $\gamma = 0.99 \pm 0.02$ may be not enough to decide about robust stability, although the approximation error is about 2% only. However, for certain other engineering applications, including robustness margin estimation, approximate values may be good enough. Namely, from an engineering perspective, a good enough approximation of γ may be as good as the exact value. Exact version may be NP -hard, but this does not necessarily imply that the approximation version will be NP -hard too. If the exact version is not tractable, and the approximation version makes sense from an engineering perspective, it may be worth to consider the approximation version and try to find algorithms for this relaxed version.

Basic Results

The first known NP -complete problem is SATISFIABILITY (Cook 1971). In this problem, there is a single Boolean equation with several variables, and we would like to test whether there exists an assignment to these variables which makes the Boolean expression true. NP -completeness of SATISFIABILITY enabled proofs of several other NP -completeness results from graph theory to quadratic optimization.

Quadratic programming is an important problem by itself, but it is also related several important robust stability against parametric

uncertainty-type problems. The quadratic programming (QP) can be defined as

$$q = \min_{Ax \leq b} x^T Qx + c^T x,$$

and the decision version is checking whether $q < 1$. When the matrix Q is positive definite, convex optimization techniques can be used; however, if the matrix Q is not positive definite, then the problem is known to be *NP*-hard.

A related problem is linear programming (LP)

$$q = \min_{Ax \leq b} c^T x,$$

which is used in certain robust control problems (Dahleh and Diaz-Bobillo 1994). The simplex method (Dantzig 1963) is a well-known computational technique for linear programming, and it has polynomial-time complexity on the “average” (Smale 1983). However, there are examples where the simplex method requires execution time which is an exponential function of the problem size (Klee and Minty 1972). In 1979, Khachiyan proposed the ellipsoid algorithm for linear programming, which was the first known polynomial-time approximation algorithm (Schrijver 1998) for linear programming. It is possible to answer the decision version, i.e., checking whether q is less than 1, in polynomial-time too. This is possible by using the ellipsoid algorithm for approximation and stopping when the error is below a certain level. But all of these results are for standard Turing machines with input parameters restricted to rational numbers. An interesting open problem is whether LP admits a polynomial-time algorithm in the real number model of computation.

Decision version of QP is *NP*-hard, but approximation of QP is quite “difficult” as well. An approximation is called a μ -approximation if the absolute value of the error is bounded by μ times the absolute value of max-min of the function. The following is a classical result on QP (Bellare and Rogaway 1995): Unless $P = NP$, QP does not admit a polynomial-time μ -approximation algorithm even for $\mu < 1/3$.

This is sometimes informally stated as “QP is *NP*-hard to approximate.”

Computational Complexity of Robust Control Problems

Our discussion of computational complexity in robust control will be limited to interval matrices, time-invariant parametric uncertainty, and the structured singular value. For other robust control problems, please see Blondel and Tsitsiklis (2000b), Blondel et al. (1999), Blondel and Megretski (2004), and references therein.

Kharitonov theorem is about robust Hurwitz stability of polynomials with coefficients restricted to intervals (Kharitonov 1978). In its original formulation, each coefficient is considered as a separate time-invariant uncertain parameter. This version of the problem has a very simple and elegant solution (see Cross-References). If algebraic dependencies between coefficients are allowed, then the problem will be *NP*-hard. To understand why, we need to look at interval matrix problem. A set of matrices defined by the equation,

$$\mathcal{A} = \{A \in \mathbb{R}^{n \times n} : \alpha_{i,j} \leq A_{i,j} \leq \beta_{i,j}, \\ i, j = 1, \dots, n\}, \quad (2)$$

where $\alpha_{i,j}, \beta_{i,j} \in \mathbb{R}$ for $i, j = 1, \dots, n$, is called an interval matrix. Consider the mathematical model of a complex system with multiple physical parameters. If we have only partial information about each physical parameter, then it may be possible to represent system dynamics as

$$\dot{x} = Mx + Bu$$

where M is some unknown but fixed matrix in an interval matrix $\mathcal{A}$. In this case, robust stability will be equivalent to Hurwitz stability of all matrices in $\mathcal{A}$. The following two stability problems about interval matrices are *NP*-hard:

Interval Matrix Problem 1 (IMP1): Decide whether a given interval matrix, $\mathcal{A}$, is robust Hurwitz stable or not. Namely, check whether all members of $\mathcal{A}$ are Hurwitz-stable matrices, i.e., all eigenvalues are in open left half plane.

Interval Matrix Problem 2 (IMP2): Decide whether a given interval matrix, $\mathcal{A}$, has a Hurwitz-stable matrix or not. Namely, check whether there exists at least one matrix in $\mathcal{A}$ which is Hurwitz stable.

For a proof of NP -hardness of IMP1, see Poljak and Rohn (1993) and Nemirovskii (1993), and for a proof of IMP2, see Blondel and Tsitsiklis (1997).

For interconnected systems with component-level uncertainties (Packard and Doyle 1993), robust stability can be checked by using structured singular values. Basically, if we do know bounds on the component-level uncertainties, robust stability can be tested by checking whether a structured singular value is less than 1 or not. However, this problem is known to be NP -hard.

Structured Singular Value Problem (SSVP):

Given a matrix M and uncertainty structure Δ , check whether the structured singular value $\mu_{\Delta}(M) < 1$.

This is proved to be NP -hard for real, and mixed, uncertainty structures (Braatz et al. 1994), as well as for complex uncertainties with no repetitions (Toker and Ozbay 1996, 1998).

Complex structured singular problem value has a simpler convex relaxation. This is known as the standard upper bound $\bar{\mu}$, which is known to give quite tight approximations for most cases (Packard and Doyle 1993). But there are very limited results about $\bar{\mu}/\mu$ ratio.

Summary and Future Directions

For all of the NP -completeness and NP -hardness results presented here, to be able to say that there is no polynomial-time algorithm, we need the assumption “ $P \neq NP$ ”, which is not proved yet. That’s why this is a central question in the theory of computation. Researchers agree that really new and innovative tools are needed to study this problem. On one other extreme, one can question whether we can really say something about this problem within the Zermelo-Fraenkel (ZF) set theory or it will have a status similar to axiom

of choice (AC) and the continuum hypothesis (CH) where we can neither refute nor provide a proof (Aaronson 1995). Therefore, the question may be indeed much deeper, and standard axioms of today’s mathematics may not be enough to provide an answer. As for any such problem, we can still hope that in the future, new “self-evident” axioms may be discovered, and with the help of them, we may provide an answer.

All of the complexity results mentioned here are with respect to the standard Turing machine which is a simplified model of today’s digital computers. Depending on the progress in science, engineering, and technology, if superior computation machines can be constructed, then some of the abovementioned problems can be solved much faster on these devices, and current results/problems of the theory of computation may no longer be of great importance or relevance for engineering and scientific applications. In such a case, definitions of algorithm, tractable, etc. may need to be revised according to these new devices.

For NP -hard robust control problems, various relaxations may be considered. For example, approximation with a bounded error may be one alternative. Similarly, algorithms which are polynomial-time on the average or polynomial-time for most of cases can be considered as potential alternatives. In summary, computational complexity theory guides research on the development of algorithms, indicating which directions are dead ends and which directions are worth to investigate. Complete understanding of the basic concepts of the theory of computation is very important for a proper engineering interpretation of these complexity results.

Cross-References

- [Optimization-Based Robust Control](#)
- [Robust Control in Gap Metric](#)
- [Robust Fault Diagnosis and Control](#)
- [Robustness Issues in Quantum Control](#)
- [Stability and Performance of Complex Systems Affected by Parametric Uncertainty](#)

► **Structured Singular Value and Applications:
Analyzing the Effect of Linear Time-Invariant
Uncertainty in Linear Systems**

Bibliography

- Aaronson S (1995) Is P versus NP formally independent? Technical report 81, EATCS
- Bellare M, Rogaway P (1995) The complexity of approximating a nonlinear program. *Math Program* 69: 429–441
- Blondel VD, Megretski A (2004) *Unsolved problems in mathematical systems and control theory*. Princeton University Press, Princeton
- Blondel VD, Nesterov Y (2005) Computationally efficient approximations of the joint spectral radius. *SIAM J Matrix Anal* 27:256–272
- Blondel VD, Nesterov Y (2009) Polynomial-time computation of the joint spectral radius for some sets of nonnegative matrices. *SIAM J Matrix Anal* 31:865–876
- Blondel VD, Tsitsiklis JN (1997) NP-hardness of some linear control design problems. *SIAM J Control Optim* 35:2118–2127
- Blondel VD, Tsitsiklis JN (2000a) The boundedness of all products of a pair of matrices is undecidable. *Syst Control Lett* 41:135–140
- Blondel VD, Tsitsiklis JN (2000b) A survey of computational complexity results in systems and control. *Automatica* 36:1249–1274
- Blondel VD, Sontag ED, Vidyasagar M, Willems JC (1999) *Open problems in mathematical systems and control theory*. Springer, London
- Blondel VD, Theys J, Vladimirov AA (2003) An elementary counterexample to the finiteness conjecture. *SIAM J Matrix Anal* 24:963–970
- Bousch T, Mairesse J (2002) Asymptotic height optimization for topical IFS, Tetris heaps and the finiteness conjecture. *J Am Math Soc* 15:77–111
- Braatz R, Young P, Doyle J, Morari M (1994) Computational complexity of μ calculation. *IEEE Trans Autom Control* 39:1000–1002
- Chang CT, Blondel VD (2013) An experimental study of approximation algorithms for the joint spectral radius. *Numer Algorithms* 64:181–202
- Chen G, Church DA, Englert BG, Henkel C, Rohwedder B, Scully MO, Zubairy MS (2006) *Quantum computing devices*. Chapman and Hall/CRC, Boca Raton
- Cook S (1971) The complexity of theorem proving procedures. In: *Proceedings of the third annual ACM symposium on theory of computing*, Shaker Heights, pp 151–158
- Dahleh MA, Diaz-Bobillo I (1994) *Control of uncertain systems*. Prentice Hall, Englewood Cliffs
- Dantzig G (1963) *Linear programming and extensions*. Princeton University Press, Princeton
- Davis M (1985) *Computability and unsolvability*. Dover, New York
- Garey MR, Johnson DS (1979) *Computers and intractability, a guide to the theory of NP-completeness*. W. H. Freeman, San Francisco
- Hopcroft JE, Motwani R, Ullman JD (2001) *Introduction to automata theory, languages, and computation*. Addison Wesley, Boston
- Kaye P, Laflamme R, Mosca M (2007) *An introduction to quantum computing*. Oxford University Press, Oxford
- Kharitonov VL (1978) Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. *Differ Uravn* 14:2086–2088
- Klee V, Minty GJ (1972) How good is the simplex algorithm? In: *Inequalities III (proceedings of the third symposium on inequalities)*, Los Angeles. Academic, New York/London, pp 159–175
- Knuth DE (1997) *Art of computer programming, volume 2: seminumerical algorithms*, 3rd edn. Addison-Wesley, Reading
- Lagarias JC, Wang Y (1995) The finiteness conjecture for the generalized spectral radius of a set of matrices. *Linear Algebra Appl* 214:17–42
- Lewis HR, Papadimitriou CH (1998) *Elements of the theory of computation*. Prentice Hall, Upper Saddle River
- Matiyasevich YV (1993) Hilbert's 10th Problem. MIT, ISBN: 9780262132954. <https://mitpress.mit.edu/books/hilberts-10th-problem>
- Nemirovskii A (1993) Several NP-hard problems arising in robust stability analysis. *Math Control Signals Syst* 6:99–105
- Packard A, Doyle J (1993) The complex structured singular value. *Automatica* 29:71–109
- Papadimitriou CH (1995) *Computational complexity*. Addison-Wesley/Longman, Reading
- Poljak S, Rohn J (1993) Checking robust nonsingularity is NP-hard. *Math Control Signals Syst* 6:1–9
- Protasov V, Jungers RM, Blondel VD (2010) Joint spectral characteristics of matrices: a conic programming approach. *SIAM J Matrix Anal Appl* 31:2146–2162
- Rota GC, Strang G (1960) A note on the joint spectral radius. *Proc Neth Acad* 22:379–381
- Schrijver A (1998) *Theory of linear and integer programming*. Wiley, Chichester
- Sipser M (2006) *Introduction to the theory of computation*. Thomson Course Technology, Boston
- Smale S (1983) On the average number of steps in the simplex method of linear programming. *Math Program* 27:241–262
- Toker O, Ozbay H (1996) Complexity issues in robust stability of linear delay differential systems. *Math Control Signals Syst* 9:386–400
- Toker O, Ozbay H (1998) On the NP-hardness of the purely complex μ computation, analysis/synthesis, and some related problems in multidimensional systems. *IEEE Trans Autom Control* 43:409–414
- Turing AM (1936) On computable numbers, with an application to the Entscheidungsproblem. *Proc Lond Math Soc* 42:230–265

Computer-Aided Control Systems Design: Introduction and Historical Overview

Andreas Varga
Gilching, Germany

Abstract

Computer-aided control system design (CACSD) encompasses a broad range of methods, tools, and technologies for system modelling, control system synthesis and tuning, dynamic system analysis and simulation, as well as validation and verification. The domain of CACSD enlarged progressively over decades from simple collections of algorithms and programs for control system analysis and synthesis to comprehensive tool sets and user-friendly environments supporting all aspects of developing and deploying advanced control systems in various application fields. This entry gives a brief introduction to CACSD and reviews the evolution of key concepts and technologies underlying the CACSD domain. Several cornerstone achievements in developing reliable numerical algorithms; implementing robust numerical software; and developing sophisticated integrated modelling, simulation, and design environments are highlighted.

Keywords

CACSD · Modelling · Numerical analysis · Simulation · Software tools

Introduction

To design a control system for a plant, a typical *computer-aided control system design* (CACSD) work flow comprises several interlaced activities.

Model building is often a necessary first step consisting in developing suitable mathematical models to accurately describe the

plant dynamical behavior. High-fidelity physical plant models obtained, for example, by using the first principles of modelling, primarily serve for analysis and validation purposes using appropriate simulation techniques. These dynamic models are usually defined by a set of *ordinary differential equations* (ODEs), *differential-algebraic equations* (DAEs), or *partial differential equations* (PDEs). However, for control system synthesis purposes, simpler models are required, which are derived by simplifying high-fidelity models (e.g., by linearization, discretization, or model reduction) or directly determined in a specific form from input-output measurement data using system identification techniques. Frequently used synthesis models are continuous or discrete-time *linear time-invariant* (LTI) models describing the nominal behavior of the plant in a specific operating point. The more accurate *linear parameter-varying* (LPV) models may serve to account for uncertainties due to various performed approximations, nonlinearities, or varying model parameters.

Simulation of dynamical systems is a closely related activity to modelling and is concerned with performing virtual experiments on a given plant model to analyze and predict the dynamic behavior of a physical plant. Often, modelling and simulation are closely connected parts of dedicated environments, where specific classes of models can be built and appropriate simulation methods can be employed. Simulation is also a powerful tool for the validation of mathematical models and their approximations. In the context of CACSD, simulation is frequently used as a control system tuning-aid, as, for example, in an optimization-based tuning approach using time-domain performance criteria.

System analysis and synthesis are concerned with the investigation of properties of the underlying synthesis model and the determination of a control system which fulfills basic requirements for the closed-loop controlled plant, such as stability or various time or frequency response requirements. The analysis also serves to check existence conditions for the solvability of synthesis problems, according to established design

methodologies. An important synthesis goal is the guarantee of the performance robustness. To achieve this goal, robust control synthesis methodologies often employ optimization-based parameter tuning in conjunction with worst-case analysis techniques. A rich collection of reliable numerical algorithms are available to perform such analysis and synthesis tasks. These algorithms form the core of CACSD, and their development represented one of the main motivations for CACSD-related research.

Performance robustness assessment of the resulting closed-loop control system is a key aspect of the verification and validation activity. For a reliable assessment, simulation-based worst-case analysis represents, often, the only way to prove the performance robustness of the synthesized control system in the presence of parametric uncertainties and variabilities.

Development of CACSD Tools

The development of CACSD tools for system analysis and synthesis started around 1960, immediately after general-purpose digital computers, and new programming languages became available for research and development purposes. In what follows, we give a historical survey of these developments in the main CACSD areas.

Modelling and Simulation Tools

Among the first developments were modelling and simulation tools for continuous-time systems described by differential equations based on dedicated simulation languages. Over 40 continuous system simulation languages had been developed as of 1974 (Nilsen and Karplus 1974), which evolved out of attempts at digitally emulating the behavior of widely used analog computers before 1960. A notable development in this period was the Continuous System Simulation Language (CSSL) standard (Augustin et al. 1967), which defined a system as an interconnection of blocks corresponding to operators which emulated the main analog simulation blocks and implied the integration of the underlying ODEs using suitable numerical

methods. For many years, the Advanced Continuous Simulation Language (ACSL) preprocessor to Fortran (Mitchel and Gauthier 1976) was one of the most successful implementations of the CSSL standard.

A turning point was the development of graphical user interfaces allowing graphical block diagram-based modelling. The most important developments were SystemBuild (Shah et al. 1985) and SIMULAB (later marketed as Simulink) (Grace 1991). Both products used a customizable set of block libraries and were seamlessly integrated in, respectively, MATRIXx and MATLAB, two powerful interactive matrix computation environments (see below). SystemBuild provided several advanced features such as event management, code generation, and DAE-based modelling and simulation. Simulink excelled from the beginning with its intuitive, easy-to-use user interface. Recent extensions of Simulink allow the modelling and simulation of hybrid systems, code generation for real-time applications, and various enhancements of the model building process (e.g., object-oriented modelling).

The object-oriented paradigm for system modelling was introduced with Dymola (Elmqvist 1978) to support physical system modelling based on interconnections of subsystems. The underlying modelling language served as the basis of the first version of Modelica (Elmqvist et al. 1997), a modern equation-based modelling language which was the result of a coordinated effort for the unification and standardization of expertise gained over many years with object-oriented physical modelling. The latest developments in this area are comprehensive model libraries for different application domains such as mechanical, electrical, electronic, hydraulic, thermal, control, and electric power systems. Notable commercial front ends for Modelica are Dymola, MapleSim, and System-Modeler, where the last two are tightly integrated in the symbolic computation environments Maple and Mathematica, respectively. Recent efforts toward extending the capabilities of Modelica, by simultaneously simplifying the modelling effort and enlarging its application area, are described

in Elmqvist et al. (2016), where the design of a new modelling language, called Modia, is presented. Modia is based on the Julia language (Bezanson et al. 2017) and relies on powerful DAE solvers.

Numerical Software Tools

The computational tools for CACSD rely on many numerical algorithms whose development and implementation in computer codes was the primary motivation of this research area since its beginnings. The Automatic Synthesis Program (ASP), developed in 1966 (Kalman and Englar 1966), was implemented in FAP (Fortran Assembly Program) and ran only on an IBM 7090–7094 machine. The Fortran II version of ASP (known as FASP) can be considered to be the first collection of computational CACSD tools ported to several mainframe computers. Interestingly, the linear algebra computations were covered by only three routines (diagonal decomposition, inversion, and pseudoinverse), and no routines were used for eigenvalue or polynomial roots computation. The main analysis and synthesis functions covered the sampled-data discretization (via matrix exponential), minimal realization, time-varying Riccati equation solution for quadratic control, filtering, and stability analysis. The FASP itself performed the required computational sequences by interpreting simple commands with parameters. The extensive documentation, containing a detailed description of algorithmic approaches and many examples, marked the starting point of an intensive research on algorithms and numerical software, which culminated in the development of the high-performance control and systems library SLICOT (Benner et al. 1999; Van Huffel et al. 2004). In what follows, we highlight the main achievements along this development process.

The direct successor of FASP is the Variable Dimension Automatic Synthesis Program (VASP) (implemented in Fortran IV on IBM 360) (White and Lee 1971), while a further development was ORACLS (Armstrong 1978), which included several routines from the newly developed eigenvalue package EISPACK (Smith et al. 1976; Garbow et al. 1977) as well as

solvers for linear (Lyapunov, Sylvester) and quadratic (Riccati) matrix equations. From this point, the mainstream development of numerical algorithms for linear system analysis and synthesis closely followed the development of algorithms and software for numerical linear algebra. A common feature of all subsequent developments was the extensive use of robust linear algebra software with the Basic Linear Algebra Subprograms (BLAS) (Lawson et al. 1979) and the Linear System Package (LINPACK) for solving linear systems (Dongarra et al. 1979). Several control libraries have been developed almost simultaneously, relying on the robust numerical linear algebra core software formed of BLAS, LINPACK, and EISPACK. Notable examples are RASP (based partly on VASP and ORACLS) (Grübel 1983) – developed originally by the University of Bochum and later by the German Aerospace Center (DLR); BIMAS (Varga and Sima 1985) and BIMASC (Varga and Davidoviciu 1986) – two Romanian initiatives; and SLICOT (van den Boom et al. 1991) – a Benelux initiative of several universities jointly with the Numerical Algorithms Group (NAG).

The last development phase was marked by the availability of the Linear Algebra Package (LAPACK) (Anderson et al. 1992), whose declared goal was to make the widely used EISPACK and LINPACK libraries run efficiently on shared memory vector and parallel processors. To minimize the development efforts, several active research teams from Europe started, in the framework of the NICONET project, a concentrated effort to develop a high-performance numerical software library for CACSD as a new significantly extended version of the original SLICOT. The goals of the new library were to cover the main computational needs of CACSD, by relying exclusively on LAPACK and BLAS, and to guarantee similar numerical performance as that of the LAPACK routines. The software development used rigorous standards for implementation in Fortran 77, modularization, testing, and documentation (similar to that used in LAPACK). The development of the latest versions of RASP and SLICOT eliminated practically any duplication of efforts and led to

a library which contained the best software from RASP, SLICOT, BIMAS, and BIMASC. The current version of SLICOT is fully maintained by the NICONET association (<http://www.niconet-ev.info/en/>) and serves as basis for implementing advanced computational functions for CACSD in interactive environments such as MATLAB (<https://www.mathworks.com>), Maple (<https://www.maplesoft.com/products/maple/>), Scilab (<https://www.scilab.org/>), and Octave (<https://www.gnu.org/software/octave/>).

Interactive Tools

Early experiments during 1970–1985 included the development of several interactive CACSD tools employing menu-driven interaction, question-answer dialogues, or command languages. The April 1982 special issue of IEEE Control Systems Magazine was dedicated to CACSD environments and presented software summaries of 20 interactive CACSD packages. This development period was marked by the establishment of new standards for programming languages (Fortran 77, C), availability of high-quality numerical software libraries (BLAS, EISPACK, LINPACK, ODEPACK), transition from mainframe computers to minicomputers, and finally to the nowadays-ubiquitous personal computers as computing platforms, spectacular developments in graphical display technologies, and application of sound programming paradigms (e.g., strong data typing).

A remarkable event in this period was the development of MATLAB, a command language-based interactive matrix laboratory (Moler 1980). The original version of MATLAB was written in Fortran 77. It was primarily intended as a student teaching tool and provided interactive access to selected subroutines from LINPACK and EISPACK. The tool circulated for a while in the public domain, and its high flexibility was soon recognized. Several CACSD-oriented commercial clones have been implemented in the C language, the most important among them being MATRIXx (Walker et al. 1982) and PC-MATLAB (Moler et al. 1985).

The period after 1985 until around 2000 can be seen as a consolidation and expansion period for

many commercial and noncommercial tools. In an inventory of CACSD-related software issued by the Benelux Working Group on Software (WGS) under the auspices of the IEEE Control Systems Society, there were in 1992 in active development 70 stand-alone CACSD packages, 21 tools based on or similar to MATLAB, and 27 modelling/simulation environments. It is interesting to look more closely at the evolutions of the two main players MATRIXx and MATLAB, which took place under harshly competitive conditions.

MATRIXx with its main components Xmath, SystemBuild, and AutoCode had over many years a leading role (especially among industrial customers), excelling with a rich functionality in domains such as system identification, control system synthesis, model reduction, modelling, simulation, and code generation. After 2003, MATRIXx (<https://www.ni.com/matrixx/>) became a product of the National Instruments Corporation and complements its main product family LabVIEW, a visual programming language-based system design platform and development environment (<https://www.ni.com/labview>).

MATLAB gained broad academic acceptance by integrating many new methodological developments in the control field into several control-related toolboxes. MATLAB also evolved as a powerful programming language, which allows easy object-oriented manipulation of different system descriptions via operator overloading. At present, the CACSD tools of MATLAB and Simulink represent the industrial and academic standard for CACSD. The existing CACSD tools are constantly extended and enriched with new model classes, new computational algorithms (e.g., structure-exploiting computations based on SLICOT), dedicated graphical user interfaces (e.g., tuning of PID controllers or control-related visualizations), advanced robust control system synthesis, etc. Also, many third-party toolboxes contribute to the wide usage of this tool.

Basic CACSD functionality incorporating symbolic processing techniques and higher-precision computations is available in the Maple product MapleSim Control Design Toolbox as

well as in the Mathematica Control Systems product. Free alternatives to MATLAB are the MATLAB-like environments Scilab, a French initiative pioneered by INRIA, and Octave, whose CACSD functionality strongly relies on SLICOT (see <https://octave.sourceforge.io/control/>).

Summary and Future Directions

The development and maintenance of integrated CACSD environments, which provide support for all aspects of the CACSD cycle such as modelling, design, and simulation, involve sustained, strongly interdisciplinary efforts. Therefore, the CACSD tool development activities must rely on the expertise of many professionals covering such diverse fields as control system engineering, programming languages and techniques, man-machine interaction, numerical methods in linear algebra and control, optimization, computer visualization, and model building techniques. This may explain why currently only a few of the commercial developments of prior years are still in use and actively maintained/developed. Unfortunately, the number of actively developed noncommercial alternative products is even lower. The dominance of MATLAB, as a de facto standard for both industrial and academic usage of integrated tools covering all aspects of the broader area of *computer-aided control engineering* (CACE), cannot be overlooked.

The new trends in CACSD are partly related to handling more complex applications, involving time-varying (e.g., periodic, multi-rate sampled-data, and differential algebraic) linear dynamic systems, nonlinear systems with many parametric uncertainties, and large-scale models (e.g., originating from the discretization of PDEs). To address many computational aspects of model building (e.g., model reduction of large order systems), optimization-based robust controller tuning using multiple-model approaches, or optimization-based robustness assessment using global-optimization techniques,

parallel computation techniques allow substantial savings in computational times and facilitate addressing computational problems for large-scale systems. A topic which needs further research is the exploitation of the benefits of combining numerical and symbolic computations (e.g., in model building and manipulation).

A promising new development for the future of CACSD is Julia, a powerful and flexible dynamic language, suitable for scientific and numerical computing, with performance comparable to traditional statically typed languages such as Fortran or C (see <https://julialang.org/>). As a programming language, Julia features optional typing, multiple dispatch, and good performance, achieved using type inference and just-in-time compilation (Bezanson et al. 2017). Julia is the basis of the implementation of the next-generation modelling language Modia (Elmqvist et al. 2016). CACSD-related functionality is provided in the Julia packages `ControlSystems` (<https://github.com/JuliaControl/ControlSystems.jl>) and `MatrixEquations` (<https://github.com/andreasvarga/MatrixEquations.jl>).

Cross-References

- ▶ [Basic Numerical Methods and Software for Computer Aided Control Systems Design](#)
- ▶ [Descriptor System Techniques and Software Tools](#)
- ▶ [Fault Detection and Diagnosis: Computational Issues and Tools](#)
- ▶ [Interactive Environments and Software Tools for CACSD](#)
- ▶ [Matrix Equations in Control](#)
- ▶ [Model Building for Control System Synthesis](#)
- ▶ [Model Order Reduction: Techniques and Tools](#)
- ▶ [Multi-domain Modeling and Simulation](#)
- ▶ [Optimization-Based Control Design Techniques and Tools](#)
- ▶ [Robust Synthesis and Robustness Analysis Techniques and Tools](#)
- ▶ [Validation and Verification Techniques and Tools](#)

Recommended Reading

The historical development of CACSD concepts and techniques was the subject of several articles in reference works of Rimvall and Jobling (1995) and Schmid (2002). A selection of papers on numerical algorithms underlying the development of CACSD appeared in Patel et al. (1994). The special issue No. 2/2004 of the IEEE Control Systems Magazine on *Numerical Awareness in Control* presents several surveys on different aspects of developing numerical algorithms and software for CACSD.

The main trends over the last three decades in CACSD-related research can be followed in the programs/proceedings of the biannual IEEE Symposia on CACSD from 1981 to 2013 (partly available at <https://ieeexplore.ieee.org>) as well as of the triennial IFAC Symposia on CACSD from 1979 to 2000. Additional information can be found in several CACSD-focused survey articles and special issues (e.g., No. 4/1982; No. 2/2000) of the IEEE Control Systems Magazine.

Bibliography

- Anderson E, Bai Z, Bishop J, Demmel J, Du Croz J, Greenbaum A, Hammarling S, McKenney A, Ostrouchov S, Sorensen D (1992) LAPACK user's guide. SIAM, Philadelphia
- Armstrong ES (1978) ORACLS – a system for linear-quadratic Gaussian control law design. Technical paper 1106 96-1, NASA
- Augustin DC, Strauss JC, Fineberg MS, Johnson BB, Linebarger RN, Sansom FJ (1967) The SCi continuous system simulation language (CSSL). *Simulation* 9:281–303
- Benner P, Mehrmann V, Sima V, Van Huffel S, Varga A (1999) SLICOT – a subroutine library in systems and control theory. In: Datta BN (ed) *Applied and computational control, signals and circuits*, vol 1. Birkhäuser, Boston, pp 499–539
- Bezanson J, Edelman A, Karpinski S, Shah VB (2017) Julia: a fresh approach to numerical computing. *SIAM Rev* 59(1):65–98. See also: <https://julialang.org/>
- Dongarra JJ, Moler CB, Bunch JR, Stewart GW (1979) LINPACK user's guide. SIAM, Philadelphia
- Elmqvist H (1978) A structured model language for large continuous systems. PhD thesis, Department of Automatic Control, Lund University, Sweden
- Elmqvist H et al (1997) Modelica – a unified object-oriented language for physical systems modeling (version 1). <https://www.modelica.org/documents/Modelica1.pdf>
- Elmqvist H, Henningsson T, Otter M (2016) Systems modeling and programming in a unified environment based on Julia. In: Margaria T, Steffen B (eds) *Leveraging applications of formal methods, verification and validation: discussion, dissemination, applications*. ISO/IEC JTC1/SC22, Lecture notes in computer science, vol 9953. Springer, Cham
- Garbow BS, Boyle JM, Dongarra JJ, Moler CB (1977) Matrix eigensystem routines – EISPACK guide extension. Springer, Heidelberg
- Grace ACW (1991) SIMULAB, an integrated environment for simulation and control. In: *Proceedings of American Control Conference*, Boston, pp 1015–1020
- Grübel G (1983) Die regelungstechnische Programmbibliothek RASP. *Regelungstechnik* 31:75–81
- Kalman RE, Englar TS (1966) A user's manual for the automatic synthesis program (program C). Technical report CR-475, NASA
- Lawson CL, Hanson RJ, Kincaid DR, Krogh FT (1979) Basic linear algebra subprograms for Fortran usage. *ACM Trans Math Softw* 5:308–323
- Mitchel EEL, Gauthier JS (1976) Advanced continuous simulation language (ACSL). *Simulation* 26:72–78
- Moler CB (1980) MATLAB users' guide. Technical report, Department of Computer Science, University of New Mexico, Albuquerque
- Moler CB, Little J, Bangert S, Kleinman S (1985) PC-MATLAB, users' guide, version 2.0. Technical report, The MathWorks Inc., Sherborn
- Nilsen RN, Karplus WJ (1974) Continuous-system simulation languages: a state-of-the-art survey. *Math Comput Simul* 16:17–25. [https://doi.org/10.1016/S0378-4754\(74\)80003-0](https://doi.org/10.1016/S0378-4754(74)80003-0)
- Patel RV, Laub AJ, Van Dooren P (eds) (1994) *Numerical linear algebra techniques for systems and control*. IEEE, Piscataway
- Rimvall C, Jobling CP (1995) Computer-aided control systems design. In: Levine WS (ed) *The control handbook*. CRC, Boca Raton, pp 429–442
- Schmid C (2002) Computer-aided control system engineering tools. In: Unbehauen H (ed) *Control systems, robotics and automation*. <https://www.eolss.net/outlinecomponents/Control-Systems-Robotics-Automation.aspx>
- Shah CS, Floyd MA, Lehman LL (1985) MATRIXx: control design and model building CAE capabilities. In: Jamshidi M, Herget CJ (eds) *Advances in computer aided control systems engineering*. North-Holland/Elsevier, Amsterdam, pp 181–207
- Smith BT, Boyle JM, Dongarra JJ, Garbow BS, Ikebe Y, Klema VC, Moler CB (1976) Matrix eigensystem routines – EISPACK guide. *Lecture notes in computer science*, vol 6, 2nd edn. Springer, Berlin/New York
- van den Boom A, Brown A, Geurts A, Hammarling S, Kool R, Vanbegin M, Van Dooren P, Van Huffel S (1991) SLICOT, a subroutine library in control and systems theory. In: *Preprints of 5th IFAC/IMACS*

- symposium of CADCS'91, Swansea. Pergamon Press, Oxford, pp 89–94
- Van Huffel S, Sima V, Varga A, Hammarling S, Delebecque F (2004) High-performance numerical software for control. *Control Syst Mag* 24:60–76
- Varga A, Davidoviciu A (1986) BIMASC – a package of Fortran subprograms for analysis, modelling, design and simulation of control systems. In: Hansen NE, Larsen PM (eds) *Preprints of 3rd IFAC/IFIP International Symposium on Computer Aided Design in Control and Engineering (CADCE'85)*, Copenhagen. Pergamon Press, Oxford, pp 151–156
- Varga A, Sima V (1985) BIMAS – a basic mathematical package for computer aided systems analysis and design. In: Gertler J, Keviczky L (eds) *Preprints of 9th IFAC World Congress, Hungary, vol 8*, pp 202–207
- Walker R, Gregory C, Shah S (1982) MATRIXx: a data analysis, system identification, control design and simulation package. *Control Syst Mag* 2:30–37
- White JS, Lee HQ (1971) Users manual for the variable automatic synthesis program (VASP). Technical memorandum TM X-2417, NASA

Conditioning of Quantum Open Systems

John Gough
Department of Physics, Aberystwyth University,
Aberystwyth, Wales, UK

Abstract

The underlying probabilistic theory for quantum mechanics is non-Kolmogorovian. The time-order in which physical observables are measured will be important if they are incompatible (non-commuting). In particular, the notion of conditioning needs to be handled with care and may not even exist in some cases. Here we layout the quantum probabilistic formulation in terms of von Neumann algebras and outline conditions (non-demolition properties) under which filtering may occur.

Keywords

Quantum filter · von Neumann algebra ·
Quantum probability · Bell's theorem

Introduction

The mathematical theory of quantum probability (QP) is an extension of the usual Kolmogorov probability to the setting inspired by quantum theory (Maassen 1988; Accardi et al. 1982; Hudson and Parthasarathy 1984; Parthasarathy 1992). In this entry, we will emphasize the quantum analogues to events (projections), random variables (operators), sigma-algebras (von Neumann algebras), probabilities (states), etc. The departure from Kolmogorov's theory is already implicit in the fact that the quantum random variables do not commute. It is advantageous to set up a noncommutative theory of probability, and one of the requirements is that Kolmogorov's theory is contained as the special case when we restrict to commuting observables. The natural analogue of measure theory for operators is the setting of von Neumann algebras (Kadison and Ringrose 1997).

Classical Probability

The standard approach to probability in the classical world is to associate with each model a Kolmogorov triple $(\Omega, \mathcal{F}, \mathbb{P})$ comprising a measurable space Ω , a sigma-algebra, $\mathcal{F}$, of subsets of Ω , and probability measure $\mathbb{P}$ on the sigma-algebra. Ω covers the entirety of all possible things that may unfold in our model, the elements of $\mathcal{F}$ are distinguished subsets of Ω referred to as events, and $\mathbb{P}[A]$ is the probability of event A occurring. From a practical point of view, we need to frame the model in terms of what we may hope to observe, and these constitute the “events,” $\mathcal{F}$, and from a mathematical point of view, restricting to a sigma-algebra clarifies all the ambiguities while also resolving all the technical issues, pathologies, etc. that would otherwise plague the subject.

A **random variable** is then defined as a function on Ω to a value space, another measurable space $(\Gamma, \mathcal{G})$: so, for each $G \in \mathcal{G}$, the set $X^{-1}[G] = \{\omega \in \Omega : X(\omega) \in G\}$ is an event, i.e., in $\mathcal{F}$. The probability that X takes a value in G is then $\mathbb{K}_X[G] = \mathbb{P}[X^{-1}[G]]$ which produces

the distribution of X as $\mathbb{K}_X = \mathbb{P} \circ \mathbb{X}^{-1}$. Let $X_1, \dots, X_n$ be random variables and then their joint probabilities are also well-defined:

$$\begin{aligned} \mathbb{K}_{X_1, \dots, X_n} [G_1, \dots, G_n] \\ = \mathbb{P} [X_1^{-1} [G_1] \cap \dots \cap X_n^{-1} [G_n]]. \end{aligned} \quad (1)$$

Let $\mathcal{A}$ be a sub-sigma-algebra of $\mathcal{F}$. The collection of bounded $\mathcal{A}$ -measurable (complex-valued) functions will be denoted as $\mathcal{A} = L^\infty(\Omega, \mathcal{A})$. This is an example of a $*$ -algebra of functions. We now show that there is a natural identification between σ -algebras of subsets of Ω and $*$ -algebras of functions on Ω . First we need some definitions.

Definition 1 A sequence $(f_n)_n$ of functions on Ω is said to be a **positive bounded monotone sequence** if there exists a finite constant $c > 0$ such that $0 \leq f_n \leq c$ and $f_n \leq f_{n+1}$ for each n . A $*$ -algebra of functions $\mathcal{A}$ on Ω is said to be a **monotone class** if every positive bounded monotone sequence in $\mathcal{A}$ has its limit in $\mathcal{A}$.

The next result can be found in Protter (2005).

Theorem 1 (Monotone class theorem) *Given a monotone class $*$ -algebra of functions, $\mathcal{A}$, on Ω . Then $\mathcal{A} = L^\infty(\Omega, \mathcal{A})$ where $\mathcal{A}$ is precisely the σ -algebra generated by $\mathcal{A}$ itself.*

Conditional probabilities are naturally defined: the probability that event A occurs given that event B occurs is

$$\mathbb{P}[A|B] \triangleq \frac{\mathbb{P}[A \cap B]}{\mathbb{P}[B]}. \quad (2)$$

This is a simple “renormalization” of the probability: one restricts the outcomes in A which also lie in B , weighted as a proportion out of all B , rather than Ω .

Quantum Probability Models

The standard presentation of quantum mechanics takes physical quantities (observables) to be self-adjoint operators on a fixed Hilbert space $\mathfrak{h}$ of wavefunctions. The normalized elements of $\mathfrak{h}$ are the **pure states**, and the expectation of an observable X for a pure state $\psi \in \mathfrak{h}$ is given

$\mathbb{E}[X] = \langle \psi | X \psi \rangle$. As such, observables play the role of random variables. More generally, we encounter quantum expectations of the form

$$\mathbb{E}[X] = \text{tr} \{ \rho X \} \quad (3)$$

where $\rho \geq 0$ is a trace-class operator normalized so that $\text{tr} \{ \rho \} = 1$. The operator ρ is called a **density matrix** and in the pure case corresponds to $\rho = |\psi\rangle\langle\psi|$.

Definition 2 A **quantum probability space** $(\mathfrak{A}, \mathbb{E})$ consists of a von Neumann algebra $\mathfrak{A}$ and a state $\mathbb{E}$ (assumed to be continuous in the normal topology).

When $\mathfrak{A}$ is commutative, then the quantum probability space is isomorphic to a Kolmogorov model. Let us motivate now why von Neumann algebras are the appropriate object. Positive operators are well defined, and we may say $X \leq Y$ if $Y - X \geq 0$. In particular, the concept of a positive bounded monotone sequence of operators makes sense, as does a monotone class algebra of operators.

Theorem 2 (van Handel 2007) *A collection of bounded operators over a fixed Hilbert space is a von Neumann algebra if and only if it is a monotone class $*$ -algebra.*

Specifically, we see that this recovers the usual monotone class theorem when we further impose commutativity of the algebra. A state on a von Neumann algebra, $\mathfrak{A}$, is then a normalized positive linear map from $\mathfrak{A}$ to the complex numbers, that is, $\mathbb{E}[1] = 1$, $\mathbb{E}[\alpha A + \beta B] = \alpha \mathbb{E}[A] + \beta \mathbb{E}[B]$, $\mathbb{E}[X] \geq 0$, whenever $X \geq 0$. Note that if (X_n) is a positive bounded monotone sequence with limit X , then $\mathbb{E}[X_n]$ converges to $\mathbb{E}[X]$ – this in fact equivalent to the condition of continuity in the normal topology and implies that $\mathbb{E}$ to take the form (3), for some density matrix ρ , (Kadison and Ringrose 1997).

Definition 3 An observable is referred to as a **quantum event** if it is an orthogonal projection. If a quantum event A corresponds to a projection P_A , then its probability of occurring is $\text{Pr} \{A\} = \text{tr} \{ \rho P_A \}$.

The requirement that P_A is an orthogonal projection means that $P_A^2 = P_A = P_A^\dagger$. The complement to the event A , which is *not* A , will be denoted as $\bar{A}$ and is the orthogonal projection given by the orthocomplement $P_{\bar{A}} = \mathbb{1} - P_A$, where $\mathbb{1}$ is the identity operator on $\mathfrak{h}$. A von Neumann algebra is a subalgebra of the bounded operators on $\mathfrak{h}$ with good closure properties: crucially it will be generated by its projections. So the von Neumann algebra generated by a collection of quantum events is the natural analogue of a sigma-algebra of classical events.

However, the new theory of quantum probability will have features not present classically. For instance, the notion of a pair of events, A and B , occurring jointly is not generally meaningful. In fact, $P_A P_B$ is in general not self-adjoint, and so it does not correspond to a quantum event. We therefore cannot interpret $\text{tr}\{\rho P_A P_B\}$ as the joint probability for quantum events A and B to occur.

Proposition 1 *The product $P_A P_B$ is an event (i.e., an orthogonal projection) if and only if P_A and P_B commute.*

Proof. Self-adjointness of $P_A P_B$ is enough to give the commutativity since then $P_A P_B = (P_A P_B)^* = P_B P_A$ as both P_A and P_B are self-adjoint. We then see that $(P_A P_B)^2 = P_A P_B P_A P_B = P_A^2 P_B^2 = P_A P_B$.

As such, given a pair of quantum events A and B , it is usually meaningless to speak of their joint probability $\text{Pr}\{A, B\}$ for a fixed state ρ . An exception is made when the corresponding projectors commute in which case we say the events are compatible and take the probability to be $\text{Pr}\{A, B\} \equiv \text{tr}\{\rho P_A P_B\}$.

By the spectral theorem, every observable may be written as

$$X = \int_{-\infty}^{\infty} x P_X[dx] \quad (4)$$

where $P_X[dx]$ is a projection valued measure, normalized so that $P_X[\mathbb{R}] = \mathbb{1}$, the identity operator. The measure is supported on the spectrum of X which is, of course, real by self-adjointness. In particular, if G_1 and G_2 are non-overlapping

Borel subsets of $\mathbb{R}$, then $P_X[G_1]$ and $P_X[G_2]$ project onto mutually orthogonal projections.

The orthogonal projection $P_X[G]$ then corresponds to the quantum event that X is measured to have a value in the subset G . The smallest von Neumann algebra containing all the projections $P_X[G]$ will be denoted as $\mathfrak{F}_X$ and plays an analogous role to the sigma-algebra generated by a random variable.

Once we fix the density matrix ρ , the spectral decomposition leads to the probability distribution of observable X : $\mathbb{K}_X[dx] = \text{tr}\{\rho P_X[dx]\}$. We say that observables $X_1, \dots, X_n$ are **compatible** if the quantum events they generate are compatible. In this case

$$\text{tr}\left\{\rho e^{i \sum_{k=1}^n u_k X_k}\right\} \equiv \int e^{i \sum_{k=1}^n u_k x_k} \mathbb{K}_{X_1, \dots, X_n}[dx_1, \dots, dx_n], \quad (5)$$

where $\mathbb{K}_{X_1, \dots, X_n}[dx_1, \dots, dx_n]$ defines a probability measure on the Borel sets of $\mathbb{R}^n$. This may be no longer true if we drop the compatibility assumption!

We remark that, given a collection of observables $X_1, \dots, X_n$, we can construct the smallest von Neumann algebra containing all their individual quantum events; this will typically be a non-commutative algebra and will effectively play the role of a sigma-algebra generated by random variables.

Positivity preserving measurable mappings are the natural morphisms between Kolmogorov spaces. The situation in quantum probability is rather more prescriptive.

First note that if $\mathfrak{A}$ and $\mathfrak{B}$ are von Neumann algebras, then so too is their formal tensor product. A map $\Phi : \mathfrak{F} \rightarrow \mathfrak{F}$ between von Neumann algebras is positive if $\Phi(A) \geq 0$ whenever $A \geq 0$. However, we need a stronger condition. The mapping has the extension $\tilde{\Phi}_{\mathfrak{F}} : \mathfrak{A} \otimes \mathfrak{F} \rightarrow \mathfrak{B} \otimes \mathfrak{F}$ for a given von Neumann algebra $\mathfrak{B}$ by $\tilde{\Phi}_{\mathfrak{F}}(A \otimes F) = \Phi(A) \otimes F$. We say that Φ is **completely positive (CP)** if $\tilde{\Phi}_{\mathfrak{F}}$ is positive for any $\mathfrak{F}$.

A **morphism between a pair of quantum probability spaces** (Maassen 1988), $\Phi :$

$(\mathfrak{A}_1, \mathbb{E}_1) \mapsto (\mathfrak{A}_2, \mathbb{E}_2)$ is a completely positive map with the properties $\Phi(\mathbb{1}_{\mathfrak{A}_1}) = \mathbb{1}_{\mathfrak{A}_2}$ and $\mathbb{E}_2 \circ \Phi = \mathbb{E}_1$. Despite its rather trivial-looking appearance, the CP property is actually quite restrictive.

Quantum Conditioning

The conditional probability of event A occurring given that B occurs is defined by $\Pr\{A|B\} = \frac{\Pr\{A, B\}}{\Pr\{B\}}$. In quantum probability, $\Pr\{A, B\}$ may make sense as a joint probability only if A and B are compatible; otherwise there is the restriction that A is measured before B .

Let X be an observable with a discrete spectrum (eigenvalues). If we start in a pure state $|\psi_{\text{in}}\rangle$ and measure X to record a value x , then this quantum event has corresponding projector $P_X(x)$. Von Neumann's projection postulate states that the state after measurement is proportional to

$$|\psi_{\text{out}}\rangle = P_X(x) |\psi_{\text{in}}\rangle \quad (6)$$

and that the probability of this event is $\Pr\{X = x\} = \langle \psi_{\text{in}} | P_X(x) | \psi_{\text{in}} \rangle \equiv \langle \psi_{\text{out}} | \psi_{\text{out}} \rangle$. Note that $|\psi_{\text{out}}\rangle$ is not normalized! A subsequent measurement of another discrete observable Y , leading to eigenvalue y , will result in $|\psi_{\text{out}}\rangle = P_Y(y) P_X(x) |\psi_{\text{in}}\rangle$ and so (ignoring dynamics from the time being)

$$\begin{aligned} \Pr\{X = x; Y = y\} &= \langle \psi_{\text{out}} | \psi_{\text{out}} \rangle \\ &= \langle \psi_{\text{in}} | P_Y(y) P_X(x) P_Y(y) | \psi_{\text{in}} \rangle. \end{aligned} \quad (7)$$

This needs to be interpreted as a sequential probability – event $X = x$ occurs first, and $Y = y$ second – rather than a joint probability. If X and Y do not commute, then the order in which they are measured matters as $\Pr\{X = y; Y = y\}$ may differ from $\Pr\{Y = y; X = x\}$.

Lemma 1 *Let P and Q be orthogonal projections, then the properties $QPQ = PQP$ and $PQP = QP$ are equivalent to $[Q, P] = 0$.*

Proof. We begin by noting that $[P, Q]^* [P, Q] = QPQ - PQPQ + PQP - QPQP$. This may

be rewritten as $(Q - P)(QPQ - PQP)$, so if $QPQ = PQP$, then $[P, Q] = 0$. If we have $PQP = QP$, then $[P, Q]^* [P, Q] \equiv QPQ - QPQ + PQP - QP = PQP - QP = 0$ and so $[P, Q] = 0$. Conversely, if P and Q commute then $QPQ = PQP$ and $PQP = QP$ hold true.

Corollary 1 *If we have $\Pr\{X = x; Y = y\}$ equal to $\Pr\{Y = y; X = x\}$ for all states $|\psi_{\text{in}}\rangle$ and all eigenvalues x and y are equal, then X and Y are compatible.*

The symmetry implies that $P_Y(y) P_X(x) P_Y(y)$ equals $P_X(x) P_Y(y) P_X(x)$, and so the spectral projections of X and Y commute.

Corollary 2 *If for all states $|\psi_{\text{in}}\rangle$ whenever we measure X and then Y and then X again, we always record the same value for X , and then X and Y are compatible.*

Setting $|\psi_{\text{out}}\rangle = P_Y(y) P_X(x) |\psi_{\text{in}}\rangle$, we must have that $P_X(x) |\psi_{\text{out}}\rangle = |\psi_{\text{out}}\rangle$; if this is true for all $|\psi_{\text{in}}\rangle$, then $P_X(x) P_Y(y) = P_X(x) P_Y(y) P_X(x)$ which likewise implies that $[P_X(x), P_Y(y)] = 0$.

Conditioning over Time

The dynamics under a (possibly time-dependent) Hamiltonian $H(t)$ is described by the two-parameter family of unitary operators

$$U(t, s) = \mathbb{1} - i \int_s^t H(\tau) U(\tau, s) d\tau, \quad t \geq s. \quad (8)$$

This is the solution to $\frac{\partial}{\partial t} U(t, s) = -i H(t) U(t, s)$, $U(s^-, s) = \mathbb{1}$. We have the **flow identity**

$$U(t_3, t_2) U(t_2, t_1) = U(t_3, t_1) \quad (9)$$

whenever $t_3 \geq t_2 \geq t_1$. In the special case where H is constant, we have $U(t, s) = e^{-i(t-s)H}$. It is convenient to introduce the maps ($t_2 > t_1$)

$$J_{(t_1, t_2)}(X) = U(t_2, t_1)^* X U(t_2, t_1) \quad (10)$$

and, from the flow identity (9), $J_{(t_1, t_2)} \circ J_{(t_2, t_3)} = J_{(t_1, t_3)}$ whenever $t_3 > t_2 > t_1$.

We now consider an experiment where we measure observables $Z_1, \dots, Z_n$ at times $t_1 < t_2 < \dots < t_n$ during a time interval 0 to T . At the end of the experiment, if we measure $Z_1 \in G_1, \dots, Z_n \in G_n$, then the output state should be

$$|\psi_{\text{out}}\rangle = U(T, t_n) P_{Z_n}[G_n] U(t_n, t_{n-1}) \dots U(t_2, t_1) P_{Z_1}[G_1] U(t_1, 0) |\psi_{\text{in}}\rangle. \quad (11)$$

It is convenient to introduce the observables

$$Y_k = J_{(0, t_k)}(Z_k). \quad (12)$$

The quantum event $Y_k \in G_k$ at time t_k is then $P_{Y_k}[G_k] = J_{(0, t_k)}(P_{Z_k}[G_k])$. The flow identity implies $U(t_k, t_{k-1}) = U(t_k, 0) U(t_{k-1}, 0)^*$, and we find

$$\begin{aligned} |\psi_{\text{out}}\rangle &= U(T, 0) J_{(0, t_n)} \\ &\quad (P_{Z_n}[G_n]) \dots J_{(0, t_1)}(P_{Z_1}[G_1]) |\psi_{\text{in}}\rangle \\ &= U(T, 0) P_{Y_n}[G_n] \dots P_{Y_1}[G_1] |\psi_{\text{in}}\rangle. \end{aligned} \quad (13)$$

We therefore have the probability

$$\begin{aligned} \Pr\{Y_1 \in G_1; \dots; Y_n \in G_n\} &= \langle \psi_{\text{out}} | \psi_{\text{out}} \rangle \\ &= \text{tr} \{ P_{Y_n}[G_n] \dots P_{Y_1}[G_1] \rho \\ &\quad P_{Y_1}[G_1] \dots P_{Y_n}[G_n] \}, \end{aligned}$$

where ρ is the initial state. Note that this takes the **pyramidal** form.

Here the Z_k are all to be understood as observables specified in the Schrödinger picture at time 0 – what we measure are the Y_k which are the Z_k at respective times t_k . The answer depends on the chronological order $t_1 < t_2 < \dots < t_n$.

It is tempting to think of $\{Y_k : k = 1, \dots, n\}$ as a discrete time stochastic process, but some caution is necessary. We cannot generally permute the events $Y_1 \in G_1; \dots; Y_n \in G_n$, so we do not have the symmetry usually associated with Kolmogorov's reconstruction theorem.

Proposition 2 *The finite-dimensional distributions satisfy the marginal consistency for the most recent variable. Specifically, this means that*

$$\begin{aligned} \sum_G \Pr\{Y_1 \in G_1; \dots; Y_n \in G\} \\ = \Pr\{Y_1 \in G_1; \dots; Y_{n-1} \in G_{n-1}\} \end{aligned} \quad (14)$$

where the sum is over a collectively exhaustive mutually exclusive set $\{G\}$.

Proof. We have

$$\sum_G \Pr\{Y_1 \in G_1; \dots; Y_n \in G\} \quad (15)$$

$$\begin{aligned} &= \sum_G \text{tr} \{ P_{Y_{n-1}}[G_{n-1}] \dots P_{Y_1}[G_1] \rho \\ &\quad P_{Y_1}[G_1] \dots P_{Y_{n-1}}[G_{n-1}] P_{Y_n}[G] \} \end{aligned} \quad (16)$$

but $\sum_G P_{Y_n}[G] \equiv \mathbb{1}$, so we obtain the desired reduction.

As an example, suppose that $\{A_k\}$ is a collection of quantum events occurring at a fixed time t_1 which are mutually exclusive (i.e., their projections project onto orthogonal subspaces). Their union $\cup_k A_k$ makes sense and corresponds to the projection onto the direct sum of these subspaces. Now if B is an event at a later time t_2 , then typically $\sum_k \Pr\{A_k; B\}$ is not the same as $\Pr\{\cup_k A_k; B\}$. The well-known two-slit experiment fits into this description, with A_k the event that an electron goes through slit k ($k = 1, 2$) and B the subsequent event that it hits a detector.

Marginal consistency is a property we take for granted in classical stochastic processes and is an essential requirement for Kolmogorov's reconstruction theorem. However, in the quantum setting, it is only guaranteed to work for the last measured observable. For instance, it may not

apply to Y_{n-1} unless we can commute its projections with $P_{Y_n}[G_n]$. The most recent measured observable has the potential to **demolish** all the measurements beforehand.

Bell's Inequalities

A Bell inequality is any constraint that applies to classical probabilities but which may fail in quantum probability. We look at one example due to Eugene Wigner.

Proposition 3 (A Bell Inequality) *Given three (classical) events A, B, C then we always have*

$$\Pr\{A, C\} \leq \Pr\{A, \overline{B}\} + \Pr\{B, C\}. \quad (17)$$

Proof. From the marginal property, we have the following three classical identities $\Pr\{A, \overline{B}\} = \Pr\{A, \overline{B}, C\} + \Pr\{A, \overline{B}, \overline{C}\}$, $\Pr\{B, C\} = \Pr\{A, B, C\} + \Pr\{\overline{A}, B, C\}$, and $\Pr\{A, C\} = \Pr\{A, B, C\} + \Pr\{A, \overline{B}, C\}$. We therefore have that

$$\begin{aligned} & \Pr\{A, \overline{B}\} + \Pr\{B, C\} - \Pr\{A, C\} \\ &= \Pr\{A, \overline{B}, \overline{C}\} + \Pr\{\overline{A}, B, C\} \geq 0. \end{aligned} \quad (18)$$

The proof relies on the fact that marginal consistency is always valid for classical events. Suppose that the A, B, C are quantum events, say Y_k taking a value in G_k at time t_k ($k = 1, 2, 3$), and chronologically ordered ($t_1 < t_2 < t_3$). Then only the first of the three classical identities is guaranteed in quantum theory (marginal consistency for the latest event at time t_3 only). The remaining two may fail, and it is easy to construct a quantum system where inequality (17) is violated.

Quantum Filtering

Given the issues raised above, one may ask whether it is actually possible to track a quantum system over time?

We shall say that the process $\{Y_k: k=1, \dots, n\}$ is **essentially classical** whenever all the observables are compatible. In this case its

finite-dimensional distributions satisfy all the requirements of Kolmogorov's theorem, and so we can model it as a classical stochastic process. The von Neumann algebra $\mathfrak{V}_n$ they generate will be commutative, and we have $\mathfrak{V}_n \subset \mathfrak{V}_{n+1}$ (a filtration of von Neumann algebras!).

For the tracking over time to be meaningful, we ask for the observed process to be essentially classical – this is the **self-non-demolition property** of the observations.

Let $X_n = J_{(0, t_n)}(X)$ be an observable X at time t_n . Suppose that it is compatible with $\mathfrak{V}_n$, then we are lead to a well-defined classical joint distribution $\mathbb{K}_{X_n, Y_1, \dots, Y_n}$ from which we may compute the conditional probability $\mathbb{K}_{X_n|Y_1, \dots, Y_n}$. This means that $\pi_{t_n}(X) = \mathfrak{E}[X_n|\mathfrak{V}_n]$ is well-defined.

We only try to condition those observables that are compatible with the measured observables! This is known as the **non-demolition property**.

Conditioning in Quantum Theory

The natural analogue of conditional expectation in quantum theory would be a projective morphism (CP map) from a von Neumann algebra into a sub-von Neumann algebra. However, this does not to always exist in the noncommutative case. Given our discussions above, this is not surprising.

In general, be $\mathfrak{V}$ be a sub-von Neumann algebra of a von Neumann algebra $\mathfrak{B}$, then its commutant is the set of all operators in $\mathfrak{B}$ which commute with each element of $\mathfrak{V}$, that is

$$\mathfrak{V}' = \{X \in \mathfrak{B} : [X, Y] = 0, \forall Y \in \mathfrak{V}\}.$$

Why explain. The von Neumann algebra generated by $\mathfrak{V}$ and fixed element $X \in \mathfrak{V}'$ will again be a commutative, so the conditional expectation of X onto $\mathfrak{V}$ is well defined. Physically, it means that $X \in \mathfrak{V}'$ is compatible with $\mathfrak{V}$, so standard classical probabilistic constructs are valid.

As an illustration, suppose that the von Neumann algebra of a system of interest is $\mathfrak{A}$ and its environment's is $\mathfrak{E}$. Let X be a simple observable of the system, say with spectral decomposition $X = \sum_x x R_x$ where $\{R_x\}$ is a set of mutually orthogonal projections. Similarly, let

$Z = \sum_y y Q_y$ be an environment observable. We entangle the system and environment with a unitary U (acting on $\mathfrak{A} \otimes \mathfrak{E}$) and measure the observable $Y = U^*(\mathbb{1}_{\mathfrak{A}} \otimes Z)U \equiv \sum_y P_y$. Here, the event corresponding to measuring eigenvalue y of Y is $P_y = U^*(\mathbb{1}_{\mathfrak{A}} \otimes Q_y)U$. The observables $X_1 = U^*(X \otimes \mathbb{1}_{\mathfrak{E}})U$ and Y commute, and so they have a well-defined joint probability to have eigenvalues x and y , respectively, for a fixed state $\mathbb{E}$: $p(x, y) = \mathbb{E}[U^*(R_x \otimes Q_y)U]$. We may think of X_1 as X evolved by one time step. Its conditional expectation given the measurement of Y is then $\mathbb{E}[X_1|\mathfrak{Y}] = \sum_{x,y} x p(x|y) P_y$ where $p(x|y) = p(x, y)/p(y)$, with $p(y) = \sum_x p(x, y)$.

Summary and Future Directions

Quantum theory can be described in a systematic manner, despite the frequently sensationalist presentations on the subject. Getting the correct mathematical setting allows one to develop an operational approach to quantum measurements and probabilities. We described the quantum probabilistic framework which uses von Neumann algebras in place of measurable functions and shown how some of the usual concepts from Kolmogorov's theory carry over. However, we were also able to highlight which features of the classical world may fail to be valid in quantum theory.

What is of interest to control theorists is that the extraction of information from quantum measurements can be addressed and, under appropriate conditions, filtering can be formulated. As we have seen, measuring quantum systems over time is problematic. However, the self-non-demolition property of the observations and the non-demolition principle are conditions which guarantee that the filtering problem is well-posed. These conditions are met in continuous-time quantum Markovian models (by causality, prediction turns out to be well-defined, though not necessarily smoothing!). Explicit forms for the filter were given by Belavkin (1980); see also Bouten et al. (2007).

Cross-References

► [Quantum Stochastic Processes and the Modelling of Quantum Noise](#)

Recommended Reading

The mathematical program of “quantizing” probability emerged in the 1970s and has produced a number of technical results used by physicists. However, it the prospect technological applications that have seen QP adopted as the natural framework for quantum analogues of engineering such as filtering and control. The philosophy of QP is given in Maassen (1988), for instance, with the main tool – quantum Ito calculus – in Hudson and Parthasarathy (1984); Parthasarathy (1992). The theory of quantum filtering was pioneered by V.P. Belavkin Belavkin (1980) with modern accounts in van Handel (2007); Bouten et al. (2007).

Bibliography

- Accardi L, Frigerio A, Lewis JT (1982) Quantum stochastic processes. *Publ Res Inst Math Sci* 18(1):97–133
- Belavkin VP (1980) Quantum filtering of Markov signals with white quantum noise. *Radiotekhnika i Elektronika* 25:1445–1453
- Bouten L, van Handel R, James MR (2007) An introduction to quantum filtering. *SIAM J Control Optim* 46:2199–2241
- Hudson RL, Parthasarathy KR (1984) Quantum Ito's formula and stochastic evolutions. *Commun Math Phys* 93:301
- Kadison RV, Ringrose JR (1997) Fundamentals of the theory of operator algebras. Volumes I (Elementary Theory) and II (Advanced Theory). American Mathematical Society, Providence
- Maassen H (1988) Theoretical concepts in quantum probability: quantum Markov processes. In: *Fractals, quasicrystals, chaos, knots and algebraic quantum mechanics*, NATO advanced science institutes series C, Mathematical and physical sciences, vol 235. Kluwer Academic Publishers, Dordrecht, pp 287–302
- Parthasarathy KR (1992) An introduction to quantum stochastic calculus. Birkhauser, Basel
- Protter P (2005) Stochastic integration and differential equations, 2nd edn. Springer, Berlin Heidelberg
- van Handel R (2007) Filtering, stability, and robustness. Ph.D. Thesis, California Institute of Technology

Connected and Automated Vehicles

A. A. Kurzhanskiy, F. Borrelli, and P. Varaiya
University of California, Berkeley, Berkeley,
CA, USA

Abstract

This is a selective survey of technologies for the coordinated control of vehicles. The vehicles are connected: they may communicate with each other and with the infrastructure. Moreover, some vehicle planning and actuation functions are coordinated and automated, i.e., determined by on-board or (cloud-based) infrastructure processors. The entry emphasizes technologies that use communication to coordinate the movement of vehicles and deemphasizes developments that focus on autonomous driving. Hence, sensing, planning, and control technologies that are oriented to autonomous (driver-less) vehicles are not considered. The entry evaluates coordinated control technologies in terms of their contribution to greater safety, mobility, and efficiency of the US road transportation system. Safety benefits translate to reduced crash rates; mobility improvements lead to increased throughput of the road system; and higher efficiency means reduced energy consumption for a given pattern of traffic.

Keywords

Connected automated vehicles (CAVs) · V2X · Platoons · Cruise control · ACC · CACC · PCC · MPC · Power train · SPaT

Introduction

Connected and automated vehicles (CAVs) have the potential to disrupt mobility, extending what is possible with driving automation and connectivity alone. Applications include real-time control and planning with increased awareness,

routing with micro-scale traffic information, coordinated platooning using traffic signals information, and eco-mobility on demand with guaranteed parking. This entry introduces the technology, the control problems, and opportunities of CAVs and surveys the state of the art of CAVs.

We start by reviewing the definition of automated driving. In 2014, the Society of Automotive Engineers (SAE International) released a six-level classification of automated driving, known as SAE J3016 (Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems [2014](#)):

- **Level 0.** The automated system has no vehicle control, but may issue warnings.
- **Level 1.** An ADAS on the vehicle can assist the human driver with steering and braking/accelerating. Here, the automated system may include adaptive cruise control (ACC) or cooperative ACC (CACC), parking assistance with automated steering, and lane keeping assistance.
- **Level 2.** An ADAS on the vehicle can control both steering and braking/accelerating under some circumstances. The driver must continue to pay full attention (“monitor the driving environment”) at all times and perform the rest of the driving task. The automated system must deactivate immediately upon takeover by the driver.
- **Level 3.** An automated driving system (ADS) on the vehicle can perform all aspects of the driving task within certain operational environments (e.g., freeways). Within these environments, the driver can safely turn their attention away from driving task. However, the driver must be ready to take back control at any time the ADS requests the driver to do so. In all other circumstances, the driver performs the driving task.
- **Level 4.** An ADS on the vehicle can itself perform all driving tasks and monitor the driving environment – essentially, do all the driving – in certain circumstances. The driver should enable the ADS only when it is safe to do

so, but when the ADS is enabled, driver's attention is no longer required.

- **Level 5.** An ADS on the vehicle can do all the driving in all circumstances, where it is legal to drive. The human occupants are just passengers and need not be involved in driving, other than starting the system and setting the destination.

The most advanced automated production vehicles today belong to Level 2: functional motion control systems exist; however, sensing and perception technologies are not yet able to fully replace the driver even in limited operational environments.

Connected vehicles are vehicles that use any of a number of different communication technologies to communicate with the road actors (such as other vehicles, pedestrian, bicyclist) and infrastructure (such as traffic lights and parking facilities). Some common definitions include:

- *Vehicle-to-vehicle (V2V)* enables vehicles to wirelessly exchange information about their speed, location, and heading. The technology behind V2V communication allows vehicles to broadcast and receive omni-directional messages (up to ten times per second), creating a 360-degree “awareness” of other vehicles in proximity.
- *Vehicle-to-infrastructure (V2I)* captures vehicle-generated traffic data, wirelessly providing information such as advisories from the infrastructure to the vehicle that inform the driver of safety, mobility, and/or environment-related conditions (e.g., signals).
- *Vehicle-to-pedestrian (V2P)* encompasses a broad set of road users including people walking, children being pushed in strollers, people using wheelchairs or other mobility devices, passengers embarking and disembarking buses and trains, and cyclists.
- *Vehicle-to-cloud (V2C)* enables communication with the “cloud.” This makes a vehicle an Internet-connected endpoint equipped with a data storage and analytics capabilities of the cloud.

All of these communication technologies are addressed as *vehicle-to-everything (V2X)*. Part of V2X messaging is defined by SAE in its J2735 standard (Dedicated Short Range Communications [DSRC](#)). V2X refers to an intelligent transport system where all vehicles and infrastructure systems are interconnected with each other.

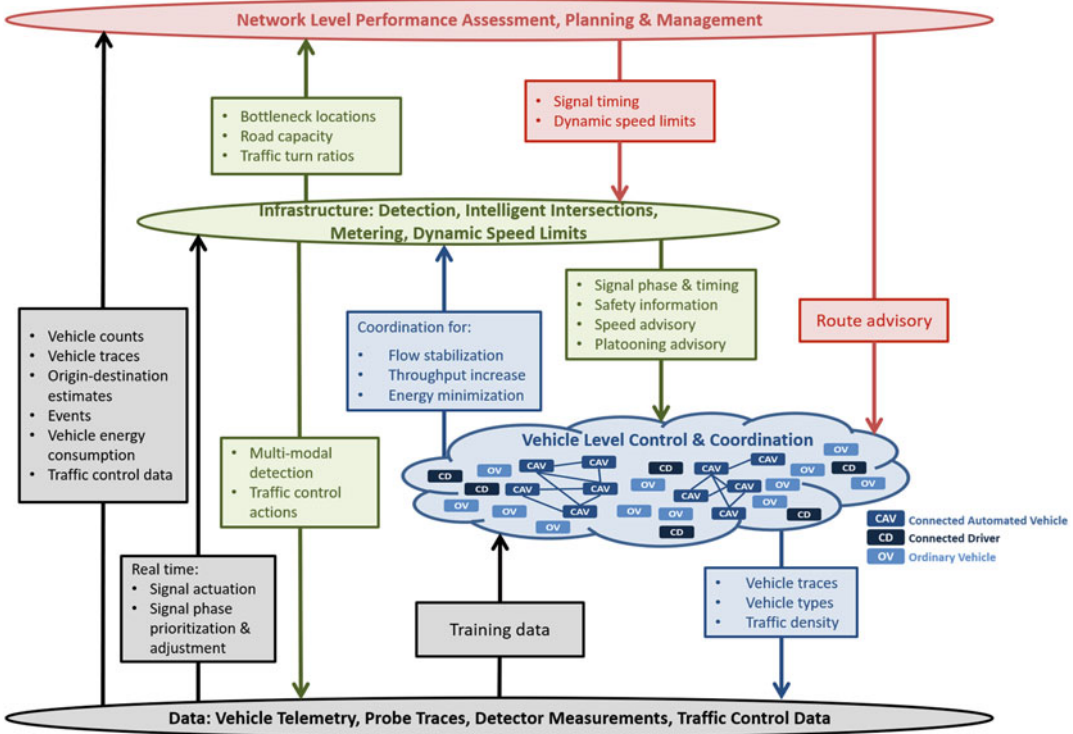
Vehicle connectivity enables many convenience features and services, including emergency calls, toll payment, and infotainment. It has also emerged as a technology to improve safety and performance and enable vehicle cooperation.

CAV technology has the potential to extend what is possible with driving automation and vehicle connectivity separately. It is envisioned that V2X will provide precise knowledge of the traffic situation across the entire road network, which in turn, using automation, should help in:

1. Reducing accidents – This provides crash-free driving and improved vehicle safety; a vehicle can monitor the environment continuously, making up for lapses in driver attention.
2. Optimizing traffic flows – This will reduce the need for building new infrastructure and reduce maintenance costs. This will also reduce uncertainty in travel times via real-time, predictive assessment of travel times on all routes.
3. Improving energy efficiency through more efficient driving; lighter, more fuel-efficient vehicles; and efficient infrastructure.

CAV Ecosystem

CAV applications for optimization of safety, traffic flow, and energy consumption build upon the extensive information fed to automatic control algorithms. The information exchange and control happens on three levels: vehicle level, roadside level, and network level. Figure 1 shows the information flow diagram articulating the three levels of control. Data generated by CAVs,



Connected and Automated Vehicles, Fig. 1 Information flow in CAV applications

roadside detection, and control units form the foundation of this diagram.

A CAV can provide information about its location coming from an onboard GNSS (Global Navigation Satellite System is the general term for GPS, GLONASS, Galileo, BeiDou, and other regional navigation systems.); dynamics (heading, speed, acceleration); and, possibly, intent to turn or switch lanes (when a blinker is on) with some periodicity (typically, 10 Hz). Present communication technologies that a CAV can use to send these data are dedicated short-range communication (DSRC) and cellular networks (LTE and 5G).

DSRC is a wireless communication technology, mostly conceived for active safety and convenience services. Advantages of the DSRC technology include security, low latency, interoperability, and resilience to extreme weather conditions. The DSRC technology may also be used to improve GPS accuracy (Alam et al. 2011) and for geo-fencing. One problem of DSRC

is that it requires special hardware. Another limitation is its short-range limitation enabling CAV communication only with nearby CAVs and infrastructure. DSRC is not suitable for transferring large data.

5G offers both, the communication with other vehicles and infrastructure and the access to multimedia and cloud services. Using 5G, a CAV would be able to provide information coming from its onboard sensors that can include lidar, radar, cameras, and ultrasonic sensors. This information would describe the CAV environment – presence and density of other agents (other vehicles, cyclists, pedestrians); workzones; obstacles; and hazards.

V2X messages presently defined in the SAE J2735 standard (Dedicated Short Range Communications [DSRC](#)) are mostly oriented toward awareness and safety applications. Advanced vehicle cooperation, including multi-vehicle formations and cooperative awareness, requires more complex protocols (van de Sluis

et al. 2015), for which standards (such as the announced J2945/6 standard) are still under development. An appealing niche for cooperative vehicles is freight transportation; heavy-duty vehicles capable of automated driving and V2V communication can form *platoons* and drive at small inter-vehicular distance, thereby reducing their air drag resistance and fuel consumption (Peloton Technology).

The roadside infrastructure includes complex systems for traffic monitoring and control deployed on freeways and at urban intersections. Freeways are instrumented with systems for vehicle detection, to monitor and potentially control the traffic flow. A variety of detection technologies are employed, including in-roadway sensors (loop detectors, magnetic detectors, magnetometers) and over-roadway sensors (cameras, radars, ultrasonic, infrared, and acoustic sensors) (Gibson et al. 2007). The use of sensors in freeways includes data collection (vehicle occupancy, speed, type) for monitoring and planning of road use (California Department of Transportation) and active traffic control via ramp metering.

A typical instrumented intersection (see, e.g., Lioris et al. 2017) features in-pavement sensors, such as loop detectors or magnetic sensors, that detect the presence of vehicles at a stop bar. Additional sensors, those located on the approaches to the intersection and those in departure lanes exiting the intersection, can also be used to estimate the speed and turn movements of vehicles (Coogan et al. 2016). The signal phase and timing (referred to as *SPaT* and describing the current combination of signal phases and the remaining time until the next change of phase) can be retrieved directly from the controller, or indirectly via image analysis. Controllers can implement a fixed cycle, change green times depending on immediate traffic conditions, or implement more advanced control strategies adapting to the congestion level; pedestrians can be part of the cycle or make requests with buttons. Intersections can be instrumented to broadcast messages to nearby vehicles using DSRC. The SAE standard J2735 (Dedicated Short Range Communications

DSRC) includes messages for signal phase and timing and intersection geometry. Note that the timing part is deterministic only if the controller has a fixed cycle; otherwise, the timing is inherently uncertain, because of the stochastic nature of traffic.

CAV communication with unconnected vehicles, cyclists, and pedestrians happens via the connected roadside infrastructure, provided that it has necessary detection capabilities. When unconnected agents are detected, the roadside infrastructure broadcasts information about their presence in a given lane or at a crosswalk. Similarly, the warnings about approaching trains are broadcasted at rail crossings.

Beyond intersections and traffic signals, urban infrastructure includes fuel stations, charging stations, and parking infrastructure. A connected charging and parking infrastructure enables better routing of vehicles and more effective pricing schemes. Moreover since vehicle charging has a significant effect on grid balancing and smart grids (Clement-Nyns et al. 2010, 2011; Fan 2012), connectivity is expected to play a major role in grid-vehicle interaction.

Modern road and traffic map services provide information that goes beyond the maps for navigation, including *static* information, such as road grade, road curvature, location of intersections, lane maps, speed limits, location of fuel and charging stations, and intersection average delays, and *dynamic* information, such as traffic speed, availability and price of fuel and charging stations, intersection delays, traffic congestion, road works, and weather conditions. Historical data pinned to maps can be relevant for planning problems, such as vehicle routing and reference velocity generation. Examples include traffic congestion on freeways (California Department of Transportation) and signal phase and timing data (Ibrahim et al. 2017); in both cases, historical data give deeper insight on traffic patterns. CAVs can use cloud services to do heavy-duty computations remotely, thus relaxing the on-board computational requirements. Computations moved to the cloud may include (dynamic) routing and long-term trajectory optimization.

Connected and Automated Vehicles, Table 1 Categorization of CAV applications

	Safety	Traffic efficiency	Energy efficiency
Vehicle level	Balancing safety and performance in car following	CACC for throughput increase on freeways and at arterial intersections	Powertrain energy minimization
Roadside level	Intelligent intersection that ensures safe crossing for all travel modes	Platoon coordination at critical junctions to maximize road capacity utilization	SPaT prediction and speed advisory for minimizing energy consumption
Network level	Improving situational awareness at hazardous locations	Routing to minimize travel time and congestion	Routing to minimize energy consumption

Examples of CAV applications categorized by the level of information and control and by the area of impact are summarized in Table 1.

Control Systems for CAV

This section discusses relevant applications of control systems for CAV operation. We will briefly cover vehicle motion control, decision-making and motion planning, coordination of CAVs at junctions, and platoon coordination. We will not address route optimization problems here, as they are often solved in the cloud and have a wider application than just CAVs.

Motion Control

Motion control ensures that the vehicle's longitudinal and lateral motion follows a reference trajectory or path. The existing longitudinal control approaches are distinguished by their use of external information: predictive cruise control (PCC), ACC, CACC, and connected cruise control (CCC).

PCC tracks a reference velocity that is generated using preview information. The information can be *static* (e.g., road grade and speed limits) or *dynamic* but slowly changing (e.g., traffic speed).

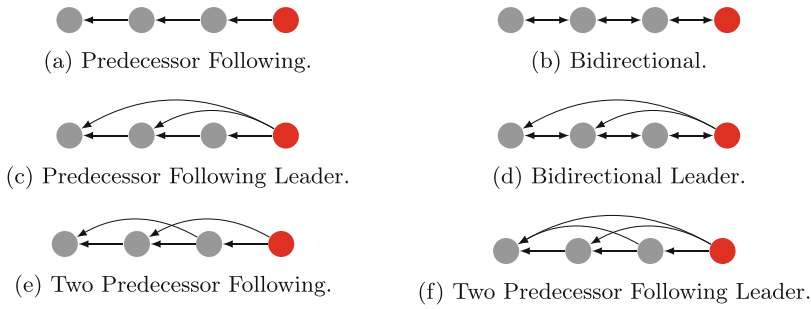
ACC detects any preceding vehicle and adjusts speed to avoid collisions. ACC is designed to enhance passengers' safety and comfort and to improve road throughput and energy efficiency.

CACC is an enhancement of ACC enabled by communication. The performance of ACC is limited by perception systems that can only measure the relative distance and velocity. V2V communication enables the exchange of vehicle

acceleration (and potentially of its forecast), which is important in dynamic driving scenarios. CACC exploits this additional piece of information to guarantee higher safety and smaller inter-vehicular distance. In addition to improved safety, this can translate into lower energy consumption, higher road throughput, and passenger comfort. For control analysis and synthesis, the multi-vehicle formation can be regarded as a one-dimensional networked dynamic system. Much research has been devoted to multi-agent consensus schemes, regarding CACC as a distributed control problem – see Li et al. (2017) for a detailed survey on CACC. Analysis, design, and synthesis can be addressed by classifying the CAV platoon problem based on:

- *Dynamics*, i.e., the dynamics of each CAV.
- *Information flow network*, i.e., the topology and quality of information flow and the type of information exchanged. Figure 2 depicts some typical communication topologies used in platooning. The information exchanged may just be the current velocity and acceleration or include forecasts thereof and information on lateral motion.
- *Local controller design* and its use of on-board information.
- *Formation geometry*, i.e., vehicle ordering, cruising speed, and inter-vehicle distance.

Note that the dynamics and the distributed controller pertain to the individual CAV, while the information flow network and the formation geometry are properties of the platoon. The latter two can be decided a priori in a specific demonstration but require some form of stan-



Connected and Automated Vehicles, Fig. 2 Information flow topologies in a four-vehicle platoon. The red node indicates the platoon leader. The nomenclature is borrowed from Li et al. (2017) (a) Predecessor Following.

(b) Bidirectional. (c) Predecessor Following Leader. (d) Bidirectional Leader. (e) Two Predecessor Following. (f) Two Predecessor Following Leader

dardization for operation on public roads. CACC can be viewed as a tracking control problem with forecast; as such it has been addressed with a variety of control techniques (Li et al. 2017). Linear consensus control and distributed robust control techniques enable insightful theoretical analysis and can provide guarantees of string stability (Naus et al. 2010).

CCC is essentially an extension of CACC that includes non-CAVs in the formation and relaxes the constraints on the information flow network. A significant difference with CACC is that CCC is not cooperative: the ego-CAV exploits the available V2X information to its advantage, not necessarily to the advantage of all the connected vehicles. We refer the interested reader to Orosz (2016), Zhang and Orosz (2016), and Ge and Orosz (2014) for a detailed theoretical analysis of CCC and to Orosz et al. (2017) for some experimental results.

Lateral control supervises the vehicle motion in the lateral direction, actuating the steering angle or torque. Generally, lateral controllers track a reference trajectory or path ensuring robustness in an uncertain and fast changing environment. Lateral control is tightly connected with the lateral stability of the vehicle, particularly in the presence of static and dynamic obstacles and during operation at the limits of friction (e.g., Funke et al. 2017). In addition to steering, lateral control can exploit differential braking and traction to influence the lateral motion of the vehicle (e.g., Park and Gerdes 2015). In vehicles

with a low degree of automation, lateral control can also just provide assistance to a human driver, implementing a shared control paradigm (Erlien et al. 2015; Ercan et al. 2018).

Decision-Making and Motion Planning

The decision-making control system controls the vehicle behavior (stay in the current lane or change, come to a stop at an intersection, yield to pedestrians, etc.) and on how to translate that behavior into a CAV trajectory. Decision-making for vehicle behavior can be implemented by heuristic rules. Since the intentions of other agents (aside from other CAVs) are uncertain, estimation, prediction, and learning techniques play a major role in this field (Paden et al. 2016).

The high-level specifications from the remote planning and the behavioral decision must then be translated into a path or trajectory for the CAV, taking into account the most recent state and predictions of the surrounding objects. In the literature, *path planning* generally indicates the problem of planning the future motion in the configuration space of the vehicle, while *trajectory planning* indicates the search of a time-parametrized solution (Paden et al. 2016; Katrakazas et al. 2015). Both problems are often formulated as an optimization problem, i.e., in terms of the minimization of some cost functional subject to constraints. Holonomic constraints include collision avoidance constraints and terminal constraints that define safe regions or *driving corridors* from the current to the target

configuration. In path planning, differential constraints are included to enforce some level of smoothness in the solution, for instance, on the path curvature. In trajectory planning, dynamic constraints can be included, and collision avoidance can in principle be enforced not only for static but also for dynamic obstacles.

Optimal path and trajectory planning are both PSPACE-hard in their general formulations. As such, past and current research have focused on computationally tractable approximations, or on methods that apply to specific scenarios. Computational methods for path planning include variational and optimization methods, graph search methods, and incremental search methods. See, respectively, Gerdts et al. (2009), Lima et al. (2017), Zhang et al. (2017), Plessen et al. (2017), Funke and Gerdes (2015), Cirillo (2017), and Kuwata et al. (2009) for some recent examples and Paden et al. (2016) for a detailed review. Trajectory planning can be solved using variational methods in the time domain, or converting it to a path planning problem with a time dimension (Paden et al. 2016).

In CAVs, communication can greatly improve the awareness of other agents and, in general, of the surrounding environment. In fact, CAVs can share not only their own states and predicted motion but also the obstacles that they detect; a distributed perception system can include multiple CAVs, roadside units, and potentially cyclists and pedestrians and can significantly outperform an advanced perception system that only uses on-board sensors. This potential is discussed in Katrakazas et al. (2015). Examples of motion planning applications using forecasts from communication can be found in Plessen et al. (2016, 2018) and Shen et al. (2015). V2V-based coordination (in both one-directional and bi-directional traffic flow) is addressed using optimization methods in Plessen et al. (2016, 2018) and using graph search methods in Shen et al. (2015).

Coordination of CAVs at Junctions

One of the simplest coordination between CAV and traffic lights is minimization of braking and stopping at red lights. A heuristic algorithm was

proposed in, e.g., Koukoumidis et al. (2012). Several optimization-based algorithms have also been proposed in the literature. The optimization goal may be to minimize travel time, reduce acceleration peaks, avoid idling at red lights, or directly minimize energy consumption. Computational approaches include dynamic programming, Dijkstra's shortest path algorithm, MPC, and genetic algorithms.

In an MPC formulation of energy consumption minimization when traveling on a street with signalized intersections, the problem is cast in the longitudinal position domain, as a finite-horizon optimal control problem of the following form (Sun et al. 2020):

$$\begin{aligned} & \underset{u_0, u_1, \dots, u_{N-1}}{\text{minimize}} \quad \sum_{k=0}^{N-1} h(x_k, u_k) \\ & \text{subject to} \quad \left. \begin{aligned} x_{k+1} &= f(x_k, u_k), \\ u_k &\in \mathcal{U}, \quad x_k \in \mathcal{X}, \end{aligned} \right\} k=0, 1, \dots, N-1, \\ & \quad x_0 = x_s, \quad x_N \in \mathcal{X}_N. \end{aligned}$$

We set the state vector as $x = [t, v]^T$ and the input vector as $u = [F_w, F_b]^T$, where t is the travel time, v is the vehicle speed, F_w is the wheel force, and F_b is the braking torque. The longitudinal dynamics is modeled as in Guzzella and Sciarretta (2013), projected into the position domain by the transformation

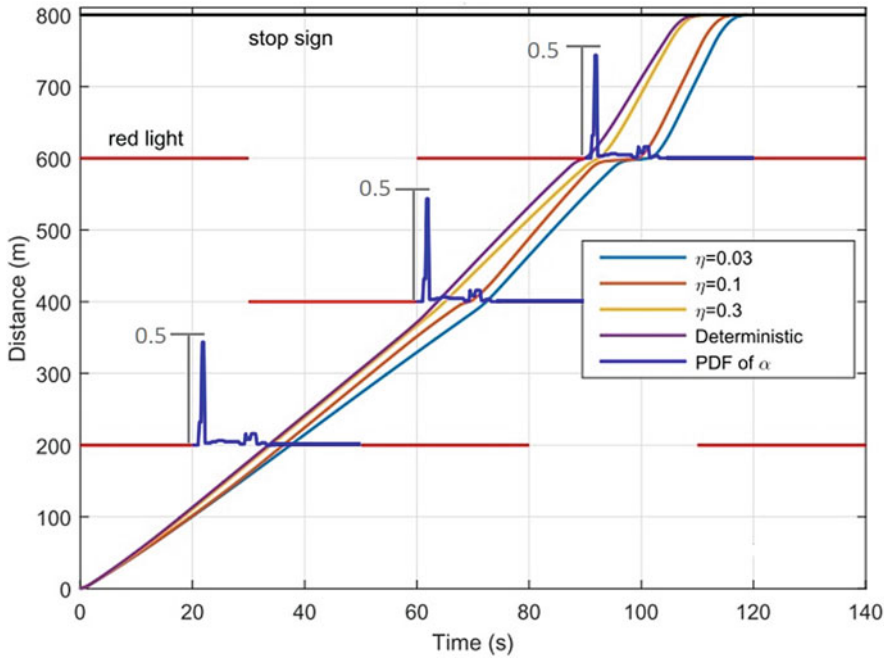
$$\frac{dv}{dt} = v \frac{dv}{ds},$$

and applies Euler discretization with step S_s , obtaining

$$f(x, u) = \begin{bmatrix} t + \frac{S_s}{v} \\ v + \frac{S_s}{Mv} (F_w - F_b - F_f) \end{bmatrix},$$

where M is the vehicle mass and

$$\begin{aligned} F_f &= Mg \sin \vartheta - Mg(C_r + C_v v) \cos \vartheta \\ &\quad - \frac{1}{2} \rho A C_x v^2, \end{aligned}$$



Connected and Automated Vehicles, Fig. 3 Eco-driving through signalized intersections. (Source: Sun et al. 2020)

g is the gravity constant, ϑ is the (position-dependent) road grade, C_r is the rolling coefficient, C_v is the viscous friction coefficient, ρ is the air density, A is the front area, and C_x is the air drag coefficient. Assuming a fuel-powered vehicle, the cost function is set to the fuel rate $h(x, u) = \dot{m}_f(v, F_w)/v$.

The convex input constraint set $\mathcal{U}$ defines the actuator limits, while the state constraint set describes the surrounding environment. A convex set $\mathcal{X}_1$ models bounds on the speed and the acceleration. The formulation above can accommodate N_s signalized intersections, assuming they can be approximated as points along the route. We assume that every traffic signal has an independent cycle time $c^i \in [0, \bar{c}^i]$, $i = 1, \dots, N_s$, where $c^i = 0$ denotes the beginning of the red light phase and $\bar{c}^i \in \mathbb{R}^+$ is the cycle period. We denote the red light phase duration by $c_r^i \in (0, \bar{c}^i)$ and by t_p^i the time at which the CAV passes through intersection i . In the domain of the intersection cycle time c_i , the passing time is computed as $c_p^i = (c_0^i + t_p^i) \bmod \bar{c}^i$, where c_0^i is the cycle time at $s = 0$. We enforce that intersections are not crossed during red light phases

by $\mathcal{X}_2 = \{t : c_p^i(t) \geq c_r^i, \forall i = 1, \dots, N_s\}$. To summarize, the state constraint set is given by $\mathcal{X} = \mathcal{X}_1 \cap \mathcal{X}_2$. The forecasts of the red phase durations c_r^i is obtained from the SPaT information broadcasted by the connected intersections.

In case some or all of these forecasts are unavailable, one could replace $\mathcal{X}_2$ with a chance constraint

$$\mathcal{X}_3 = \left\{t : \Pr \left[c_p^i(t) \geq c_r^i + \alpha^i \right] \geq 1 - \eta^i, \forall i = 1, \dots, N_s \right\},$$

where $\Pr[A]$ is the probability of event A , $\alpha^i \in [0, \bar{c}^i - c_r^i]$ models the adaptation of the red light phase, and $\eta^i \in [0, 1]$ is the level of constraint enforcement.

Figure 3 shows the solutions to the deterministic problem (i.e., enforcing $\mathcal{X} = \mathcal{X}_1 \cap \mathcal{X}_2$) and to some instances of the robust problem (i.e., enforcing $\mathcal{X} = \mathcal{X}_1 \cap \mathcal{X}_3$ for different values of η^i). There are three intersections here ($N_s=3$) denoted by the red lines that represent red light periods. For each intersection, the blue

line depicts a probability distribution of α_i in the green time period. It can be noted that the deterministic solution crosses the third intersection very close to a phase switching (red to green); conversely, the robust solutions show different levels of conservatism, which can be adjusted by tuning η^i . We refer the reader to Sun et al. (2020) for an extensive analysis, implementation details, and approaches to define X_3 based on historical signal timing data.

Platoon Coordination

Longitudinal control of platoons is mostly regarded as a distributed control problem, but some basic centralized coordination is still required; in particular, the formation geometry (inter-vehicular spacing policy, cruising speed, vehicle ordering) and the information flow network (communication topology and quality, communication protocol) affect platoon safety and performance. These aspects have not been standardized to date, and due to the heterogeneity of vehicles, sensors, actuators, and communication technologies available on the market, they could reasonably be coordinated remotely on a case-by-case basis. For instance, some communication topologies listed in Fig. 2 may not be feasible, depending on the number and the specific instrumentation of the vehicles involved in the platoon. Another example is the choice of the inter-vehicle spacing policy, which depends on a number of factors, including the dynamics and imperfections of sensors, actuators, and wireless communication.

For a formed platoon, motion is mostly longitudinal within a lane, although lateral motion is needed for lane changes and collision avoidance. Platoon management (formation and dissolution) is a closely related problem which also requires coordination. A CAV that joins or leaves a platoon needs authority on both longitudinal and lateral motion, while the vehicles in the platoon may only move longitudinally to open or close a gap. Merging of (or splitting into) two platoons is not fundamentally different; instead of just one gap, there may be multiple gaps to open or close. Merging is a critical maneuver because the trail vehicle or platoon must temporarily move faster

than the lead platoon, and with a smaller distance gap, a sudden deceleration of the lead platoon may result in a collision at high speed. Thus, their relative velocities must be kept bounded to avoid dangerous collisions. An approach for platoon merging, splitting, and lane changing is presented in Frankel et al. (2007).

When the platooning objective is energy efficiency, the main question is whether the energy saved by the platoon exceeds the energy cost to form the platoon. A problem at a slightly higher level is the clustering of vehicles into platoons, based on their routes and departure and arrival times. An opportunity lies in the coordination of vehicle fleets: when origins, destinations, and time constraints are known in advance, a centralized planner can be set up, aggregating vehicles in platoons and thereby maximizing the overall fuel economy.

The idea of optimizing individual CAV dynamics using signal phase duration forecasts on a street with signalized intersections can be extended to platoons. An MPC framework for groups of CAVs with a decentralized approach, where every CAV only receives information from neighboring vehicles and traffic signals, is discussed in HomChaudhuri et al. (2017).

Summary and Future Directions

We conclude this short review about CAVs by listing some CAV research questions that are presently open and present a challenge:

- Platoon organization. Present platoon experiments are conducted under the assumption that addresses of all present CAVs and their places in the platoon are known. If this assumption is relaxed, how do CAVs find each other and agree on the platoon structure in multi-lane environment? This is especially difficult to do in a congested traffic.
- A proper detection or a correct estimation of a vehicle queue length at an urban intersection and a freeway on-ramp is an important component of effective traffic management as well as efficient operation of individual vehicles.

CAV coordination and performance optimization at road junctions are impossible without accurate and robust queue estimation. Queue estimation can be difficult, especially in the congested environment when traffic spillbacks to upstream junctions occur, and it becomes unclear where one queue ends and the other begins. Another challenge arises when multiple queues form on a single approach, e.g., when part of the traffic is queued up to make a left turn and the other part of the traffic waits for a straight movement.

- Solving the two problems above and adopting the related technology would enable the implementation of safe and efficient intersections for all connected road users, including cyclists and pedestrians. This implementation would include identification of travelers' blind zones and information exchange about agent presence in those blind zones, as well as signal optimization for groups of multimodal travelers – technologies that need to be developed.

Cross-References

- [Controlling Collective Behavior in Complex Systems](#)
- [Multi-vehicle Routing](#)
- [Routing in Transportation Networks](#)

Bibliography

- Alam N, Tabatabaei Balaei A, Dempster AG (2011) A DSRC doppler-based cooperative positioning enhancement for vehicular networks with GPS availability. *IEEE Trans Veh Technol* 60(9):4462–4470
- California Department of Transportation. PeMS homepage. <http://pems.dot.ca.gov>. Accessed on 2020
- Cirillo M (2017) From videogames to autonomous trucks: a new algorithm for lattice-based motion planning. In: 2017 IEEE intelligent vehicles symposium, pp 148–153
- Clement-Nyns K, Haesen E, Driesen J (2010) The impact of Charging plug-in hybrid electric vehicles on a residential distribution grid. *IEEE Trans Power Syst* 25(1):371–380
- Clement-Nyns K, Haesen E, Driesen J (2011) The impact of vehicle-to-grid on the distribution grid. *Electr Power Syst Res* 81(1):185–192
- Coogan S, Muralidharan A, Flores C, Varaiya P (2016) Management of intersections with multi-modal high-resolution data. *Transp Res Part C* 68:101–112
- Dedicated Short Range Communications (DSRC) Message Set Dictionary (2016)
- Ercan Z, Carvalho A, Tseng HE, Gökaşan M, Borrelli F (2018) A predictive control framework for torque-based steering assistance to improve safety in highway driving. *Veh Syst Dyn* 56(5):810–831
- Erlien SM, Fujita S, Gerdes JC (2015) Shared steering control using safe driving envelopes for obstacle avoidance and stability. *IEEE Trans Intell Transp Syst* 17:441–451
- Fan Z (2012) A distributed demand response algorithm and its application to PHEV charging in smart grids. *IEEE Trans Smart Grid* 3(3):1280–1290
- Frankel J, Alvarez L, Horowitz R, Li P (2007) Safety oriented maneuvers for IVHS. *Veh Syst Dyn* 26(4):271–299
- Funke J, Gerdes JC (2015) Simple clothoid lane change trajectories for automated vehicles incorporating friction constraints. *J Dyn Syst Meas Control* 138(2):021002
- Funke J, Brown M, Erlien SM, Gerdes JC (2017) Collision avoidance and stabilization for autonomous vehicles in emergency scenarios. *IEEE Trans Control Syst Technol* 25(4):1204–1216
- Ge JI, Orosz G (2014) Dynamics of connected vehicle systems with delayed acceleration feedback. *Transp Res Part C Emerg Technol* 46:46–64
- Gerdts M, Karrenberg S, Müller-Beler B, Stock G (2009) Generating locally optimal trajectories for an automatically driven car. *Optim Eng* 10(4):439–463
- Gibson D, Mills MK, Klein LA (2007) A new look at sensors. *Public Roads* 71:28–35
- Guzzella L, Sciarretta A (2013) Vehicle propulsion systems. Springer, Berlin
- HomChaudhuri B, Vahidi A, Pisu P (2017) Fast model predictive control-based fuel efficient control strategy for a group of connected vehicles in urban road conditions. *IEEE Trans Control Syst Technol* 25(2):760–767
- Ibrahim S, Kalathil D, Sanchez RO, Varaiya P (2017) Estimating phase duration for SPaT messages
- Katrakazas C, Quddus M, Chen W-H, Deka L (2015) Real-time motion planning methods for autonomous on-road driving: state-of-the-art and future research directions. *Transp Res Part C Emerg Technol* 60:416–442
- Koukoumidis E, Martonosi M, Peh LS (2012) Leveraging smartphone cameras for collaborative road advisories. *IEEE Trans Mobile Comput* 11(5):707–723
- Kuwata Y, Teo J, Fiore G, Karaman S, Frazzoli E, How JP (2009) Real-time motion planning with applications to autonomous urban driving. *IEEE Trans Control Syst Technol* 17(5):1105–1118
- Li SE, Zheng Y, Li K, Wu Y, Hedrick JK, Gao F, Zhang H (2017) Dynamical modeling and distributed control of connected and automated vehicles: challenges and opportunities. *IEEE Intell Transp Syst Mag* 9(3):46–58

- Lima PF, Oliveira R, Mårtensson J, Wahlberg B (2017) Minimizing long vehicles overhang exceeding the drivable surface via convex path optimization. In: 20th IEEE conference on intelligent transportation systems, pp 1374–1381
- Lioris J, Pedarsani R, Tascikaraoglu FY, Varaiya P (2017) Platoons of connected vehicles can double throughput in urban roads. *Transp Res Part C Emerg Technol* 77:292–305
- Naus GJL, Vugts RPA, Ploeg J, Van De Molengraft MJG, Steinbuch M (2010) String-stable CACC design and experimental validation: a frequency-domain approach. *IEEE Trans Veh Technol* 59(9):4268–4279
- Orosz G (2016) Connected cruise control: modelling, delay effects, and nonlinear behaviour. *Veh Syst Dyn* 54(8):1147–1176
- Orosz G, Ge JI, He CR, Avedisov SS, Qin WB, Zhang L (2017) Seeing beyond the line of sight: controlling connected automated vehicles. *ASME MEch Eng Mag* 139(12):S8–S12
- Paden B, Cap M, Yong SZ, Yershov D, Frazzoli E (2016) A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Trans Intell Veh* 1(1):33–55
- Park H, Gerdes JC (2015) Optimal tire force allocation for trajectory tracking with an over-actuated vehicle. In: IEEE intelligent vehicles symposium, pp 1032–1037
- Peloton Technology. <https://peloton-tech.com>. Accessed on 2020
- Plessen MG, Bernardini D, Esen H, Bemporad A (2016) Multi-automated vehicle coordination using decoupled prioritized path planning for multi-lane one- and bi-directional traffic flow control. In: 55th IEEE conference on decision and control, pp 1582–1588
- Plessen MG, Lima PF, Mårtensson J, Bemporad A, Wahlberg B (2017) Trajectory planning under vehicle dimension constraints using sequential linear programming
- Plessen MG, Bernardini D, Esen H, Bemporad A (2018) Spatial-based predictive control and geometric corridor planning for adaptive cruise control coupled with obstacle avoidance. *IEEE Trans Control Syst Technol* 26(1):38–50
- Shen X, Chong ZJ, Pendleton S, Liu W, Qin B, Fu GMJ, Ang MH (2015) Multi-vehicle motion coordination using V2V communication. In: IEEE intelligent vehicles symposium, pp 1334–1341
- Sun C, Guanetti J, Borrelli F, Moura SJ (2020) Optimal eco-driving control of connected and autonomous vehicles through signalized intersections. *IEEE Internet Things J* 7(5):3759–3773
- Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems (2014)
- van de Sluis J, Baijer O, Chen L, Bengtsson HH, Garcia-Sol L, Balaguer P (2015) Proposal for extended message set for supervised automated driving. Technical report
- Zhang L, Orosz G (2016) Motif-based design for connected vehicle systems in presence of heterogeneous connectivity structures and time delays. *IEEE Trans Intell Transp Syst* 17(6):1638–1651
- Zhang X, Liniger A, Borrelli F (2017) Optimization-based collision avoidance

Consensus of Complex Multi-agent Systems

Fabio Fagnani

Dipartimento di Scienze Matematiche ‘G.L. Lagrange’, Politecnico di Torino, Torino, Italy

Abstract

This article provides a broad overview of the basic elements of consensus dynamics. It describes the classical Perron-Frobenius theorem that provides the main theoretical tool to study the convergence properties of such systems. Classes of consensus models that are treated include simple random walks on grid-like graphs and in graphs with a bottleneck, consensus on graphs with intermittently randomly appearing edges between nodes (gossip models), and models with nodes that do not modify their state over time (stubborn agent models). Application to cooperative control, sensor networks, and socio-economic models are mentioned.

Keywords

Consensus dynamics · Multi-agent system · Gossip model · Stubborn agent

Introduction

Multi-agent systems constitute one of the fundamental paradigms of the science and technology of present century (Castellano et al. 2009; Strogatz 2003). The main idea is that of creating complex dynamical evolutions from the interactions of many simple units. Indeed such collective behaviors are quite evident in biological and social systems and were indeed considered in earlier times. More recently, the

digital revolution and the miniaturization in electronics have made possible the creation of man-made complex architectures of interconnected simple devices (computers, sensors, cameras). Moreover, the creation of Internet has opened totally new form of social and economic aggregation. This has strongly pushed toward a systematic and deep study of multi-agent dynamical systems. Mathematically they typically consist of a graph where each node possesses a state variable; states are coupled at the dynamical level through dependences determined by the edges in the graph. One of the challenging problems in the field of multi-agent systems is to analyze the emergence of complex collective phenomena from the interactions of the units which are typically quite simple. Complexity is typically the outcome of the topology and the nature of interconnections.

Consensus dynamics (also known as average dynamics) (Jadbabaie and Morse 2003; Carli et al. 2008) is one of the most popular and simplest multi-agent dynamics. One convenient way to introduce it is with the language of social sciences. Imagine that a number of independent units possess an information represented by a real number, for instance, such number can represent an opinion on a given fact. Units interact and change their opinion by averaging with the opinions of other units. Under certain assumptions this will lead the all community to converge to a consensus opinion which takes into consideration all the initial opinion of the agents. In social sciences, empiric evidences (Galton 1907) have shown how such aggregate opinion may give a very good estimation of unknown quantities: such phenomenon has been proposed in the literature as wisdom of crowds (Surowiecki 2007).

Consensus Dynamics, Graphs, and Stochastic Matrices

Mathematically, consensus dynamics are special linear dynamical systems of type

$$x(t+1) = Px(t) \quad (1)$$

where $x(t) \in \mathbb{R}^{\mathcal{V}}$ and $P \in \mathbb{R}^{\mathcal{V} \times \mathcal{V}}$ are *stochastic matrix* (e.g., a matrix with non-negative elements

such that every row sums to 1). $\mathcal{V}$ represents the finite set of units (agents) in the network, and $x(t)_v$ is to be interpreted as the state (opinion) of agent v at time t . Equation (1) implies that states of agents at time $t+1$ are convex combinations of the components of $x(t)$: this motivates the term averaging dynamics. Stochastic matrices owe their name to their use in probability: the term P_{vw} can be interpreted as the probability of making a jump in the graph from state v to state w . In this way you construct what is called a random walk on the graph $\mathcal{G}$.

The network structure is hidden in the non-zero pattern of P . Indeed we can associate to P a graph: $\mathcal{G}_P = (\mathcal{V}, \mathcal{E}_P)$ where the set of edges is given by $\mathcal{E}_P := \{(u, v) \in \mathcal{V} \times \mathcal{V} \mid P_{uv} > 0\}$. Elements in $\mathcal{E}_P$ represent the communication edges among the units: if $(u, v) \in \mathcal{E}_P$ it means that unit u has access to the state of unit v . Denote by $\mathbb{1} \in \mathbb{R}^{\mathcal{V}}$ the all 1's vector. Notice that $P\mathbb{1} = \mathbb{1}$: this shows that once the states of units are at consensus, they will no longer evolve. Will the dynamics always converge to a consensus point?

Remarkably, some of the key properties of P responsible for the transient and asymptotic behavior of the linear system (1) are determined by the connectivity properties of the underlying graph $\mathcal{G}_P$. We recall that, given two vertices $u, v \in \mathcal{V}$, a *path* (of length l) from u to v in $\mathcal{G}_P$ is any sequence of vertices $u = u_1, u_2, \dots, u_{l+1} = v$ such that $(u_i, u_{i+1}) \in \mathcal{E}_P$ for every $i = 1, \dots, l$. $\mathcal{G}_P$ is said to be *strongly connected* if for any pair of vertices $u \neq v$ in $\mathcal{V}$ there is a path in $\mathcal{G}_P$ connecting u to v . The *period* of a node u is defined as the greatest common divisor of the lengths of all closed paths from u to u . In strongly connected graph, all nodes have the same period, and the graph is called *aperiodic* if such a period is 1. If x is a vector, we will use the notation x^* to denote its transpose. If A is a finite set, $|A|$ denotes the number of elements in A . The following classical result holds true (Gantmacher 1959):

Theorem 1 (Perron-Frobenius) *Assume that $P \in \mathbb{R}^{\mathcal{V} \times \mathcal{V}}$ is such that $\mathcal{G}_P$ is strongly connected and aperiodic. Then:*

1. 1 is an algebraically simple eigenvalue of P .
2. There exists a (unique) probability vector $\pi \in \mathbb{R}^{\mathcal{V}}$ ($\pi_v > 0$ for all v and $\sum_v \pi_v = 1$) which is a left eigenvector for P , namely, $\pi^* P = \pi^*$.
3. All the remaining eigenvalues of P are of modulus strictly less than 1.

A straightforward linear algebra consequence of this result is that $P^t \rightarrow \mathbb{1}\pi^*$ for $t \rightarrow +\infty$. This yields

$$\lim_{t \rightarrow +\infty} x(t) = \lim_{t \rightarrow +\infty} P^t x(0) = \mathbb{1}(\pi^* x(0)) \quad (2)$$

All agents' states are thus converging to the common value $\pi^* x(0)$, called *consensus point* which is a convex combination of the initial states with weights given by the invariant probability components.

If π is the uniform vector (i.e., $\pi_v = |\mathcal{V}|^{-1}$ for all units v), the common asymptotic value is simply the arithmetic mean of the initial states: all agents equally contribute to the final common state. A special case when this happens is when P is symmetric. The distributed computation of the arithmetic mean is an important step to solve estimation problems for sensor networks. As a specific example, consider the situation where there are N sensors deployed in a certain area and each of them makes a noisy measurement of a physical quantity x . Let $y_v = x + \omega_v$ be the measure obtained by sensor v , where ω_v is a zero-mean Gaussian noise. It is well-known that if noises are independent and identically distributed, the optimal mean square estimator of the quantity x given the entire set of measurements $\{y_v\}$ is exactly given by $\hat{x} = N^{-1} \sum_v y_v$. Other fields of application is in the control of cooperative autonomous vehicles (Fax and Murray 2004; Jadbabaie and Morse 2003).

Basic linear algebra allows to study the rate of convergence to consensus. Indeed, if $\mathcal{G}_P$ is strongly connected and aperiodic, the matrix P has all its eigenvalues in the unit ball: 1 is the only eigenvalue with modulo equal to 1, while all the others have modulo strictly less than 1. If we denote by $\rho_2 < 1$ the largest modulo of such eigenvalues (different from 1), we can show that

$x(t) - \mathbb{1}(\pi^* x(0))$ converges exponentially to 0 as ρ_2^t . In the following we will briefly refer to ρ_2 as to the *second eigenvalue* of P .

Examples and Large-Scale Analysis

In this section we present some classical examples. Consider a strongly connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The *adjacency matrix* of $\mathcal{G}$ is a square matrix $A_{\mathcal{G}} \in \{0, 1\}^{\mathcal{V} \times \mathcal{V}}$ such that $(A_{\mathcal{G}})_{uv} = 1$ iff $(u, v) \in \mathcal{E}$. $\mathcal{G}$ is said to be symmetric (also called undirected in the literature) if $A_{\mathcal{G}}$ is symmetric. Given a symmetric graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we can consider the stochastic matrix P given by $P_{uv} = d_u^{-1}(A_{\mathcal{G}})_{uv}$ where $d_u = \sum_v (A_{\mathcal{G}})_{uv}$ is the *degree* of node u . P is called the *simple random walk* (SRW) on $\mathcal{G}$: each agent gives the same weight to the state of its neighbors. Clearly, $\mathcal{G}_P = \mathcal{G}$. A simple check shows that $\pi_v = d_v/|\mathcal{E}|$ is the invariant probability for P . The consensus point is given by

$$\pi^* x(0) = |\mathcal{E}|^{-1} \sum_v d_v x(0)_v$$

Each node contributes with its initial state to this consensus with a weight which is proportional to the degree of the node. Notice that the SRW P is symmetric iff the graph is regular, namely, all units have the same degree.

We now present a number of classical examples based on families of graphs with larger and larger number of nodes N . In this setting, particularly relevant is to understand the behavior of the second eigenvalue ρ_2 as a function of N . Typically, one considers $\epsilon > 0$ fixed and solves the equation $\rho_2^t = \epsilon$. The solution $\tau = (\ln \rho_2^{-1})^{-1} \ln \epsilon^{-1}$ will be called the *convergence time*: it essentially represents the time needed to shrink of a factor ϵ the distance to consensus. Dependence of ρ_2 on N will also yield that τ will be a function of N . In the sequel we will investigate such dependence for SRWs on certain classical families of graphs.

Example 1 (SRW on a complete graph) Consider the complete graph on the set $\mathcal{V}$: $K_{\mathcal{V}} := (\mathcal{V}, \mathcal{V} \times \mathcal{V})$

(also self-loops are present). The SRW on $K_{\mathcal{V}}$ is simply given by $P = N^{-1} \mathbb{1} \mathbb{1}^*$ where $N = |\mathcal{V}|$. Clearly, $\pi = N^{-1} \mathbb{1}$. Eigenvalues of P are 1 with multiplicity 1 and 0 with multiplicity $N - 1$. Therefore, $\rho_2 = 0$. Consensus in this case is achieved in just one step: $x(t) = N^{-1} \mathbb{1} \mathbb{1}^* x(0)$ for all $t \geq 1$.

Example 2 (SRW on a cycle graph) Consider the symmetric cycle graph $C_N = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{0, \dots, N - 1\}$ and $\mathcal{E} = \{(v, v + 1), (v + 1, v)\}$ where sum is mod N . The graph C_N is clearly strongly connected and is also aperiodic if N is odd. The corresponding SRW P has eigenvalues

$$\lambda_k = \cos \frac{2\pi k}{N}$$

Therefore, if N is odd, the second eigenvalue is given by

$$\begin{aligned} \rho_2 &= \cos \frac{2\pi}{N} \\ &= 1 - 2\pi^2 \frac{1}{N^2} + o(N^{-2}) \quad \text{for } N \rightarrow +\infty \end{aligned} \quad (3)$$

while the corresponding convergence time is given by

$$\tau = (\ln \rho_2^{-1})^{-1} \ln \epsilon^{-1} \asymp N^2 \quad \text{for } N \rightarrow +\infty$$

Example 3 (SRW on toroidal grids) The toroidal d -grids C_n^d are formally obtained as a product of cycle graphs. The number of nodes is $N = n^d$. It can be shown that the convergence time behaves as

$$\tau \asymp N^{2/d} \quad \text{for } N \rightarrow +\infty$$

Convergence time exhibits a slower growth in N as the dimension d of the grid increases: this is intuitive since the increase in d determines a better connectivity of the graph and a consequently faster diffusion of information.

For a general stochastic matrix (even for SRW on general graphs), the computation of the second eigenvalue is not possible in closed form and can actually be also difficult from a numerical point of view. It is therefore important to develop

tools for efficient estimation. One of these is based on the concept of bottleneck: if a graph can be split into two loosely connected parts, then consensus dynamics will necessarily exhibit a slow convergence.

Formally, given a symmetric graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a subset of nodes $S \subseteq \mathcal{V}$, define e_S as the number of edges with at least one node in S and e_{SS} as the number of edges with both nodes in S . The bottleneck of S in G is defined as $\Phi(S) = e_{SS}/e_S$. Finally, the *bottleneck ratio* of $\mathcal{G}$ is defined as

$$\Phi_* := \min_{S: e_S/e \leq 1/2} \Phi(S)$$

where $e = |\mathcal{E}|$ is the number of edges in the graph.

Let P be the SRW on $\mathcal{G}$ and let ρ_2 be its second eigenvalue. Then,

Proposition 1 (Cheeger bound Levin et al. 2008)

$$1 - \rho_2 \leq 2\Phi_*. \quad (4)$$

Example 4 (Graphs with a bottleneck) Consider two complete graphs with n nodes connected by just one edge. If S is the set of nodes of one of the two complete graphs, we obtain

$$\Phi(S) = \frac{1}{n^2 + 1}$$

Bound (4) implies that the convergence time is at least of the order of n^2 in spite of the fact that in each complete graph, convergence would be in finite time!

Other Models

The systems studied so far are based on the assumptions that units all behave the same – they share a common clock and update their state in a synchronous fashion. In this section, we discuss more general models.

Random Consensus Models

Regarding the assumption of synchronicity, it turns out to be unfeasible in many contexts. For

instance, in the opinion dynamics modelling, it is not realistic to assume that all interactions happen at the same time: agents are embedded in a physical continuous time, and interactions can be imagined to take place at random, for instance, in a pairwise fashion.

One of the most famous random consensus model is the gossip model. Fix a real number $q \in (0, 1)$ and a symmetric graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. At every time instant t , an edge $(u, v) \in \mathcal{E}$ is activated with uniform probability $|\mathcal{E}|^{-1}$, and nodes u and v exchange their states and produce a new state according to the equations

$$\begin{aligned} x_u(t+1) &= (1-q)x_u(t) + qx_v(t) \\ x_v(t+1) &= qx_u(t) + (1-q)x_v(t) \end{aligned}$$

The states of the other units remain unchanged.

Will this dynamics lead to a consensus? If the same edge is activated at every time instant, clearly consensus will not be achieved. However, it can be shown that, with probability one, consensus will be reached (Boyd et al. 2006).

Consensus Dynamics with Stubborn Agents

In this chapter, we investigate consensus dynamics models where some of the agents do not modify their own state (stubborn agents). These systems are of interest in socioeconomic models (Acemoglu et al. 2013).

Consider a symmetric connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We assume a splitting $\mathcal{V} = \mathcal{S} \cup \mathcal{R}$ with the understanding that agents in $\mathcal{S}$ are *stubborn* agents not changing their state while those in $\mathcal{R}$ are *regular* agents whose state modifies in time according to a SRW consensus dynamics, namely,

$$x_u(t+1) = \frac{1}{d_u} \sum_{v \in \mathcal{V}} (A_{\mathcal{G}})_{uv} x_v(t), \quad \forall u \in \mathcal{R}$$

By assembling the state of the regular and of the stubborn agents in vectors denoted, respectively, as $x^{\mathcal{R}}(t)$ and $x^{\mathcal{S}}(t)$, dynamics can be recasted in matrix form as

$$\begin{aligned} x^{\mathcal{R}}(t+1) &= Q^{11} x^{\mathcal{R}}(t) + Q^{12} x^{\mathcal{S}}(t) \\ x^{\mathcal{S}}(t+1) &= x^{\mathcal{S}}(t) \end{aligned} \quad (5)$$

It can be proven that Q^{11} is asymptotically stable ($(Q^{11})^t \rightarrow 0$). Henceforth, $x^{\mathcal{R}}(t) \rightarrow x^{\mathcal{R}}(\infty)$ for $t \rightarrow +\infty$ with the limit opinions satisfying the relation

$$x^{\mathcal{R}}(\infty) = Q^{11} x^{\mathcal{R}}(\infty) + Q^{12} x^{\mathcal{S}}(0) \quad (6)$$

If we define $\mathcal{E} := (I - Q^{11})^{-1} Q^{12}$, we can write $x^{\mathcal{R}}(\infty) = \mathcal{E} x^{\mathcal{S}}(0)$. It is easy to see that $\mathcal{E}$ has non-negative elements and that $\sum_s \mathcal{E}_{us} = 1$ for all $u \in \mathcal{R}$: asymptotic opinions of regular agents are thus convex combinations of the opinions of stubborn agents. If all stubborn agents are in the same state x , then consensus is reached by all agents in the point x . However, typically, consensus is not reached in such a system: we will discuss an example below.

There is a useful alternative interpretation of the asymptotic opinions. Interpreting the graph $\mathcal{G}$ as an electrical circuit where edges are unit resistors, relation (6) can be seen as a Laplace-type equation on the graph $\mathcal{G}$ with boundary conditions given by assigning the voltage $x^{\mathcal{S}}(0)$ to the stubborn agents. In this way $x^{\mathcal{R}}(\infty)$ can be interpreted as the vector of voltages of the regular agents when stubborn agents have fixed voltage $x^{\mathcal{S}}(0)$. Thanks to the electrical analogy, we can compute the asymptotic opinion of the agents by computing the voltages in the various nodes in the graph. We propose a concrete application in the following example:

Example 5 (Stubborn agents in a line graph)

Consider the line graph $L_N = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, 2, \dots, N\}$ and where $\mathcal{E} = \{(u, u+1), (u+1, u) \mid u = 1, \dots, N-1\}$. Assume that $\mathcal{S} = \{1, N\}$ and $\mathcal{R} = \mathcal{V} \setminus \mathcal{S}$. Consider the graph as an electrical circuit. Replacing the line with a single edge connecting 1 and N having resistance $N-1$ and applying Ohm's law, we obtain that the current flowing from 1 to N is equal to $\Phi = (N-1)^{-1} [x_N^{\mathcal{S}}(0) - x_1^{\mathcal{S}}(0)]$. If we now fix an arbitrary node $v \in \mathcal{V}$ and applying again

the same arguments in the part of graph from 1 to v , we obtain that the voltage at v , $x_v^{\mathcal{R}}(\infty)$ satisfies the relation $x_v^{\mathcal{R}}(\infty) - x_1^{\mathcal{S}}(0) = \Phi(v-1)$. We thus obtain

$$x_v^{\mathcal{R}}(\infty) = x_1^{\mathcal{S}}(0) + \frac{v-1}{N-1} [x_N^{\mathcal{S}}(0) - x_1^{\mathcal{S}}(0)].$$

In Acemoglu et al. (2013) further examples are discussed showing how, because of the topology of the graph, different asymptotic configurations may show up. While in graphs presenting bottlenecks polarization phenomena can be recorded, in graphs where the convergence rate is low, there will be a typical asymptotic opinion shared by most of the regular agents.

Summary and Future Directions

We have presented an overview of linear consensus multiagent dynamics over general networks encompassing deterministic and random models. We have also briefly analyzed models in the presence of stubborn agents.

A number of other directions, not considered in the present paper, are worth of analysis. Among them are inferential models in the presence of heterogeneous agents, second-order models, nonlinear models.

Cross-References

- [Averaging Algorithms and Consensus](#)
- [Dynamical Social Networks](#)
- [Information-Based Multi-agent Systems](#)
- [Noisy Channel Effects on Multi-agent Consensus](#)

Acknowledgments This work was partially supported by MIUR grant Dipartimenti di Eccellenza 2018-2022 [CUP: E11G18000350001]

Bibliography

Acemoglu D, Como G, Fagnani F, Ozdaglar A (2013) Opinion fluctuations and disagreement in social networks. *Math Oper Res* 38(1):1–27

- Boyd S, Ghosh A, Prabhakar B, Shah D (2006) Randomized gossip algorithms. *IEEE Trans Inf Theory* 52(6):2508–2530
- Carli R, Fagnani F, Speranzon A, Zampieri S (2008) Communication constraints in the average consensus problem. *Automatica* 44(3):671–684
- Castellano C, Fortunato S, Loreto V (2009) Statistical physics of social dynamics. *Rev Mod Phys* 81:591–646
- Fax JA, Murray RM (2004) Information flow and cooperative control of vehicle formations. *IEEE Trans Autom Control* 49(9):1465–1476
- Galton F (1907) Vox populi. *Nature* 75:450–451
- Gantmacher FR (1959) The theory of matrices. Chelsea Publishers, New York
- Jadbabaie A, Lin J, Morse AS (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans Autom Control* 48(6):988–1001
- Levin DA, Peres Y, Wilmer EL (2008) Markov chains and mixing times. AMS, Providence
- Strogatz SH (2003) Sync: the emerging science of spontaneous order. Hyperion, New York
- Surowiecki J (2007) The wisdom of crowds: why the many are smarter than the few and how collective wisdom shapes business, economies, societies and nations. Little, Brown, 2004. Traduzione italiana: La saggezza della folla, Fusi Orari

Control and Optimization of Batch Processes

Dominique Bonvin
Laboratoire d'Automatique, École
Polytechnique Fédérale de Lausanne (EPFL),
1015 Lausanne, Switzerland

Abstract

A batch process is characterized by the repetition of time-varying operations of finite duration. Due to the repetition, there are two independent “time” variables, namely, the run time during a batch and the batch counter. Accordingly, the control and optimization objectives can be defined for a given batch or over several batches. This entry describes the various control and optimization strategies available for the operation of batch processes. These include conventional feedback control, predictive control, iterative learning control,

and run-to-run control on the one hand and model-based repeated optimization and model-free self-optimizing schemes on the other.

Keywords

Batch control · Batch process optimization · Dynamic optimization · Iterative learning control · Run-to-run control · Run-to-run optimization

Introduction

Batch processing is widely used in the manufacturing of goods and commodity products, in particular in the chemical, pharmaceutical, and food industries. These industries account for several billion US dollars in annual sales. Batch operation differs significantly from continuous operation. While in continuous operation the process is maintained at an economically desirable operating point, the process evolves continuously from an initial to a final time in batch processing. In the chemical industry, for example, since the design of a continuous plant requires substantial engineering effort, continuous operation is rarely used for low-volume production. Discontinuous operations can be of the batch or semi-batch type. In batch operations, the products to be processed are loaded in a vessel and processed without material addition or removal. This operation permits more flexibility than continuous operation by allowing adjustment of the operating conditions and the final time. Additional flexibility is available in semi-batch operations, where products are continuously added by adjusting the feed rate profile. We use the term batch process to include semi-batch processes.

Batch processes dealing with reaction and separation operations include reaction, distillation, absorption, extraction, adsorption, chromatography, crystallization, drying, filtration, and centrifugation. The operation of batch processes involves recipes developed in the laboratory. A sequence of operations is performed in a prespecified order in specialized process

equipment, yielding a fixed amount of product. The sequence of tasks to be carried out on each piece of equipment, such as heating, cooling, reaction, distillation, crystallization, and drying, is predefined. The desired production volume is then achieved by repeating the processing steps on a predetermined schedule.

The main characteristics of batch process operations include the absence of steady state, the presence of constraints, and the repetitive nature. These characteristics bring both challenges and opportunities to the operation of batch processes (Bonvin 1998). The challenges are related to the fact that the available models are often poor and incomplete, especially since they need to represent a wider range of operating conditions than in the case of continuous processes. Furthermore, although product quality must be controlled, this variable is usually not available online but only at run end. On the other hand, opportunities stem from the fact that industrial chemical processes are often slow, which facilitates larger sampling periods and extensive online computations. In addition, the repetitive nature of batch processes opens the way to run-to-run process improvement (Bonvin et al. 2006). More information on batch processes and their operation can be found in Seborg et al. (2004) and Nagy and Braatz (2003). Next, we will successively address the control and the optimization of batch processes.

Control of Batch Processes

Control of batch processes differs from control of continuous processes in two main ways. First, since batch processes have no steady-state operating point, at least some of the set points are time-varying profiles. Second, batch processes are repeated over time and are characterized by two independent variables, the run time t and the run counter k . The independent variable k provides additional degrees of freedom for meeting the control objectives when these objectives do not necessarily have to be completed in a single batch but can be distributed over several successive batches. This situation brings into

Implementation aspect	Control objectives	
	Run-time references $y_{\text{ref}}(t)$ or $y_{\text{ref}}[0, t_f]$	Run-end references z_{ref}
Online (within-run)	1 Feedback control $u_k(t) \rightarrow y_k(t) \rightarrow y_k[0, t_f]$ $\uparrow$ FBC	2 Predictive control $u_k(t) \rightarrow z_{\text{pred},k}(t)$ $\uparrow$ MPC
Iterative (run-to-run)	3 Iterative learning control $u_k[0, t_f] \rightarrow y_k[0, t_f]$ $\uparrow$ ILC with run delay	4 Run-to-run control $\mathcal{U}(\pi_k) = u_k[0, t_f] \rightarrow z_k$ $\uparrow$ R2R with run delay

Control and Optimization of Batch Processes, Fig. 1

Control strategies for batch processes. The strategies are classified according to the control objectives (horizontal division) and the implementation aspect (vertical division). Each objective can be met either online or iteratively over several batches depending on the type of measurements available. u_k represents the input vector for the

k th batch, $u_k[0, t_f]$ the corresponding input trajectories, $y_k(t)$ the run-time outputs measured online, and z_k the run-end outputs available at the final time. FBC stands for “feedback control,” MPC for “model predictive control,” ILC for “iterative learning control,” and R2R for “run-to-run control”

focus the concept of run-end outputs, which need to be controlled but are only available at the completion of the batch. The most common run-end output is product quality. Consequently, the control of batch processes encompasses four different strategies (Fig. 1):

1. *Online control of run-time outputs.* This control approach is similar to that used in continuous processing. However, although some controlled variables, such as temperature in isothermal operation, remain constant, the key process characteristics, such as process gain and time constants, can vary considerably because operation occurs along state trajectories rather than at a steady-state operating point. Hence, adaptation in run time t is needed to handle the expected variations. Feedback control is implemented using PID techniques or more sophisticated alternatives (Seborg et al. 2004).
2. *Online control of run-end outputs.* In this case it is necessary to predict the run-end outputs z based on measurements of the run-time outputs y . Model predictive control (MPC) is well suited to this task (Nagy and Braatz 2003). However, the process models available

for prediction are often simplified and thus of limited accuracy.

3. *Iterative control of run-time outputs.* The manipulated variable profiles can be generated using iterative learning control (ILC), which exploits information from previous runs (Moore 1993). This strategy exhibits the limitations of open-loop control with respect to the current run, in particular the fact that there is no feedback correction for run-time disturbances. Nevertheless, this scheme is useful for generating a time-varying feedforward input term.
4. *Iterative control of run-end outputs.* In this case the input profiles are parameterized as $u_k[0, t_f] = \mathcal{U}(\pi_k)$ using the input parameters π_k . The batch process is thus seen as a static map between the input parameters π_k and the run-end outputs z_k (Francois et al. 2005).

It is also possible to combine online and run-to-run control for both y and z . However, in such a combined scheme, care must be taken so that the online and run-to-run corrective actions do not oppose each other. Stability during run time and convergence in run index must be guaranteed (Srinivasan and Bonvin 2007a).

Optimization of Batch Processes

The process variables undergo significant changes during batch operation. Hence, the major objective in batch operations is not to keep the system at optimal constant set points but rather to determine input profiles that optimize an objective function expressing the system performance.

Problem Formulation

A typical optimization problem in the context of batch processes is

$$\min_{u_k[0,t_f]} J_k = \phi(x_k(t_f)) + \int_0^{t_f} L(x_k(t), u_k(t), t) dt \quad (1)$$

subject to

$$\dot{x}_k(t) = F(x_k(t), u_k(t)), \quad x_k(0) = x_{k,0} \quad (2)$$

$$S(x_k(t), u_k(t)) \leq 0, \quad T(x_k(t_f)) \leq 0, \quad (3)$$

where x represents the state vector, J the scalar cost to be minimized, S the run-time constraints, T the run-end constraints, and t_f the final time.

In constrained optimal control problems, the solution often lies on the boundary of the feasible region. Batch processes involve run-time constraints on inputs and states as well as run-end constraints.

Optimization Strategies

As can be seen from the cost objective (1), optimization requires information about the complete run and thus cannot be implemented in real time using only online measurements. Some information regarding the future of the run is needed in the form of either a process model capable of prediction or measurements from previous runs. Accordingly, measurement-based optimization methods can be classified depending on whether or not a process model is used explicitly for implementation, as illustrated in Fig. 2 and discussed next:

1. *Online explicit optimization.* This approach is similar to model predictive control (Nagy and Braatz 2003). Optimization uses a process model explicitly and is repeated whenever a new set of measurements becomes available. This scheme involves two steps, namely, updating the initial conditions for the subsequent optimization (and optionally the param-

Implementation aspect	Use of process model	
	Explicit optimization (with process model)	Implicit optimization (without process model)
Online (within-run)	1 Repeated online optimization $y_k[0,t] \xrightarrow{\text{EST}} \hat{x}_k(t) \xrightarrow{\text{OPT}} u_k^*[t,t_f]$ 	2 Online input update using measurements $y_k(t) \xrightarrow{\text{Approx. of opt. solution}} u_k^*(t)$ $y_k[0,t] \xrightarrow{\text{NCO prediction}} \text{NCO} \rightarrow u_k^*(t)$
Iterative (run-to-run)	3 Repeated run-to-run optimization $y_k[0,t_f] \xrightarrow{\text{IDENT}} \hat{\theta}_k \xrightarrow{\text{OPT}} u_{k+1}^*[0,t_f]$ 	4 Run-to-run input update using measurements $y_k[0,t_f] \xrightarrow{\text{NCO evaluation}} \text{NCO} \rightarrow u_{k+1}^*[0,t_f]$

Control and Optimization of Batch Processes, Fig. 2 Optimization strategies for batch processes. The strategies are classified according to whether or not a process model is used for implementation (horizontal division). Furthermore, each class can be implemented either online

or iteratively over several runs (vertical division). EST stands for “estimation,” IDENT for “identification,” OPT for “optimization,” and NCO for “necessary conditions of optimality”

eters of the process model) and numerical optimization based on the updated process model (Abel et al. 2000). Since both steps are repeated as measurements become available, the procedure is also referred to as repeated online optimization. The weakness of this method is its reliance on the model; if the model is not updated, its accuracy plays a crucial role. However, when the model is updated, there is a conflict between parameter identification and optimization since parameter identification requires persistency of excitation, that is, the inputs must be sufficiently varied to uncover the unknown parameters, a condition that is usually not satisfied when near-optimal inputs are applied. Note that, instead of computing the input $u_k^*[t, t_f]$, it is also possible to use a receding horizon and compute only $u_k^*[t, t + T]$, with T the control horizon (Abel et al. 2000).

2. *Online implicit optimization.* In this scenario, measurements are used to update the inputs directly, that is, without the intermediary of a process model. Two classes of techniques can be identified. In the first class, an update law that approximates the optimal solution is sought. For example, a neural network is trained with data corresponding to optimal behavior for various uncertainty realizations and used to update the inputs (Rahman and Palanki 1996). The second class of techniques relies on transforming the optimization problem into a control problem that enforces the necessary conditions of optimality (NCO) (Srinivasan and Bonvin 2007b). The NCO involve constraints that need to be made active and sensitivities that need to be pushed to zero. Since some of these NCO are evaluated at run time and others at run end, the control problem involves both run-time and run-end outputs. The main issue is the measurement or estimation of the controlled variables, that is, the constraints and sensitivities that constitute the NCO.
3. *Iterative explicit optimization.* The steps followed in run-to-run explicit optimization are the same as in online explicit optimization. However, there is substantially more data

available at the end of the run as well as sufficient computational time to refine the model by updating its parameters and, if needed, its structure. Furthermore, data from previous runs can be collected for model update (Rastogi et al. 1992). As with online explicit optimization, this approach suffers from the conflict between estimation and optimization.

4. *Iterative implicit optimization.* In this scenario, the optimization problem is transformed into a control problem, for which the control approaches in the second row of Fig. 1 are used to meet the run-time and run-end objectives (Francois et al. 2005). The approach, which is conceptually simple, might be experimentally expensive since it relies more on data.

These complementary measurement-based optimization strategies can be combined by implementing some aspects of the optimization online and others on a run-to-run basis. For instance, in explicit schemes, the states can be estimated online, while the model parameters can be estimated on a run-to-run basis. Similarly, in implicit optimization, approximate update laws can be implemented online, leaving the responsibility for satisfying terminal constraints and sensitivities to run-to-run controllers.

Summary and Future Directions

Batch processing presents several challenges. Since there is little time for developing appropriate dynamic models, there is a need for improved data-driven control and optimization approaches. These approaches require the availability of online concentration-specific measurements such as chromatographic and spectroscopic sensors, which are not yet readily available in production.

Technically, the main operational difficulty in batch process improvement lies in the presence of run-end outputs such as final quality, which cannot be measured during the run. Although model-based solutions are available, process models in the batch area tend to be poor. On

the other hand, measurement-based optimization for a given batch faces the challenge of having to know about the future to act during the batch. Consequently, the main research push is in the area of measurement-based optimization and the use of data from both the current and previous batches for control and optimization purposes.

Cross-References

- ▶ [Industrial MPC of continuous processes](#)
- ▶ [Iterative Learning Control](#)
- ▶ [Multiscale Multivariate Statistical Process Control](#)
- ▶ [Scheduling of Batch Plants](#)
- ▶ [State Estimation for Batch Processes](#)

Bibliography

- Abel O, Helbig A, Marquardt W, Zwick H, Daszkowski T (2000) Productivity optimization of an industrial semi-batch polymerization reactor under safety constraints. *J Process Control* 10(4):351–362
- Bonvin D (1998) Optimal operation of batch reactors – a personal view. *J Process Control* 8(5–6):355–368
- Bonvin D, Srinivasan B, Hunkeler D (2006) Control and optimization of batch processes: improvement of process operation in the production of specialty chemicals. *IEEE Control Syst Mag* 26(6):34–45
- Francois G, Srinivasan B, Bonvin D (2005) Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. *J Process Control* 15(6):701–712
- Moore KL (1993) Iterative learning control for deterministic systems. *Advances in industrial control*. Springer, London
- Nagy ZK, Braatz RD (2003) Robust nonlinear model predictive control of batch processes. *AIChE J* 49(7):1776–1786
- Rahman S, Palanki S (1996) State feedback synthesis for on-line optimization in the presence of measurable disturbances. *AIChE J* 42:2869–2882
- Rastogi A, Fotopoulos J, Georgakis C, Stenger HG (1992) The identification of kinetic expressions and the evolutionary optimization of specialty chemical batch reactors using tendency models. *Chem Eng Sci* 47(9–11):2487–2492
- Seborg DE, Edgar TF, Mellichamp DA (2004) *Process dynamics and control*. Wiley, New York
- Srinivasan B, Bonvin D (2007a) Controllability and stability of repetitive batch processes. *J Process Control* 17(3):285–295

- Srinivasan B, Bonvin D (2007b) Real-time optimization of batch processes by tracking the necessary conditions of optimality. *Ind Eng Chem Res* 46(2):492–504

Control Applications in Audio Reproduction

Yutaka Yamamoto
Graduate School of Informatics, Kyoto
University, Kyoto, Japan

Abstract

This entry gives a brief overview of the recent developments in audio sound reproduction via modern sampled-data control theory. We first review basics in the current sound processing technology and then proceed to the new idea derived from sampled-data control theory, which is different from the conventional Shannon paradigm based on the perfect band-limiting hypothesis. The hybrid nature of sampled-data systems provides an optimal platform for dealing with signal processing where the ultimate objective is to reconstruct the original analog signal one started with. After discussing some fundamental problems in the Shannon paradigm, we give our basic problem formulation that can be solved using modern sampled-data control theory. Examples are given to illustrate the results.

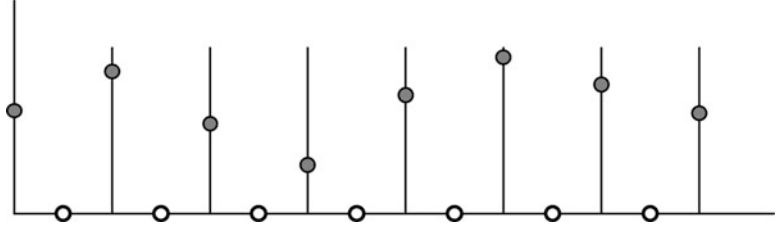
Keywords

Digital signal processing · Multirate signal processing · Sampled-data control · Sampling theorem · Sound reconstruction

Introduction: Status Quo

Consider the problem of reproducing sounds from recorded media such as compact discs. The current CD format is recorded at the sampling frequency 44.1 kHz. It is commonly claimed that the highest frequency for human audibility is

Control Applications in Audio Reproduction,
Fig. 1 Upsampler for $M = 2$



20 kHz, whereas the upper bound of reproduction in this format is believed to be the half of 44.1 kHz, i.e., 22.05 kHz, and hence, this format should have about 10 % margin against the alleged audible limit of 20 kHz.

CD players of early days used to process such digital signals with the simple zero-order hold at this frequency, followed by an analog low-pass filter. This process requires a sharp low-pass characteristic to cut out unnecessary high frequency beyond 20 kHz. However, a sharp cut-off low-pass characteristic inevitably requires a high-order filter which in turn introduces a large amount of phase shift distortion around the cutoff frequency.

To circumvent this defect, the idea of an oversampling DA converter that is realized by the combination of a digital filter and a low-order analog filter was introduced (Zelniker and Taylor 1994). This is based on the following principle:

Let $\{f(nh)\}_{n=-\infty}^{\infty}$ be a discrete-time signal obtained from a continuous-time signal $f(\cdot)$ by sampling it with sampling period h . The *upsampler* appends the value 0, $M - 1$ times, between two adjacent sampling points:

$$(\uparrow Mw)[k] := \begin{cases} w(\ell), & k = M\ell \\ 0, & \text{elsewhere.} \end{cases} \quad (1)$$

See Fig. 1 for the case $M = 2$. This has the effect of making the unit operational time M times faster.

The bandwidth will also be expanded by M times and the *Nyquist frequency* (i.e., half the sampling frequency) becomes $M\pi/h$ [rad/sec]. As we see in the next section, the Nyquist frequency is often regarded as the true bandwidth of the discrete-time signal $\{f(nh)\}_{n=-\infty}^{\infty}$. But this upsampling process just inserts zeros between

sampling points, and the real information contents (the true bandwidth) is not really expanded. As a result, the copy of the frequency content for $[0, \pi/h)$ appears as a mirror image repeatedly over the frequency range above π/h . This distortion is called *imaging*. In order to avoid the effect of such mirrored frequency components, one often truncates the frequency components beyond the (original) Nyquist frequency via a digital low-pass filter that has a sharp roll-off characteristic. One can then complete the digital to analog (DA) conversion process by postposing a slowly decaying analog filter. This is the idea of an *oversampling DA converter* (Zelniker and Taylor 1994). The advantage here is that by allowing a much wider frequency range, the final analog filter can be a low-order filter and hence yields a relatively small amount of phase distortion supported in part by the linear-phase characteristic endowed on the digital filter preceding it.

Signal Reconstruction Problem

As before, consider the sampled discrete-time signal $\{f(nh)\}_{n=-\infty}^{\infty}$ obtained from a continuous-time signal f . The main question is how we can recover the original continuous-time signal $f(\cdot)$ from sampled data. This is clearly an ill-posed problem without any assumption on f because there are infinitely many functions that can match the sampled data $\{f(nh)\}_{n=-\infty}^{\infty}$. Hence, one has to impose a reasonable a priori assumption on f to sensibly discuss this problem.

The following sampling theorem gives one answer to this question:

Theorem 1 Suppose that the signal $f \in L^2$ is perfectly band-limited, in the sense that there exists $\omega_0 \leq \pi/h$ such that the Fourier transform

$\hat{f}$ of f satisfies

$$\hat{f}(\omega) = 0, \quad |\omega| \geq \omega_0, \quad (2)$$

Then

$$f(t) = \sum_{n=-\infty}^{\infty} f(nh) \frac{\sin \pi(t/h - n)}{\pi(t/h - n)}. \quad (3)$$

This theorem states that if the signal f does not contain any high-frequency components beyond the *Nyquist frequency* π/h , then the original signal f can be uniquely reconstructed from its sampled-data $\{f(nh)\}_{n=-\infty}^{\infty}$. On the other hand, if this assumption does not hold, then the result does not necessarily hold. This is easy to see via a schematic representation in Fig. 2.

If we sample the sinusoid in the upper figure in Fig. 2, these sampled values would turn out to be compatible with another sinusoid with much lower frequency as the lower figure shows. In other words, this sampling period does not have enough resolution to distinguish these two sinusoids. The maximum frequency below where there does not occur such a phenomenon is the Nyquist frequency. The sampling theorem above asserts that it is half of the sampling frequency $2\pi/h$, that is, π/h [rad/sec]. In other words, if we can assume that the original signal contains no frequency components beyond the Nyquist frequency, then one can uniquely reconstruct the original analog signal f from its sampled-

data $\{f(nh)\}_{n=-\infty}^{\infty}$. On the other hand, if this assumption does not hold, the distortion depicted in Fig. 2 occurs; this is called *aliasing*.

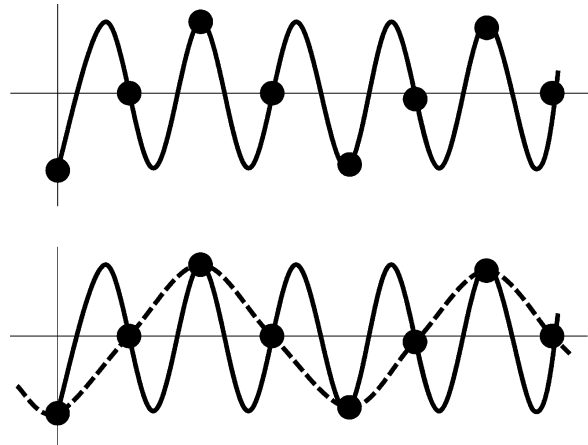
This is the content of the sampling theorem. It has been widely accepted as the basis for digital signal processing that bridges analog to digital. Concrete applications such as CD, MP3, and digital images are based on this principle in one way or another.

Difficulties

However, this paradigm (hereafter the *Shannon paradigm*) of the perfect band-limiting hypothesis and the resulting sampling theorem renders several difficulties as follows:

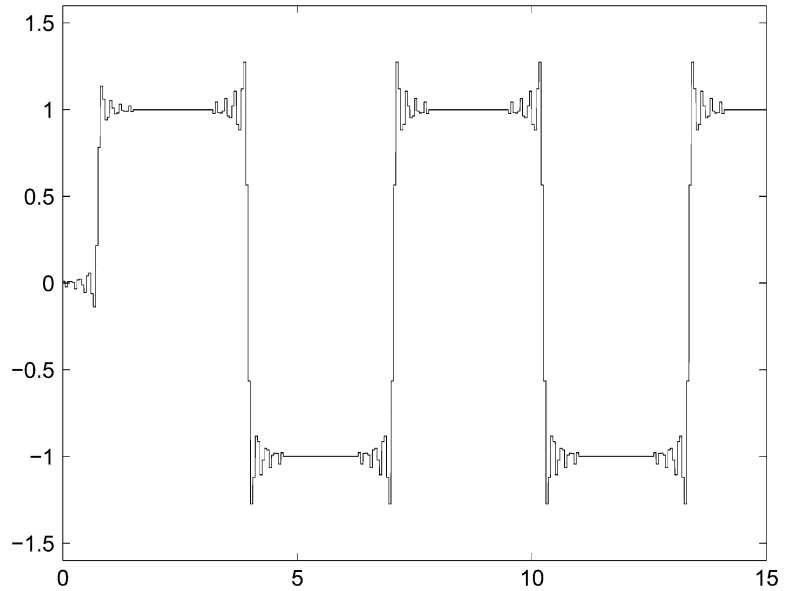
- The reconstruction formula (3) is not causal, i.e., one needs future sampled values to reconstruct the present value $f(t)$. One can remedy this defect by allowing a certain amount of delay in reconstruction, but this delay can depend on how fast the formula converges.
- This formula is known to decay slowly; that is, we need many terms to approximate if we use this formula as it is.
- The perfect band-limiting hypothesis is hardly satisfied in reality. For example, for CDs, the Nyquist frequency is 22.05 kHz, and the energy distribution of real sounds often extends way over 20 kHz.

Control Applications in Audio Reproduction,
Fig. 2 Aliasing



Control Applications in Audio Reproduction,

Fig. 3 Ringing due to the Gibbs phenomenon



- To remedy this, one often introduces a band-limiting low-pass filter, but it can introduce distortions due to the Gibbs phenomenon well-known in Fourier analysis. A sharp truncation in the frequency domain yields such a ringing effect. See Fig. 3.

In view of such drawbacks, there has been revived interest in the extension of the sampling theorem in various forms since the 1990s. There is by now a stream of papers that aim at studying signal reconstruction under the assumption of nonideal signal acquisition devices; an excellent survey is given in Unser (2000). In this research framework, the incoming signal is supposed to be acquired through a nonideal analog filter (acquisition device) and sampled, and then the reconstruction process attempts to recover the original signal. The idea is to place the problem into the framework of the (orthogonal or oblique) projection theorem in a Hilbert space (usually L^2) and then project the signal space to the subspace generated by the shifted reconstruction functions. It is often required that the process gives a *consistent* result, i.e., if we subject the reconstructed signal to the whole process again, it should yield the same sampled values from which it was reconstructed (Unser and Aldroubi 1994).

In what follows, we take a similar viewpoint, that is, the incoming signals are acquired through a nonideal filter but develop a methodology different from the projection method, relying on sampled-data control theory.

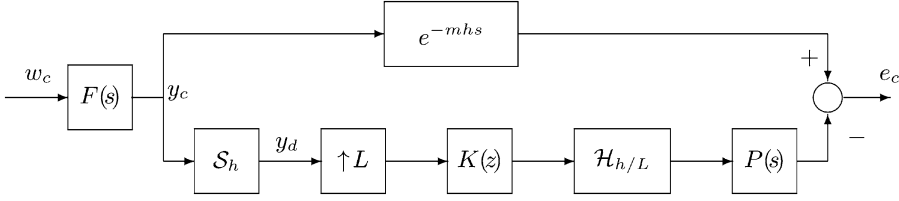
The Signal Class

We have seen that the perfect band-limiting hypothesis is restrictive. Even if we adopt it, it is a fairly crude model for analog signals to allow for a more elaborate study.

Let us now pose the question: *What class of functions should we process in such systems?*

Consider the situation where one plays a musical instrument, say, a guitar. A guitar naturally has a frequency characteristic. When one picks a string, it produces a certain tone along with its harmonics, as well as a characteristic transient response. All these are governed by a certain frequency decay curve, demanded by the physical characteristics of the guitar. Let us suppose that such a frequency decay is governed by a rational transfer function $F(s)$ and it is driven by varied exogenous inputs.

Consider Fig. 4. The exogenous analog signal $w_c \in L^2$ is applied to the analog filter $F(s)$.



Control Applications in Audio Reproduction, Fig. 4 Error system for sampled-data design

This $F(s)$ is not an ideal filter, and hence its bandwidth is not restricted to below the Nyquist frequency. The signal w_c drives $F(s)$ to produce the target analog signal y_c , which should be the signal to be reconstructed. It is then sampled by sampler S_h and becomes the recorded or transmitted digital signal y_d . The objective here is to reconstruct the target analog signal y_c out of this sampled signal y_d . In order to recover the frequency components beyond the Nyquist frequency, one needs a faster sampling period, so we insert the upsampler $\uparrow L$ to make the sampling period h/L . This upsampled signal is processed by digital filter $K(z)$ and then becomes a continuous-time signal again by going through the hold device $\mathcal{H}_{h/L}$. It will then be processed by analog filter $P(s)$ to be smoothed out. The obtained signal is then compared with delayed analog signal $y_c(t - mh)$ to form the delayed error signal e_c . The objective is then to make this error e_c as small as possible. The reason for allowing delay e^{-mhs} is to accommodate certain processing delays. This is the idea of the block diagram Fig. 4.

The performance index we minimize is the induced norm of the transfer operator T_{ew} from w_c to e_c :

$$\|T_{ew}\|_\infty := \sup_{w_c \neq 0} \frac{\|e_c\|_2}{\|w_c\|_2}. \quad (4)$$

This is the H^∞ norm of the sampled-data control system Fig. 4. Our objective is then to solve the following problem:

Filter Design Problem

Suppose we are given the system specified by Fig. 4. For a given performance level $\gamma > 0$, find a filter $K(z)$ such that

$$\|T_{ew}\|_\infty < \gamma.$$

This is a sampled-data H^∞ (sub-)optimal control problem. It can be solved by using the standard solution method for sampled-data control systems (Chen and Francis 1995a; Yamamoto 1999; Yamamoto and Nagahara 2017; Yamamoto et al. 2012). The only anomaly here is that the system in Fig. 4 contains a delay element e^{-mhs} which is infinite dimensional. However, by suitably approximating this delay by successive series of shift registers, one can convert the problem to an appropriate finite-dimensional discrete-time problem (Yamamoto et al. 1999, 2002, 2012).

This problem setting has the following features:

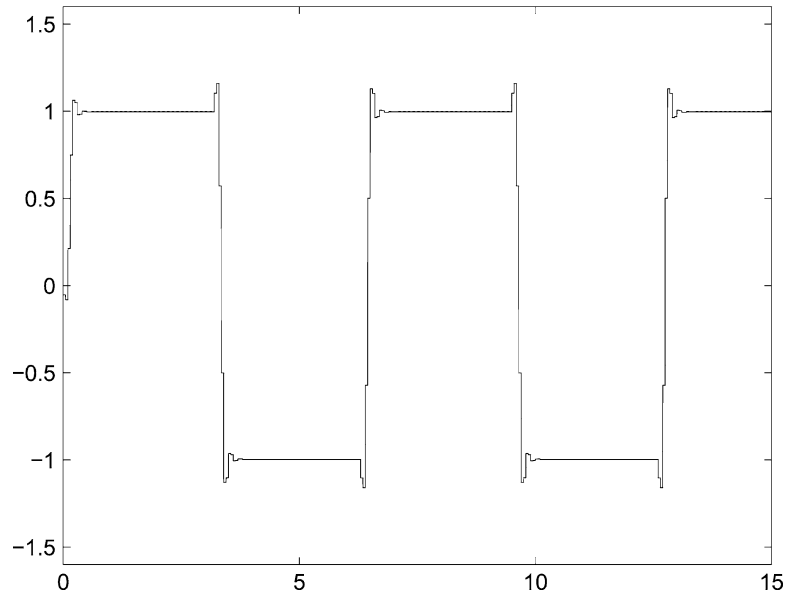
1. One can optimize the continuous-time performance under the constraint of discrete-time filters.
2. By setting the class of input functions as L^2 functions band-limited by $F(s)$, one can capture the continuous-time error signal e_c and its worst-case norm in the sense of (4).

The first feature is due to sampled-data control theory. A great advantage of sampled-data control theory is that it allows the mixture of continuous- and discrete-time components. This is in marked contrast to the Shannon paradigm where continuous-time performance is really demanded by the artificial perfect band-limiting hypothesis.

The second feature is an advantage due to H^∞ control theory. Naturally, we cannot have an access to each *individual* error signal e_c , but we can still control the *overall performance* from w_c to e_c in terms of the H^∞ norm that guarantees the

Control Applications in Audio Reproduction,

Fig. 5 Response of the proposed sampled-data filter against a rectangular wave



worst-case performance. This is in clear contrast with the classical case where only a representative response, e.g., impulse response in the case of H^2 , is targeted. Furthermore, since we can control the continuous-time performance of the worst-case error signal, the present method can indeed minimize (continuous-time) phase errors. This is an advantage usually not possible with conventional methods since they mainly discuss the gain characteristics of the designed filters only. By the very property of minimizing the H^∞ norm of the *continuous-time error signal* e_c , the present method can even control the phase errors and yield much less phase distortion even around the cutoff frequency.

Figure 5 shows the response of the proposed sampled-data filter against a rectangular wave, with a suitable first- or second-order analog filter $F(s)$; see Yamamoto et al. (2012) for more details. Unlike Fig. 3, the overshoot is controlled to be minimum.

The present method has been patented (Yamamoto 2006; Yamamoto and Nagahara 2006; Fujiyama et al. 2008) and implemented into sound processing LSI chips as a core technology by Sanyo Semiconductor and successfully used in mobile phones, digital voice

recorders, and MP3 players; their cumulative production has exceeded 70 million units as of the end of 2017.

Summary and Future Directions

We have presented basic ideas of new signal processing theory derived from sampled-data control theory. The theory presents advantages over conventional projection methods, whether based on the perfect band-limiting hypothesis or not.

The application of sampled-data control theory to digital signal processing was first made by Chen and Francis (1995b) with performance measure in the discrete-time domain; see also Hassibi et al. (2006). The present author and his group have pursued the idea presented in this article since 1996 (Khargonekar and Yamamoto 1996). See Yamamoto et al. (2012) and references therein. For the background of sampled-data control theory, consult, e.g., Chen and Francis (1995a), Yamamoto (1999), and Yamamoto and Nagahara (2017).

The same philosophy of emphasizing the importance of analog performance was proposed and pursued recently by Unser and co-workers

(1994), Unser (2005), and Eldar and Dvorkind (2006). The crucial difference is that they rely on L^2/H^2 type optimization and orthogonal or oblique projections, which are very different from our method here. In particular, such projection methods can behave poorly for signals outside the projected space. The response shown in Fig. 3 is a typical such example.

Applications to image processing are discussed in Yamamoto et al. (2012). An application to delta-sigma DA converters is studied in Nagahara and Yamamoto (2012a); see also Nagahara and Yamamoto (2012b) for a design with non-integer delays that is effective in pitch shifting. The reader is also referred to Yamamoto et al. (2017, 2018) for filter designs involving generalized sampling or with compactly supported acquisition devices. Again, the crux of the idea is to assume a signal generator model and then design an optimal filter in the sense of Fig. 4 or a similar diagram with the same idea. This idea should be applicable to a much wider class of problems in signal processing and should prove to have more impact.

Some processed examples of still and moving images are downloadable from the site: <https://sites.google.com/site/yutakayamamotoen/>

For the sampling theorem, see Shannon (1949), Unser (2000), and Zayed (1996), for example. Note, however, that Shannon himself (1949) did not claim originality on this theorem; hence, it is misleading to attribute this theorem solely to Shannon. See Unser (2000) and Zayed (1996) for some historical accounts. For a general background in signal processing, Vetterli et al. (2013) is useful.

Cross-References

- [H-Infinity Control](#)
- [Optimal Sampled-Data Control](#)
- [Sampled-Data Systems](#)

Acknowledgments The author would like to thank Masaaki Nagahara and Masashi Wakaiki for their help with the numerical examples. Part of this entry is based on the exposition (Yamamoto 2007) written in Japanese.

Bibliography

- Chen T, Francis BA (1995a) Optimal sampled-data control systems. Springer, New York
- Chen T, Francis BA (1995b) Design of multirate filter banks by H_∞ optimization. *IEEE Trans Signal Process* 43:2822–2830
- Eldar YC, Dvorkind TG (2006) A minimum squared-error framework for generalized sampling. *IEEE Trans Signal Process* 54(6):2155–2167
- Fujiyama K, Iwasaki N, Hirasawa Y, Yamamoto Y (2008) High frequency compensator and reproducing device. US patent 7,324,024 B2
- Hassibi B, Erdogan AT, Kailath T (2006) MIMO linear equalization with an H^∞ criterion. *IEEE Trans Signal Process* 54(2):499–511
- Khargonekar PP, Yamamoto Y (1996) Delayed signal reconstruction using sampled-data control. In: *Proceedings of 35th IEEE CDC, Kobe*, pp 1259–1263
- Nagahara M, Yamamoto Y (2012a) Frequency domain min-max optimization of noise-shaping delta-sigma modulators. *IEEE Trans Signal Process* 60(6):2828–2839
- Nagahara M, Yamamoto Y (2012b) H^∞ -optimal fractional delay filters. *IEEE Trans Signal Process* 61(18):4473–4480
- Shannon CE (1949) Communication in the presence of noise. *Proc IRE* 37(1):10–21
- Unser M (2000) Sampling – 50 years after Shannon. *Proc IEEE* 88(4):569–587
- Unser M (2005) Cardinal exponential splines: part II – think analog, act digital. *IEEE Trans Signal Process* 53(4):1439–1449
- Unser M, Aldroubi A (1994) A general sampling theory for nonideal acquisition devices. *IEEE Trans Signal Process* 42(11):2915–2925
- Vetterli M, Kováčević J, Goyal V (2013) *Foundations of signal processing*. Cambridge University Press, Cambridge
- Yamamoto Y (1999) Digital control. In: Webster JG (ed) *Wiley encyclopedia of electrical and electronics engineering*, vol 5. Wiley, New York, pp 445–457
- Yamamoto Y (2006) Digital/analog converters and a design method for the pertinent filters. Japanese patent 3,820,331
- Yamamoto Y (2007) New developments in signal processing via sampled-data control theory–continuous-time performance and optimal design. *Meas Control (Jpn)* 46:199–205
- Yamamoto Y, Anderson BDO, Nagahara M (2002) Approximating sampled-data systems with applications to digital redesign. In: *Proceedings of the 41st IEEE CDC, Las Vegas*, pp 3724–3729
- Yamamoto Y, Nagahara M (2006) Sample-rate converters. Japanese patent 3,851,757
- Yamamoto Y, Nagahara M (2017) Digital control In: Webster JG (ed) *Wiley encyclopedia of electrical and electronics engineering online*. <https://doi.org/10.1002/047134608X.W1010.pub2>

- Yamamoto Y, Madievski AG, Anderson BDO (1999) Approximation of frequency response for sampled-data control systems. *Automatica* 35(4):729–734
- Yamamoto Y, Nagahara M, Khargonekar PP (2012) Signal reconstruction via H^∞ sampled-data control theory—beyond the Shannon paradigm. *IEEE Trans Signal Process* 60:613–625
- Yamamoto K, Nagahara M, Yamamoto Y (2017) Signal reconstruction with generalized sampling. In: *Proceedings of the 456th IEEE CDC*, Melbourne, pp 6253–6258
- Yamamoto Y, Yamamoto K, Nagahara M (2018) Sampled-data filters with compactly supported acquisition pre-filters. In: *Proceedings of the 57th IEEE CDC*, Miami, pp 6650–6655
- Zayed AI (1996) *Advances in Shannon's sampling theory*. CRC Press, Boca Raton
- Zelniker G, Taylor FJ (1994) *Advanced digital signal processing: theory and applications*. Marcel Dekker, New York

Control for Precision Mechatronics

Tom Oomen
Department of Mechanical Engineering,
Eindhoven University of Technology,
Eindhoven, The Netherlands

Abstract

Motion systems are a key enabling technology in manufacturing machines and scientific instruments. Indeed, these motion systems perform the positioning with high speed and accuracy, where typical requirements are in the micrometer or even nanometer range. This is achieved through a mechatronic system design, which includes actuators, sensors, mechanics, and control. The aim of this entry is to outline advanced motion control for precision mechatronics. Both feedback and feedforward control are covered. Their specific designs are heavily influenced by considerations regarding efficient and accurate modeling techniques. Extensions to complex multivariable systems are outlined, as well as challenges induced by envisaged future application requirements.

Keywords

Mechatronics · Feedback control · Feedforward control · Learning control · System identification

Introduction

Manufacturing machines and scientific instruments have a key role in our society, and the role of these mechatronic systems will increase in the future. For instance, future developments in manufacturing machines as used in lithographic integrated circuit (IC) production will enable ubiquitous computing power for emerging fields in communication, medical equipment, transportation, etc. In fact, Moore's law, dictating computational progress, is determined by wafer scanner technology. Ongoing developments in printing systems are foreseen to lead to additive manufacturing techniques that enable the design of new products with a functionality that is unimaginable in classical production techniques. Highly accurate scientific instruments, including microscopy, will facilitate scientific discoveries in nanotechnology and biotechnology. Increasingly large telescopes will spur deep space exploration. Finally, medical equipment, for instance, computerized tomography (CT) scanners, will enable improved medical treatments. To keep up with performance requirements imposed by societal demands, future machines must achieve unprecedented performance.

Motion systems are key subsystems essential for meeting future requirements. These mechatronic subsystems are responsible for the motion that positions the product in the machine, e.g., the wafer in lithography, the substrate in printing, the sample in microscopy, the mirror alignment in telescopes and lithographic optics, and the detector in CT scanners. Future machines, e.g., in lithography, must achieve an extreme accuracy of 0.1 nm over a range of 1 m, exceeding a 1 billion ratio, with speeds up to $1 \frac{\text{m}}{\text{s}}$ and accelerations of $100 \frac{\text{m}}{\text{s}^2}$.

Control is an essential aspect in such motion systems (Ohnishi et al 1996), (Lee and Tomizuka 1996), (Steinbuch and Norg 1998), Devasia et al (2007). Indeed, mechatronic system design involves many aspects, including actuators, mechanics, and sensors (Munnig Schmidt et al 2011). Despite a very large variety in designs and applications, motion control is remarkably similar from system to system.

Motion Systems

Mechatronic systems typically consist of actuators, mechanics, and sensors. These actuators generate a force or a torque for example, while the sensors can measure position, rotation, etc. The dynamics of such systems are typically dominated by the mechanics, with sensing and actuation assumed to be perfect because their dynamics are much faster and more precise than the mechanics. As a result Gawronski (2004),

$$y(s) = \left(\underbrace{\sum_{i=1}^{n_{RB}} \frac{c_i b_i^T}{s^2}}_{\text{rigid-body modes}} + \underbrace{\sum_{i=n_{RB}+1}^{n_s} \frac{c_i b_i^T}{s^2 + 2\zeta_i \omega_i s + \omega_i^2}}_{\text{flexible modes}} \right) u(s), \quad (1)$$

where y is the output position, u is the actuator input, n_{RB} is the number of rigid-body modes, the vectors $c_i \in \mathbb{R}^{n_y}$ and $b_i \in \mathbb{R}^{n_u}$ are associated with the mode shapes, and $\zeta_i, \omega_i \in \mathbb{R}_+$. Here, $n_s \in \mathbb{N}$ may be very large and even infinite. Also, s is a complex indeterminate due to the Laplace transform. Note that in (1), it is assumed that the rigid-body modes are not suspended, i.e., the term $\frac{1}{s^2}$ relates to Newton's second law. In the case of suspended rigid-body modes, e.g., in case of flexures as in Fleming and Leang (2014), (1) can directly be extended.

The desktop printer in Fig. 1 is a low-complexity example that satisfies (1). In particular, in Fig. 2, the Bode plot of the transfer function in (1) is shown, together with a measurement of the real system. A single rigid-body mode and a single flexible mode can clearly be observed. Note that both (1) and the Bode plot inherently assume linear dynamics; friction effects that are present in the printer are further investigated in a later section.

Control Goal and Architecture

The main goal in motion control is a servo task: let the output y follow a prespecified reference r . This is typically achieved through the control architecture in Fig. 3. Assuming a single-input single-output system,

$$e = \underbrace{S(1 - GF)}_{\text{feedforward}} r - \underbrace{S}_{\text{feedback}} v - \underbrace{S}_{\text{feedback}} \eta \quad (2)$$

where v are disturbances affecting the system, and η is measurement noise. Also,

$$S = \frac{1}{1 + GK} \quad (3)$$

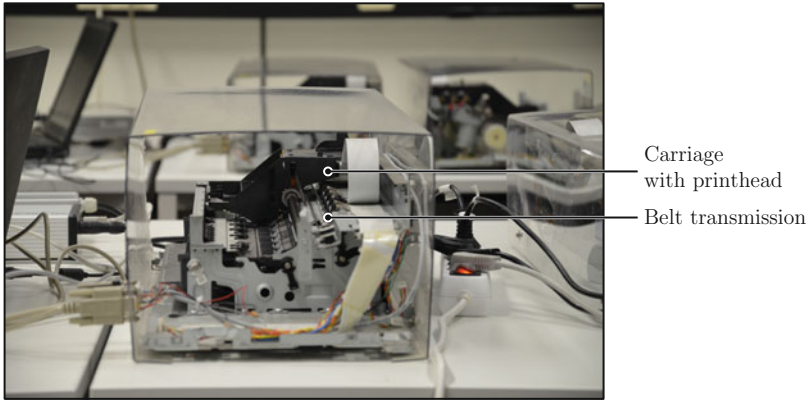
and

$$T = \frac{GK}{1 + GK} \quad (4)$$

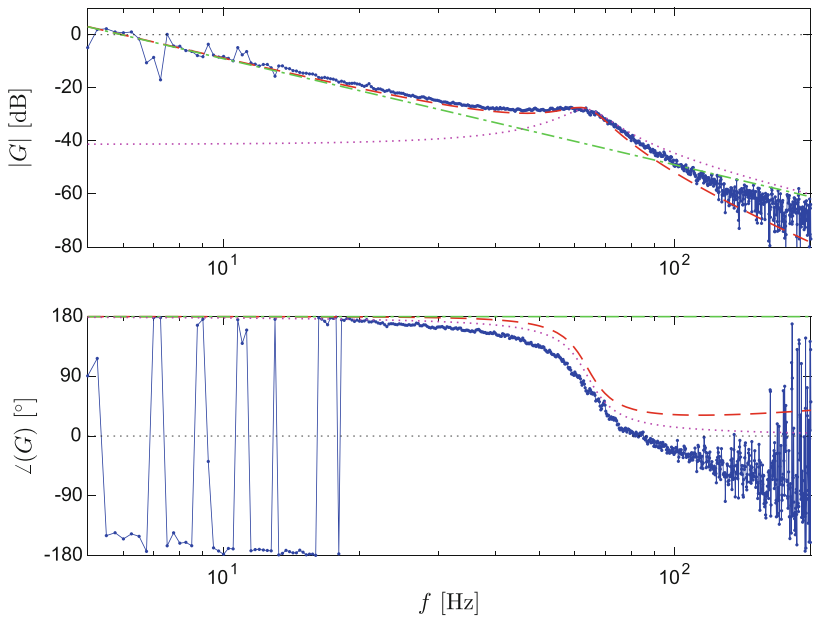
are the sensitivity function and complementary sensitivity function, respectively. Essentially, three design variables have to be selected by the control engineer:

1. The reference signal r
2. The feedback controller K
3. The feedforward controller F

The reference often involves a point-to-point motion, where the goal is to reach an endpoint with high accuracy and minimal time. In particular, the reference is specified such that the actuator constraints are not violated, see Lambrechts et al (2005). In subsequent sections, feedback control and feedforward control are investigated.



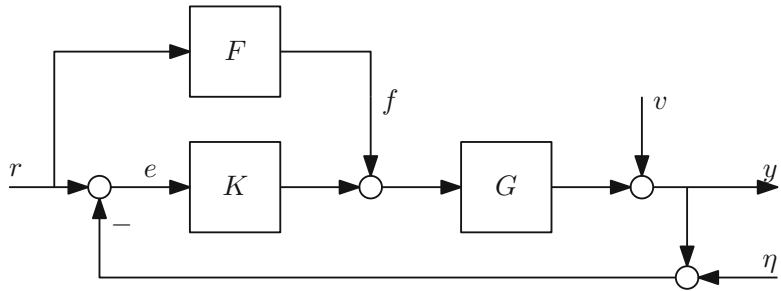
Control for Precision Mechatronics, Fig. 1 Desktop printer system



Control for Precision Mechatronics, Fig. 2 Model of the desktop printer. Overall model of the mechanics (1) (dashed red), rigid-body mode (dash-dotted green), flexible mode (dotted magenta). Furthermore, an identified

frequency response function is depicted, confirming the model characteristics (note the inaccuracies at low and high frequencies due to a poor signal to noise ratio) (solid blue).

Control for Precision Mechatronics, Fig. 3 Motion control architecture.



Feedback Design for Performance

In this section, a motion control design procedure is outlined, leading to the feedback controller K in Fig. 3.

Accurate Models for Motion Feedback Control: Nonparametric System Identification

Any control design approach requires knowledge of the true system G , often in the form of a mathematical model. On the one hand, these models can be obtained through first-principles modeling, which typically leads to a model of the form (1) directly. This approach is often time consuming for mechanical systems and often leads to inaccurate parameters, e.g., the natural frequencies ω_i in (1). On the other hand, system identification or experimental modeling can be used, directly employing measured data. For mechanical systems, system identification is fast, accurate, and inexpensive, due to high signal-to-noise ratios and high sampling rates.

In system identification, roughly two approaches can be distinguished: nonparametric approach and parametric approaches. Nonparametric approaches, more specifically frequency response functions $G(\omega)$, have a key role in motion control for several reasons. An example of such a frequency response function is given in Fig. 2. First, these are used as a basis for loop-shaping design in the next section. Second, these are used as a basis for parametric identification, where these contribute to model order selection, i.e., the number of flexible modes n_s can be estimated from a Bode magnitude plot of $G(\omega)$, and model validation.

Frequency response identification of motion systems consists of the following steps.

1. Implement a stabilizing controller K
2. Apply a suitable excitation signal f and/or r in Fig. 3 and measure closed-loop signals y and u
3. Estimate $G(\omega)$ based on the measured data

An excellent treatment of these steps is given in Pintelon and Schoukens (2012, Chapter 2).

Major recent advancements include the use of periodic excitations instead of noise excitation and approaches to reduce transients, substantially increasing accuracy and reducing measurement time, see Voorhoeve et al (2018) for details on these recent advances and experimental results on motion systems.

Loop-Shaping Design

The goal of feedback in motion control is to minimize the error in (2) through selecting K in view of

1. Attenuating disturbances v through minimizing $|S|$.
2. Compensating for feedforward controller inaccuracies, i.e., the term $(1 - GF)r$ in (2) through minimizing $|S|$.
3. Reducing the effect of measurement noise η through minimizing $|T|$ since the transfer function from η to y is given by $-T$.
4. Guaranteeing closed-loop stability, i.e., stability of the closed-loop transfer functions S , T , etc.

These goals are conflicting due to performance constraints, including the easily verifiable relation $S + T = 1$, in addition to the Bode sensitivity integral that reveals that S cannot be small at all frequencies, e.g., Åström and Murray (2008, Sect. 11.5). For motion systems, Requirement 2 is mainly important at low frequencies, whereas Requirement 3 is mainly important at high frequencies where the measurement noise η is relatively large compared to the reference signal. Minimizing $|T|$ implies $|S| = 1$, which essentially implies the controller does not react on measurement noise in the measured error (2).

The idea in loop-shaping is to select K that directly affects the loop-gain $L = GK$. Indeed, L can be directly manipulated by the choice of K using Bode and Nyquist diagrams, see, e.g., Åström and Murray (2008, Sect. 11.4). In turn, L directly connects to the closed-loop transfer functions in certain frequency ranges. Loop-shaping typically consists of the following steps, which are typically re-tuned in an iterative fashion.

1. Pick a crossover frequency f_{bw} [Hz], which is the frequency where $|L(2\pi f_{bw})| = 1$. Typically, $|S| < 1$ below f_{bw} , while $|T| < 1$ beyond f_{bw} . Furthermore, f_{bw} is typically selected in the region where the rigid-body modes dominate, i.e., $f_{bw} < \frac{\omega_i}{2\pi}$, where $\omega_i, i = n_{rb} + 1, \dots, n_s$ is defined in (1).
2. Implement a lead filter, typically specified as $K_{lead} = p_{lead} \frac{\frac{1}{2\pi \cdot \frac{1}{3} f_{bw}} + 1}{\frac{1}{2\pi \cdot 3 f_{bw}} + 1}$, such that sufficient phase lead is generated around f_{bw} . In particular, the zero is placed at $\frac{1}{3} f_{bw}$, while the pole is placed at $3 f_{bw}$. Next, p_{lead} is adjusted such that $|GK_{lead}(2\pi f_{bw})| = 1$. This lead filter generates a phase margin at f_{bw} . Indeed, the phase corresponding to the rigid-body dynamics in (1) is typically -180° , see Fig. 2, which leads to a phase margin of 0° .
3. Check stability using Nyquist plot of GK and include possible notch filters in the case where the flexible modes endanger robust stability.
4. Include integral action through $K = K_{int} K_{lead}$, with $K_{int} = \frac{s + 2\pi \frac{f_{bw}}{5}}{s}$. Since $GK \gg 0$ at low frequencies, $S \approx \frac{1}{GK}$, the integrator pole at $s = 0$ improves low frequency disturbance attenuation while not affecting phase margin due to the zero at $\frac{1}{5} f_{bw}$.

These steps are done iteratively based on Nyquist of GK and Bode plots of GK and closed-loop transfer functions. Most importantly, all these steps can be performed on the basis of frequency response functions. This is essential, since nonparametric frequency response function identification leads to fast and inexpensive models, while at the same time delivering much more accurate models for feedback control. This latter accuracy aspect is directly confirmed by Fig. 2, where the frequency response function takes into account additional artifacts such as delay in the entire actuation chain, which is essential for stability, i.e., it directly affects the phase margin. To show this, the controller design procedure is directly applied to the frequency response function of the system G , see Fig. 4. An interesting observation is that the frequency response function is typically accurate in the

crossover region, i.e., where $|L(2\pi f_{bw})| \approx 1$, which is the most important region for closed-loop stability. In sharp contrast, the frequency response function is very noisy at low frequencies and high frequencies, but these regions much less relevant for closed-loop control (Oomen 2018).

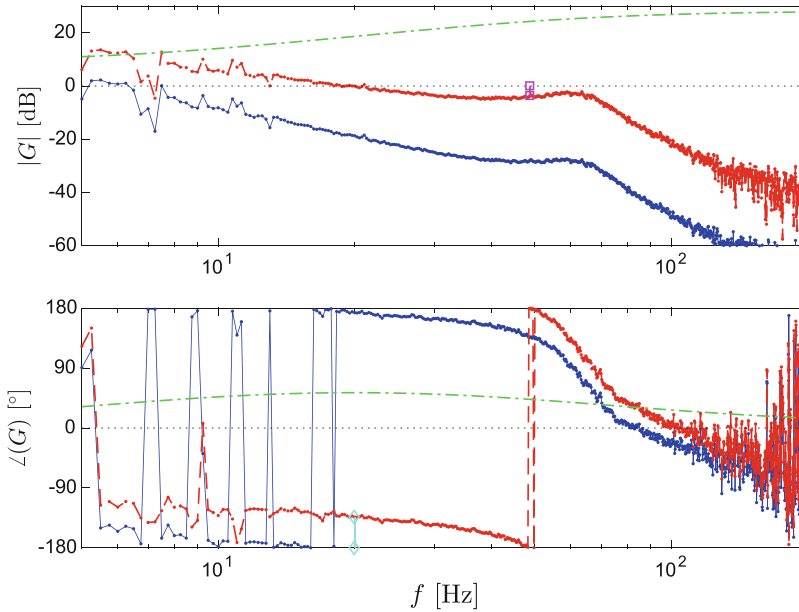
Multivariable Motion Control: A Systematic Design Framework

Most motion systems are multivariable, and commonly positioning is required in all six motion degrees of freedom. Since the design approach in the previous section focuses on single-input single-output systems, approaches developed particularly for multivariable systems, including those based on $\mathcal{H}_2$ or $\mathcal{H}_\infty$ optimization, may become appealing. However, these approaches require parametric models, and these models are user-intensive and time consuming to obtain. Therefore, a systematic motion feedback control design procedure based on frequency response functions has been developed that extends the loop-shaping procedure in the preceding approach toward multivariable systems. This procedure consists of the following steps.

Procedure 1 Advanced motion control design procedure.

1. Interaction analysis. Is the system decoupled?
 - Yes: independent SISO design. No: next step
 2. Static decoupling. Is the system decoupled?
 - Yes: independent SISO design. No: next step
 3. Decentralized MIMO design: loop-closing procedures
 - Robustness for interaction, e.g., using factorized Nyquist design
 - Design for interaction, e.g., sequential loop closing design
 Not successful? Next step
 4. Optimal & robust control
 - Design of centralized controller using a parametric model and optimization algorithms
-

Details on the procedure are provided in, e.g., Oomen (2018, Section 4), where many technical steps from Skogestad and Postlethwaite (2005) are employed. The main point is that the overall design complexity gradually increases.



Control for Precision Mechatronics, Fig. 4 Feedback controller tuning. Identified system G frequency response function (solid blue), loop-gain GK_{lead} with lead con-

troller (dashed red), lead controller K_{lead} (dash-dotted green), gain margin (magenta square), phase margin (cyan diamond)

Therefore, the complexity is only increased if necessitated by the control requirements. In particular, Step 1 – Step 3 are based on inexpensive frequency response functions and essentially involve an increased complexity in controller tuning. Step 4 involves a major increase in complexity from two perspectives: it requires an accurate parametric model in view of the control objectives and it requires the specification of all performance and robustness objectives in a single scalar criterion.

Feedforward

Feedforward Control Structure

Essentially, the main goal of the feedback controller in the preceding section is to deal with uncertainty, both in terms of disturbances, e.g., v in (2), and in terms of system dynamics, e.g., residual $(1 - GF)r$ due to incomplete knowledge of G in the design of feedforward control. In sharp contrast, the goal of feedforward control is

to obtain an accurate feedforward F that compensates for the known reference signal r .

The design of the feedforward controller is based on the physical model (1), and a systematic manual tuning approach is developed by exploiting (2). To exemplify the approach, note that the model (1) for single-input single output systems at low frequencies approximately corresponds to rigid-body behavior:

$$\frac{1}{ms^2}, \quad (5)$$

where $m \in \mathbb{R}_+$ corresponds to the mass of the system. Therefore, a candidate feedforward controller F is $F = k_{fa}s^2$, which leads to the feedforward signal $f(s) = k_{fa}s^2r(s)$ corresponding to $f(t) = k_{fa}\frac{d^2r(t)}{dt^2}$. This feedforward therefore is also referred to as acceleration feedforward, being the second derivative of the reference signal $r(t)$. Then, the first term in (2) becomes

$$e = S(1 - GF)r = S\left(1 - \frac{k_{fa}}{m}\right)r \quad (6)$$

Manual Tuning Approach

In (6), a feedforward structure has been selected, which clearly minimizes the error if $k_{fa} = m$. However, accurate knowledge of m is typically not available. The main idea is to directly use experimental data to estimate the optimal value of k_{fa} . To this end, assume that a stabilizing feedback controller is implemented as in the preceding section, typically of the form K_{lead} , yet without implementing the integral action K_{int} . This implies that at low frequencies, where $|GK| \gg 1$,

$$S \approx \frac{1}{GK} \approx cs^2 \quad \text{for frequencies} \\ \text{where } |GK| \gg 1, \quad (7)$$

with c a constant. Combining this with (6) leads to

$$e = cs^2 \left(1 - \frac{k_{fa}}{m}\right) r, \quad (8)$$

and after taking an inverse Laplace transform, this becomes

$$e(t) = c \left(1 - \frac{k_{fa}}{m}\right) \frac{d^2 r(t)}{dt^2}. \quad (9)$$

Equation (9) now directly reveals how to tune the coefficient k_{fa} : select it such that the measured error $e(t)$ is not correlated with the acceleration signal $\frac{d^2 r(t)}{dt^2}$. This is done by performing an experiment under normal operating conditions, i.e., a servo task with a certain reference r implemented. Typically, the experimental length for motion systems is in the order of seconds; hence, this procedure can be performed in a short amount of time. This allows to directly tune k_{fa} , typically through a manual adaptation. This is straightforward, since from (9) the error depends affinely on k_{fa} .

This idea of tuning naturally extends to more feedforward components, and a typical feedforward controller consists of

$$f(t) = k_{fc} \text{sign} \left(\frac{dr(t)}{dt} \right) + k_{fv} \frac{dr(t)}{dt} \\ + k_{fa} \frac{d^2 r(t)}{dt^2} + k_{fs} \frac{d^4 r(t)}{dt^4}, \quad (10)$$

where k_{fc} and k_{fv} compensate for Coulomb and viscous friction, and k_{fs} is a snap feedforward, aiming to compensate for the compliance of the flexible dynamics in (1). Indeed, at low frequencies, (1) becomes

$$\frac{1}{ms^2} + D, \quad (11)$$

where $D \in \mathbb{R}$ is a static term that corresponds to the low frequency behavior of the flexible modes in (1), i.e., below the resonance frequencies ω_i , $i = n_{rb} + 1, \dots, n_s$. Next, since the essential mechanism is to select the feedforward controller F as the inverse of the model, inverting (11) leads to

$$\left(\frac{1}{ms^2} + D \right)^{-1} = \frac{ms^2}{1 + ms^2 D} \\ = ms^2 - \frac{m^2 s^4 D}{1 + ms^2 D}. \quad (12)$$

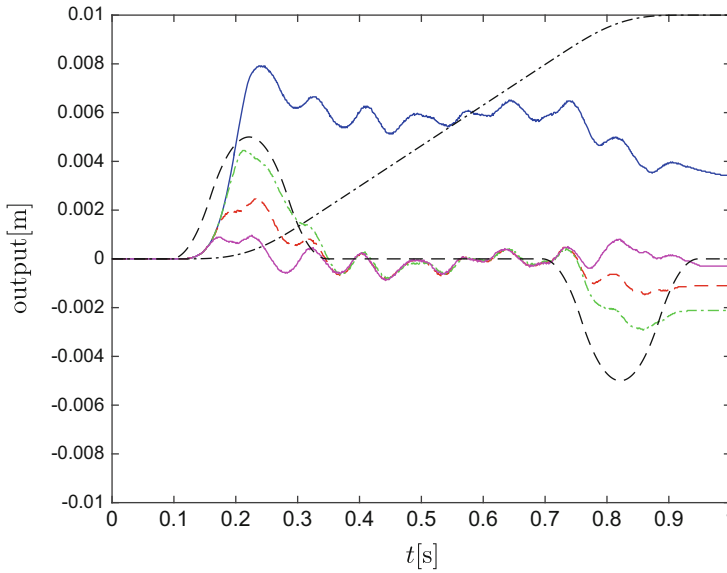
Again considering this inversion at low frequencies implies that the limit $s \downarrow 0$ leads to the approximation

$$ms^2 - m^2 D s^4, \quad (13)$$

thereby extending the earlier acceleration feedforward toward a fourth derivative, which is referred to as snap feedforward. Practically, this snap feedforward compensates for the compliance in mechanical systems.

The tuning of these coefficients involves the selection of an appropriate reference, typically consisting of equal parts of acceleration, constant velocity, and deceleration. Then, these are tuned systematically in the typical order: k_{fc} , k_{fv} , k_{fa} , k_{fs} .

In Fig. 5, this procedure is applied to the printer system in Fig. 1. Here, the parameters k_{fc} and k_{fv} are already set to their optimal value. Next, the value of k_{fa} is varied. It can clearly be observed that an incorrect value leads to a strong correlation between the reference and the measured error profile, see (9). The value of k_{fa} is then varied until there is no correlation between the acceleration profile.



Control for Precision Mechatronics, Fig. 5 Feed-forward controller tuning. Error using feedback only (solid blue). Next, friction feedforward has been optimally tuned, and acceleration feedforward tuning is further investigated. Indeed, note that the error profile (dash-dotted green) has a strong correlation with the acceleration

profile. Increasing the parameter value leads to a reducing error, from dash-dotted green toward dotted magenta which is considered the optimal tuning. Also shown are the scaled acceleration profile (black dashed) and scaled reference profile r (black dash-dotted).

Automated Tuning and Learning

In the preceding section, a systematic approach to manually tune feedforward signals has been outlined. Recently, this approach has also been extended toward automated tuning of feedforward controllers from data. Two important classes can be distinguished:

1. Approaches that learn a signal f in Fig. 3 directly, including iterative learning control Bristow et al (2006)
2. Approaches that learn the parameters in a parameterized form such as (10), including iterative learning control with basis functions and instrumental variable identification based, see Blanken et al (2017) for an overview as well as experimental results for tuning multiple parameters as in (10)

While the former approaches have a larger optimization space, and can therefore potentially

achieve a smaller error e , these approaches lead to a high sensitivity for setpoint changes r . Indeed, the second class of approaches is invariant under a change of setpoint, which is a very common requirement in motion control applications, where a class of references trajectories is common.

Summary and Future Directions

Control in precision mechatronics involves a deep understanding of the physical system properties, control theory, and design engineering aspects, which is reflected by the systematic design procedure outlined in this entry. This procedure forms the basis for successful motion control.

In the near envisaged future, an immense increase in system complexity is foreseen, where it is expected that more steps are required in Procedure 1. Indeed, system complexity will dras-

tically increase due to the following envisaged developments, see Oomen (2018) for details.

- Motion control will become a multiphysics problem. For instance, increasing accuracy requirements in the sub-nanometer range lead to a situation where thermal aspects become increasingly important. These thermal aspects lead to deformations, which will lead to an interaction with motion control. Already active thermal control and setpoint adjustment for thermal deformations are being developed.
- Subsystems will interact. In high-tech systems, multiple motion systems cooperatively perform tasks. For instance, in modern wafer scanners for lithographic integrated circuit production, two precision systems have to position relative to each other, directly leading to coupling.
- Many actuators and sensors will be used. In particular, the flexible dynamics in (1) limit the achievable crossover frequency f_{bw} . By using additional inputs and outputs, these flexible dynamics can be actively controlled. However, this leads to an increased complexity of the control design problem.
- Deformations lead to a situation where performance variables are not directly measurable. In particular, the flexible dynamics in (1) lead to a situation where flexible dynamics lead to deformation in between different locations on the mechanical structure. Since the sensors can typically not be placed exactly at the desired location, e.g., due to the presence of a product, this implies that performance cannot be measured directly. This leads to an inferential motion control problem, essentially introducing another set of unmeasurable performance signals into the control problem.

From a feedforward perspective, the mechatronic system design with embedded controller implementations leads to the availability of large amounts of sensor data. This facilitates the learning from data in complex motion systems. These algorithms, which have to be applicable to massively multivariable systems, are foreseen to revitalize existing machines through inexpensive,

smart updates, that lead to continuous control performance improvement during normal operation through optimizing f in Fig. 3.

Summarizing, positioning systems have a key role in engineered systems, and their role is foreseen to further increase in the near future. Smart control solutions will enable a further increase toward unparalleled accuracy and speed and create opportunities for revolutionary machine designs. This will enable manufacturing of user-customizable and unimaginable products and scientific instruments that facilitate scientific discoveries in nanotechnology, biotechnology, and space exploration.

Cross-References

- ▶ [Mechanical Design and Control for Speed and Precision](#)
- ▶ [Nonparametric Techniques in System Identification](#)
- ▶ [PIDs and Biquads: Practical Control of Mechatronic Systems](#)

Bibliography

- Åström KJ, Murray RM (2008) Feedback systems: an introduction for scientists and engineers. Princeton University Press, Princeton
- Blanken L, Boeren F, Bruijnen D, Oomen T (2017) Batch-to-batch rational feedforward control: from iterative learning to identification approaches, with application to a wafer stage. *IEEE Trans Mechatron* 22(2): 826–837
- Bristow DA, Tharayil M, Alleyne AG (2006) A survey of iterative learning control: a learning-based method for high-performance tracking control. *IEEE Control Syst Mag* 26(3):96–114
- Devasia S, Eleftheriou E, Moheimani S (2007) A survey of control issues in nanopositioning. *IEEE Trans Control Syst Technol* 15(5):802–823
- Fleming AJ, Leang KK (2014) Design, modeling and control of nanopositioning systems. Springer, Cham
- Gawronski WK (2004) Advanced structural dynamics and active control of structures. Springer, New York
- Lambrechts P, Boerlage M, Steinbuch M (2005) Trajectory planning and feedforward design for electromechanical motion systems. *Control Eng Pract* 13: 145–157
- Lee HS, Tomizuka M (1996) Robust motion controller design for high-accuracy positioning systems. *IEEE Trans Ind Electron* 43(1):48–55

- Munnig Schmidt R, Schitter G, van Eijk J (2011) The design of high performance mechatronics. Delft University Press, Delft
- Ohnishi K, Shibata M, Murakami T (1996) Motion control for advanced mechatronics. *IEEE Trans Mechatron* 1(1):56–67
- Oomen T (2018) Advanced motion control for precision mechatronics: control, identification, and learning of complex systems. *IEEJ IEEE Trans Ind Appl* 7(2): 127–140
- Pintelon R, Schoukens J (2012) System identification: a frequency domain approach, 2nd edn. IEEE Press, New York
- Skogestad S, Postlethwaite I (2005) Multivariable feedback control: analysis and design, 2nd edn. John Wiley & Sons, West Sussex
- Steinbuch M, Norg ML (1998) Advanced motion control: An industrial perspective. *Eur J Control* 4(4):278–293
- Voorhoeve R, van der Maas A, Oomen T (2018) Non-parametric identification of multivariable systems: A local rational modeling approach with application to a vibration isolation benchmark. *Mech Syst Signal Process* 105:129–152

Control Hierarchy of Large Processing Plants: An Overview

César de Prada

Departamento de Ingeniería de Sistemas y Automática, University of Valladolid, Valladolid, Spain

Abstract

This entry provides an overview of the so-called automation pyramid, which organizes the different types of control and operation tasks in a processing plant in a set of interconnected layers, from basic control and instrumentation to plant-wide economic optimization. These layers have different functions, all of them necessary for the optimal functioning of large processing plants.

Keywords

Control hierarchy · Automation pyramid · Plant-wide control · MPC · RTO · Optimization

Introduction

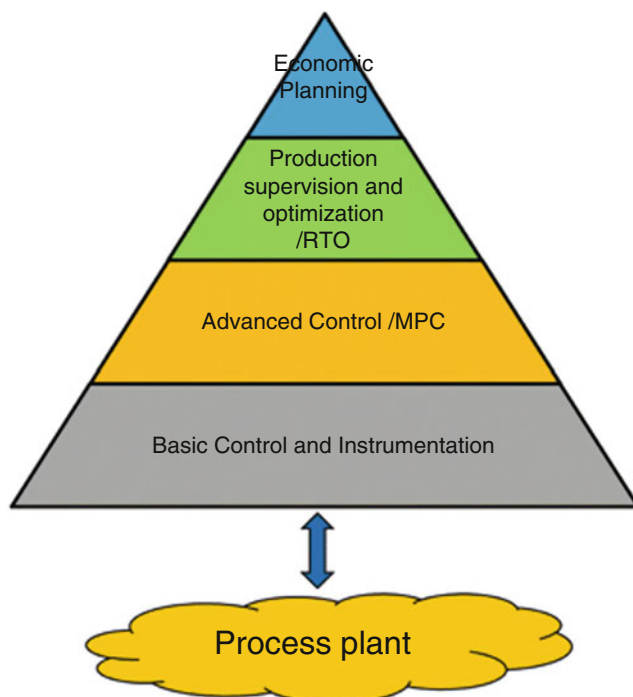
Operating a process plant is a complex task involving many different aspects ranging from the control of individual pieces of equipment or process units to the management of the plant or factory as a whole, including interactions with other plants or suppliers.

From a control point of view, the corresponding tasks are traditionally organized in several layers, with the ones closer to the physical processes placed in the bottom and those closer to plant-wide management in the top, forming the so-called automation pyramid represented in Fig. 1.

The process industry currently faces many challenges, originating from factors such as increased competition among companies, better global market information, new environmental regulations and safety standards, improved quality, or energy efficiency requirements. Many years ago, the main tasks in the operation of a process plant were associated with the correct and safe functioning of the individual process units and to the global management of the factory from the point of view of organization and economy. Therefore, only the lower and top layers of the automation pyramid were realized by computer-based systems, whereas the intermediate tasks were largely performed by human operators and managers. Nevertheless, more and more, the automation of the intermediate layers and the integration among all decision levels are gaining importance in order to face the above mentioned challenges.

Above the physical plant represented in the bottom of Fig. 1, there is a layer related to instrumentation and basic control, devoted to obtaining direct process information and stabilizing the plant and maintaining selected process variables close to their desired targets by means of local controllers. Motivated by the need for more efficient operation and better-quality assurance, an improvement of this basic control can be obtained using control structures such as cascades, feed forwards, ratios, and selectors. Sometimes, this is called advanced control in industry, but not in

Control Hierarchy of Large Processing Plants: An Overview, Fig. 1 The automation pyramid



C

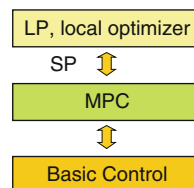
academia, where the term is reserved for more sophisticated controls.

A big step forward took place in the control field with the introduction of model-based predictive control (MPC) in the late 1970s and 1980s (Industrial MPC of Continuous Processes; Camacho and Bordóns 2004). MPC aims at commanding a process unit as a whole considering simultaneously all its manipulated and controlled variables. It uses a process model in order to compute the control actions that optimize a control performance index considering all interactions among its variables, disturbances, and process constraints. MPC is built on top of the basic control loops and partly replaces the complex control structures of the advanced control layer adding new functionalities and better control performance.

The improvements in control quality and the management of constraints and interactions of the model-predictive controllers open the door for the implementation of local economic optimization in an upper layer. Linked to the MPC

Control Hierarchy of Large Processing Plants: An Overview, Fig. 2

MPC implementation with a local optimizer



controller, and taking advantage of its model, an optimizer may look for the best operating point of the unit by computing the controller set points that optimize an economic cost function of the process considering the operational constraints of the unit. This task is usually formulated and solved as a linear programming (LP) problem, i.e., based on linear or linearized economic models and cost function (see Fig. 2).

A natural extension of these ideas was to consider the interrelations among the different parts of a processing plant and to look for the steady-state operating point that provides the best economic return and minimum energy expense or optimizes any other economic criterion

while satisfying the global production aims and constraints. These optimization tasks are known as real-time optimization (RTO) (Real-Time Optimization of Industrial Processes) and form another layer of the automation pyramid. In batch plants, this layer involves production scheduling, focused in determining when and where tasks should be performed in order to maximize a certain cost function while respecting operation constraints.

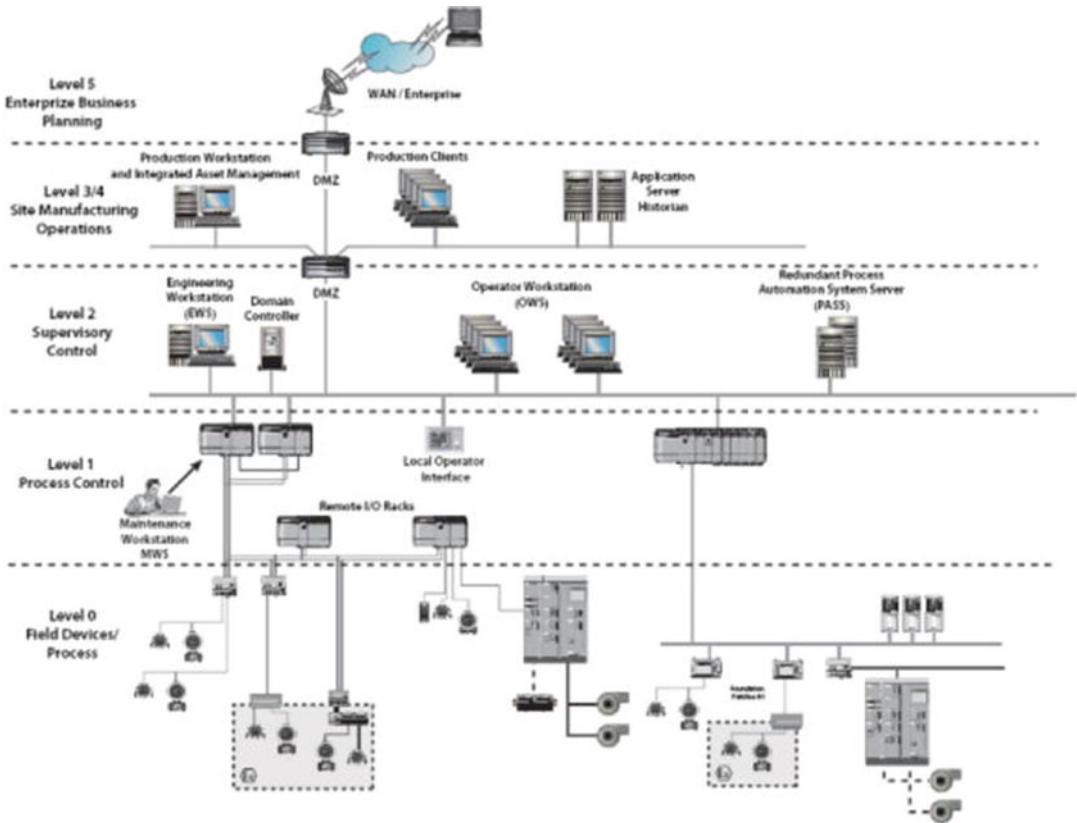
Finally, when we consider the operation of the whole plant, obvious links between the RTO and the planning and economic management of the company appear. In particular, the organization and optimization of the flows of raw materials, purchases, etc., involved in the supply chains, present important challenges that are placed in the top layer of Fig. 1.

This entry provides an overview of the different layers and associated tasks so that the

reader can place in context the different types of controllers and related functionalities and tools. It also aims at summarizing current trends in process control and process operation, focusing the attention toward the higher levels of the hierarchy and the optimal operation of large-scale processes.

Industrial Implementation

The implementation of the tasks and layers previously mentioned in a process factory is possible nowadays due to important advances in many fields, such as modeling and identification, control and estimation, optimization methods, and, in particular, software tools, communications, and computing power. Today, it is rather common to find in many process plants an information network that follows also a pyramidal structure represented in Fig. 3.



Control Hierarchy of Large Processing Plants: An Overview, Fig. 3 Information network in a modern process plant

At the bottom, there is the instrumentation layer that includes not only sensors and actuators connected by the classical analog 4–20 mA signals (possibly enhanced by the transmission of information to and from the sensors using the HART protocol) but also digital field buses and wireless networks connected to smart transmitters and actuators that deliver improved information and intelligence. New functionalities, such as remote calibration, filtering, self-test, and disturbance compensation, provide more accurate measurements that contribute to improving the functioning of local controllers, in the same way as the new methods and tools available nowadays for instrument monitoring and fault detection and diagnosis. The increased installation of wireless transmitters and the advances in analytical instrumentation will lead in the near future, without doubt, to the development of a stronger information base to support better decisions and operations in the plants.

Information from transmitters is collected in control rooms that perform the control tasks. Many of them are equipped with distributed control systems (DCS) that implement monitoring and control tasks. Field signals are received in the control cabinets where a large number of microprocessors execute the data acquisition and regulatory control tasks, sending signals back to the field actuators. Internal buses connect the controllers with the computers that support the displays of the human-machine interface (HMI) for the plant operators of the control room. In the past, DCS were mostly in charge of the regulatory control tasks, including basic control, alarm management, and historians, while interlocking systems related to safety and sequences related to batch operations were implemented in programmable logic controllers (PLCs): Programmable Logic Controllers. Today, the differences between both types of systems are not so clear, due to the increase in computing power of the PLCs and the added functionalities of the DCS. Safety instrumented systems (SIS) for the maintenance of plant safety are usually implemented in dedicated PLCs, if not hardwired, but for the rest of the functions, a combination of PLC-like processors with I/O cards and SCADAs

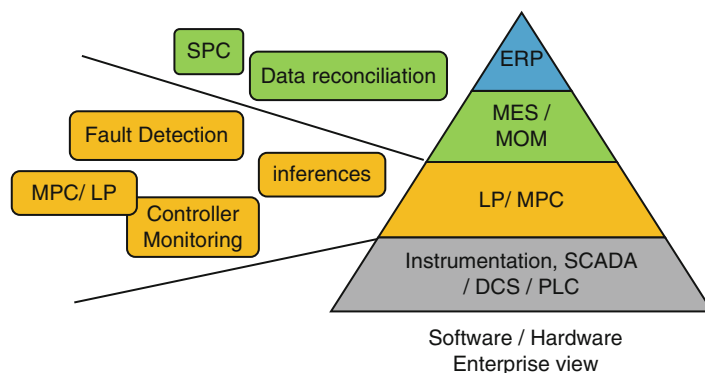
(Supervision, Control, and Data Acquisition Systems) is the prevailing architecture. SCADAs act as HMI and information systems collecting large amounts of data that can be used at other levels with different purposes. At the same time, as these systems are based on standard operating system (e.g., Windows), they are more vulnerable to external software attacks, which has brought the topic of cybersecurity of the control systems to the front line of the concerns affecting safety and security of process plants.

Above the control layer, using the information stored in the SCADA as well as other sources, there is an increased number of applications covering diverse fields. Figure 3 depicts the perspective of the computing and information flow architecture and includes a level called supervisory control, placed in direct connection with the control room and the production tasks. It includes, for instance, MPC with local optimizers, statistical process control (SPC) for quality and production supervision (Multiscale Multivariate Statistical Process Control), data reconciliation, inferences and estimation of unmeasured quantities, fault detection and diagnosis, or performance controller monitoring (Controller Performance Monitoring) (CPM).

The information flow becomes more complex when we move up the basic control layer, looking more like a web than a pyramid when we enter the world of what can be called generally as asset (plant and equipment) management: a collection of different activities oriented to sustain performance and economic return, considering their entire cycle of life and, in particular, aspects such as maintenance, efficiency, or production organization. Above the supervisory layer, one can usually distinguish at least two levels denoted generically as manufacturing execution systems (MES) and enterprise resource planning (ERP) (Scholten 2009) as can be seen in Fig. 4.

MES are information systems that support the functions that a production department must perform in order to prepare and to manage work instructions, schedule production activities, monitor the correct execution of the production process, gather and analyze information about the production process, and optimize procedures.

Control Hierarchy of Large Processing Plants: An Overview,
Fig. 4 Software/hardware view



Notice that regarding the control of process units, up to this level, no fundamental differences appear between continuous and batch processes. But at the MES level, in which RTO of Fig. 1 is inserted, many process units may be involved, and the tools and problems are different, the main task in batch production being the optimal scheduling of those process units (Scheduling of Batch Plants; Mendez et al. 2006).

MES are part of a larger class of systems called manufacturing operation management (MOM) that cover not only the management of production operations but also other functions such as maintenance, quality, laboratory information systems, or warehouse management. One of their main tasks is to generate elaborate information, quite often in the form of key performance indicators (KPIs), with the purpose of facilitating the implementation of corrective actions.

Finally, ERP systems represent the top of the pyramid, corresponding to the enterprise business planning activities that allow assigning global targets to production scheduling. For many years, it has been considered to be out of the scope of the field of control, but nowadays, more and more, supply chain management is viewed and addressed as a control and optimization problem in research.

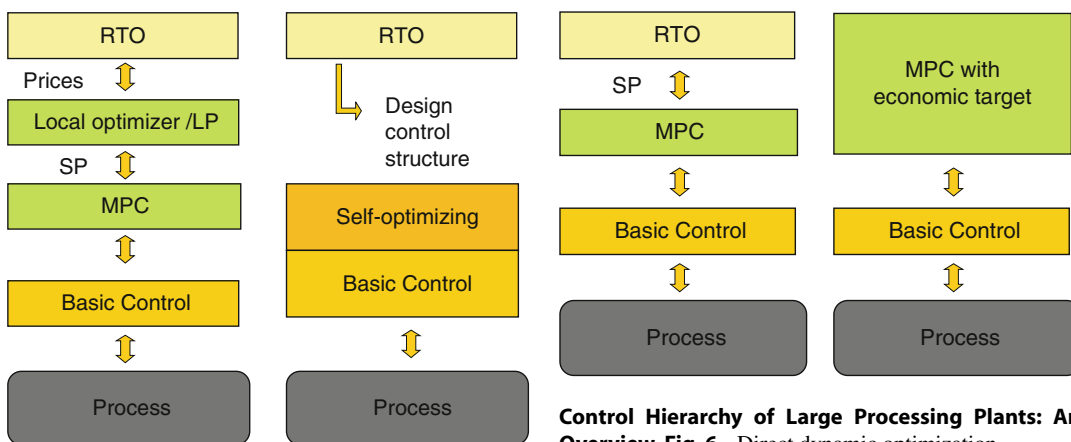
Future Control and Optimization at Plant Scale

Going back to Fig. 1, the variety of control and optimization problems increases as we move up in the control hierarchy, entering the field

of dynamic process operations which considers not only individual process units but also larger sets of equipment or whole plants. Examples of challenges at the RTO (or MES) level are optimal management of shared resources or utilities deciding on their best use or distribution, production bottleneck avoidance, optimal energy use or maximum efficiency, smooth transitions against production changes, optimal process operation taking into account its long-term effects on efficiency degradation (fouling, catalyst degradation, etc.) and maintenance, etc.

Above, we have mentioned RTO as the most common approach for plant-wide optimization. Normally, RTO systems perform the optimization of an economic cost function using a nonlinear process model in steady state and the corresponding operational constraints to generate targets for the control systems on the lower layers. The implementation of RTO provides consistent benefits by looking at the optimal operation problem from a plant-wide perspective. Nevertheless, in practice, when MPCs with local optimizers are operating the process units, many coordination problems appear between these layers, due to differences in models and targets, so that driving the operation of these process units in a coherent way with the global economic targets is an additional challenge.

A different implementation perspective is taken by the so-called self-optimizing control (Fig. 5 right, Skogestad 2000) that, instead of implementing the RTO solution online, uses it to design a control structure that assures a near-optimum operation if some specially chosen variables are maintained closed to their targets.



Control Hierarchy of Large Processing Plants: An Overview, Fig. 5 Two possible implementations of RTO

As with any model-based approach, the important problem of how to implement or modify the theoretical optimum computed by RTO so that the optimum computed with the model and the real optimum of the process coincide in spite of model errors, disturbances, etc., emerges. A common choice to deal with this problem is to update periodically the model using parameter estimation methods or data reconciliation with plant data in steady state, but this is known not to solve the problem if there are structural errors in the model. Also, uncertainty can be explicitly taken into account by considering different scenarios and optimizing the worst case, but this is conservative and does not take advantage of the plant measurements. Along this line, there are proposals of other solutions such as modifier-adaptation methods that use a fixed model and process measurements to modify the optimization problem so that the final result corresponds to the process optimum (Marchetti et al. 2009) or the use of stochastic optimization where several scenarios are taken into account and future decisions are used as recourse variables (Lucia et al. 2013).

RTO is formulated in steady state but, in practice, most of the time the plants are in transients, and there are many problems, such as start-up or batch optimization, that require a dynamic formulation. A natural evolution in this direction is to integrate nonlinear MPC with economic optimization so that the target of the NMPC

is not set point following but direct economic optimization as in the right-hand side of Fig. 6: Economic Model-Predictive Control and Model-Based Performance Optimizing Control (Engell 2007).

The type of problems that can be formulated within this framework is very wide, as well as the possible fields of application. Processes with distributed parameter structure or mixtures of real and on/off variables, batch and continuous units, statistical distribution of particle sizes or properties, etc., give rise to special type of NMPC problems (see, e.g., Lunze and Lamnabhi-Lagarrigue 2009), but a common characteristic of all of them is the fact that they are computational intensive and should be solved taking into account the different forms of uncertainty always present.

Control and optimization are nowadays inseparable essential parts of any advanced approach to dynamic process operation. Progress in the field and spreading of the industrial applications are possible, thanks to the advances in optimization methods and tools and computing power available on the plant level, but implementation is still a challenge from many points of view, not only technical. Developing efficient process models for advanced control and optimization is perhaps the most difficult point, but, in addition, few suppliers offer commercial products at this level, and finding optimal operation policies for a whole factory is a complex task that requires taking into consideration many aspects, cooperation among departments and elaborate information not available directly as process measurements. Regarding

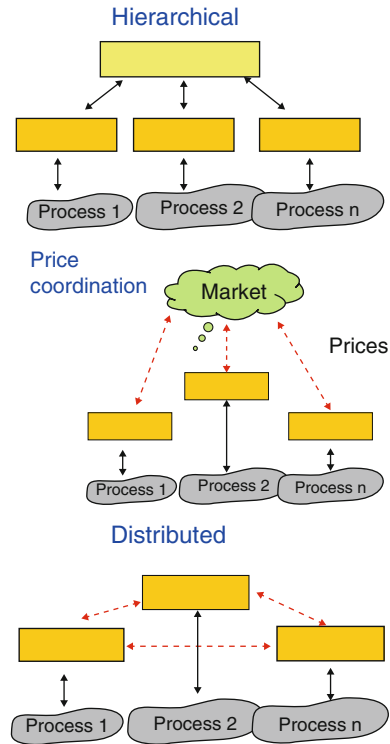
Control Hierarchy of Large Processing Plants: An Overview, Fig. 6 Direct dynamic optimization

this point, data analysis techniques may help in obtaining that information complementing well-established modeling methodologies.

Another difficulty about implementation of advanced solutions resides in the fact that most current commercial products do not facilitate the integration of new methods or best-in-class, third-party components into fundamentally closed, proprietary systems, in spite of the facilities provided by tools like *OPC*. Moreover, today's systems generally lack the built-in cybersecurity needed to protect operations or equipment assets.

This has originated proposals for changing the hierarchical architecture of Fig. 1, where the information flows vertically inside quite closed boundaries, by other ones oriented to facilitate the integration of external hardware/software into the control systems. One example is the NAMUR Open Architecture (*NOA*), which claims for a second open communication channel at each automation pyramid level for additional functionalities that are not part of the core control components. In the same line, the architecture proposed by the Open Process Automation Forum (*OPAF*) within the Open Group replaces the automation pyramid by a flat structure where each production oriented IT-system can communicate directly with each other IT-system in a secure, reliable and timely manner. The collection of standards defined as O-PAS (Open Process Automation Standards) define specifications for an open, interoperable, secure by design automation architecture. The standards enable the development of systems composed of cohesive, loosely coupled, functional elements acquired from different suppliers and easily integrated via a structured modular architecture.

Within the new elements that likely we will see included in the systems supported by these new architectures are digital twins of the plants. They will combine detailed simulation of the processes with updated real-time information of the process and databases of different functional elements, helping in the decision-making by providing integrated information and online what-if analysis.



Control Hierarchy of Large Processing Plants: An Overview, Fig. 7 Hierarchical, price coordination, and distributed approaches

Another point related to architectures is the fact that implementation of large advanced control or optimization solutions in real time may require breaking the associated optimization problem in more manageable subproblems that can be solved in parallel. This leads to several local controllers/optimizers, each one solving one subproblem involving variables of a part of the process and linked by some type of coordination. These solutions offer a new point of view of the control hierarchy. Typically, three types of architectures are mentioned for dealing with the problem, represented in Fig. 7: In the hierarchical approach, coordination between local controllers is made by an upper layer that deals with the interactions, assigning targets to them. In price coordination, the coordination task is performed by a market-like mechanism that assigns different prices to the cost functions of every local controller/optimizer as a way of coordinating their actions. Finally, in the distributed approach,

the local controllers coordinate their actions by interchanging information about its decisions or states with neighbors (Scattolini 2009).

Regarding topics considered in the different problems that have been mentioned, perhaps there are two types of them that draw some of the trends that can be observed in process operation optimization. One considers the long-term effects of current decisions on the useful life and future state of the processes. Here, we refer to phenomena such as heat-exchanger fouling or catalyst degradation in chemical reactors that are affected by the way one operates a process. Notice that modes of operation that may be optimum considering only cost and benefits at present may not be so optimum in the long term due to the additional cost of more frequent maintenance that may imply. Models of degradation and more sophisticated mathematical formulations covering large time scales are required to deal with them, but there is no doubt that they represent a more realistic view of what is needed in the industrial practice.

The second trend is associated with the integration of decisions at different levels of the automation pyramid, in particular between production scheduling with upper- and lower-level decisions, ranging from logistics and supply chains to MPC and RTO. This points at generating optimal production schedules able to make an efficient use of the resources and adapt themselves to the changing environmental conditions, internal or external, such as varying electricity prices, products demand, transport delays, or maintenance works. The problems of this type are formulated normally as large-scale Mix Integer Programming ones, involving many real and binary variables and, on addition to computing power, efficient model formulation and perhaps decomposition techniques for its solution, claim for more integrated flows of information in the factories and new open control architectures for implementation like the ones mentioned above (NAMUR, OPAF).

Summary and Future Research

Process control is a key element in the operation of process plants. At the lowest layer, it can

be considered a mature, well-proven technology, even if there are still open problems such as control structure selection or automatic controller tuning. But the range of open problems increases in the upper layers of the hierarchy: from modelling and estimation to efficient large-scale optimization, robustness against uncertainty and the integration of the decisions at the different levels. This leads to new defies and research problems which solution will bring large improvements in the efficiency and operability of process plants. Different methods and architectures are being proposed to meet the challenges. Their implementation at industrial scale requires efforts and people with multidisciplinary background, but they will pave the road to the future leading to fully automated and efficient process factories.

Cross-References

- [Controller Performance Monitoring](#)
- [Economic Model Predictive Control](#)
- [Industrial MPC of Continuous Processes](#)
- [Model-Based Performance Optimizing Control](#)
- [Multiscale Multivariate Statistical Process Control](#)
- [Programmable Logic Controllers](#)
- [Real-Time Optimization of Industrial Processes](#)
- [Scheduling of Batch Plants](#)

Bibliography

- Camacho EF, Bordóns C (2004) Model predictive control. Springer, London, pp 1–405. ISBN:1-85233-694-3
- de Prada C, Gutierrez G (2012) Present and future trends in process control. *Ingeniería Química* 44(505):38–42. Special edition *ACHEMA*, ISSN:0210–2064
- Engell S (2007) Feedback control for optimal process operation. *J Process Control* 17:203–219
- Engell S, Harjunkoski I (2012) Optimal operation: scheduling, advanced control and their integration. *Comput Chem Eng* 47:121–133
- Lucia S, Finkler T, Engell S (2013) Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *J Process Control* 23:1306–1319
- Lunze J, Lamnabhi-Lagarrigue F (2009) *HYCON handbook of hybrid systems control*. Theory, tools, appli-

- cations. Cambridge University Press, Boca Raton. ISBN:978-0-521-76505-3
- Marchetti A, Chachuat B, Bonvin D (2009) Modifier-adaptation methodology for real-time optimization. *Ind Eng Chem Res* 48(13):6022–6033
- Mendez CA, Cerdá J, Grossmann I, Harjunkski I, Fahl M (2006) State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput Chem Eng* 30:913–946
- NAMUR (2017) Open Architecture. <https://www.namur.net/en/focus-topics/namur-open-architecture.html>
- OPC (2020) Foundation. <https://opcfoundation.org/>
- Scattolini R (2009) Architectures for distributed and hierarchical model predictive control – a review. *J Process Control* 19:723–731
- Scholten B (2009) MES guide for executives: why and how to select, implement, and maintain a manufacturing execution system. ISA, Research Triangle Park. ISBN:978-1-936007-03-5
- Skogestad S (2000) Plantwide control: the search for the self-optimizing control structure. *J Process Control* 10:487–507
- The Open Group (2020). <https://www.opengroup.org/forum/open-process-automation-forum>

Control of Additive Manufacturing

Sandipan Mishra

Department of Mechanical, Aerospace, and Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract

Control of additive manufacturing (AM) processes involves delivery of material and energy to precise locations at prescribed times during the manufacturing process. While each process is distinct and thus the challenges associated with designing control algorithms are unique, *motion control* and *process control* are the two primary objectives in all AM processes. Standard motion control strategies are used to position nozzles, motion stages, and steering mirrors. Model-free feedback and feedforward design strategies such as PID and iterative learning control are often used for process control of low-level process variables. On the other hand, geometry and

part-level control strategies are designed through physics-based in-layer and layer-to-layer models built on momentum, mass, and/or energy balance equations.

Keywords

3D printing · Motion control · Process control · Model predictive control · Physics-based modeling · Gray-box modeling · Iterative learning control · 2D systems · Feedback linearization

Introduction

Additive manufacturing (AM), also called 3D printing, refers to a class of manufacturing processes where material is added layer upon layer to construct 3D objects. Recently, AM has seen significant increase in popularity for commercial as well as R&D applications. AM processes encompass several technologies that use different materials to build parts, including polymers, plastics, ceramics, and metals. As defined by ASTM Standard F2792-12a (ASTM International 2012), AM is “a process of joining materials to make objects from three-dimensional (3D) model data, usually layer upon layer.” Broadly speaking, AM processes may be classified into the following categories: (1) binder and material jetting, (2) directed energy deposition, (3) material extrusion, (4) powder bed fusion, (5) sheet lamination, and (6) vat polymerization. This classification is based on the type of material, the type of energy source, and/or the mechanism for energy and material delivery used in the AM process. Thus, *from a control standpoint, AM processes fundamentally require delivering a specified amount of material and energy at prescribed locations and time.* Naturally, the specific modality of the process dictates the mechanisms by which material and energy are delivered, and consequently the sensing, modeling, and control methodologies developed are significantly distinct for each process. The following sections briefly describe AM modeling and control paradigms for three candidate processes.

Binder and Material Jetting: Inkjet 3D Printing

Material and binder jetting AM processes consist of a moving nozzle depositing ink droplets either of the build material itself or a binder ink onto a bed of build material (In some configurations, the nozzle is held stationary while the substrate is moved by motion stages.), followed by a material curing step. One example of such a process is inkjet 3D printing, where 3D objects are manufactured by jetting polymeric inks layer upon layer with ultraviolet (UV) light or heat curing steps in between. The droplets are ejected from the nozzle through either (1) piezoelectric actuation, (2) thermal actuation, or (3) ultrasonic actuation. In commercial systems, the stage motion is executed by an industrial motion stage controller (with position or angle feedback), which may send triggers to the nozzle to eject droplets. The droplet generation process is open-loop, i.e., the number of layers to be deposited and the droplet patterns for each layer are determined in advance. Furthermore, the voltage profiles for ejecting droplets from the nozzle are determined through trial-and-error and pre-programmed for each print job.

Control of Droplet Ejection

For the traditional 2D ink-jet printing process, ink channel dynamics models that describe the relationship between the input voltage waveform and droplet characteristics (sensed through the piezo sensor itself and/or camera measurements

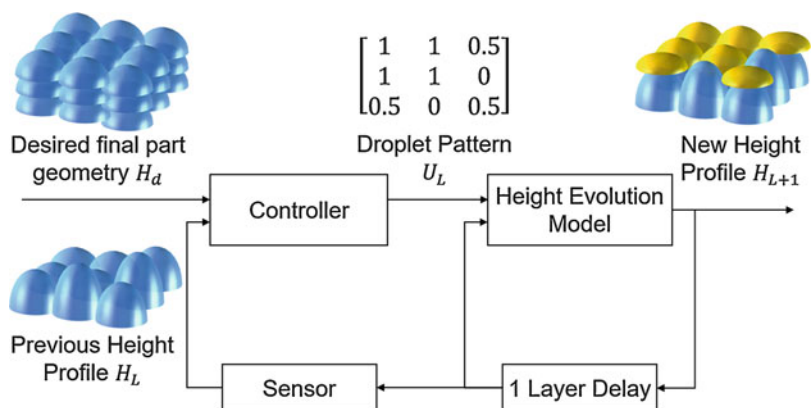
of the droplet ejection) have been studied extensively to design the waveforms for high speed and reliable droplet deposition (Kwon and Kim 2007). These models relate the input waveform to droplet volume and jetting frequency. The input waveform is optimized empirically to generate consistent prescribed droplet volumes and jetting frequency. Iterative learning control (ILC) has also been used for waveform optimization for inkjet nozzles (Wassink et al. 2005).

Geometry Control in Inkjet 3D Printing

Given a consistent and reliable droplet generation mechanism, the natural next step for process control in inkjet 3D printing is *geometry-level* control (Doumanidis and Skordeli 2000; Cohen and Lipson 2010; Lu et al. 2015). Geometry control strategies determine droplet patterns (and volume deposited at each location) based on in situ geometry measurements of the part being built.

Figure 1 illustrates the schematic of closed-loop control of geometry in inkjet 3D printing. The desired part geometry is denoted by $h_d \in \mathbb{R}^n$, and the current layer height profile is denoted by $h_L \in \mathbb{R}^n$, where L indicates the layer number and n is the number of grid points. h_L , h_d , and u_L represented the vectorized versions of the 2D height and input maps H_L , H_d , and U_L , respectively. The input sequence (droplet pattern) $u_L \in [V_{\min}, V_{\max}]^n$ and the current height h_L together determine the height profile for the subsequent layer, h_{L+1} . Droplet volumes at each location are limited by nozzle and ink properties, with bounds

Control of Additive Manufacturing, Fig. 1 A schematic block diagram of the closed-loop layer-to-layer printing process



given by $V_{\min}$ and $V_{\max}$. The control objective is to generate a droplet pattern u_{L+1} that minimizes the geometric tracking error $e_{L+1} = h_d - h_{L+1}$, based on feedback of the height profile, h_L . The control algorithm is designed based on layer-to-layer height evolution models of the printing process, which can be either physics-based, identified from input-output data (black box), or obtained from a combination of the two (gray-box).

Layer-to-Layer Models A generic *time-step wise* height evolution model is given by $h_{k+1} = f(h_k, u_k)$ where k denotes the time-step within a layer and u_k is the droplet volume at a given time-step. The layer-to-layer height evolution model is then obtained by lifting this equation N time-steps, where N is the number of time-steps in a single layer. Abusing notation, the layer-to-layer model can be written in the form $h_{L+1} = F(h_L, U_L)$, where U_L is the droplet volume vector for an entire layer and L is the *layer number*.

The simplest model describing time-step geometry evolution is a linear additive height profile model: $h_{k+1} = h_k + B_k u_k$, where B_k captures the effect of a deposited droplet, which depends on the nozzle location (which depends on time k) and the droplet shape (which is independent of k). We refer the reader to Hoelzle and Barton (2016) for a detailed development of this model. The corresponding lifted LTI layer-to-layer model is

$$h_{L+1} = h_L + \mathcal{B}U_L, \quad (1)$$

where $\mathcal{B} \in \mathbb{R}^{n \times n_u}$, n_u being the number of droplet deposition locations in each layer. Adding flow effects to this model results in layer-to-layer dynamics of the form

$$h_{L+1} = \mathcal{A}h_L + \mathcal{B}U_L, \quad (2)$$

where $\mathcal{A} \in \mathbb{R}^{n \times n}$ depends on the adjacency of grid points and the flowability of the ink (Guo et al. 2018). This model is still discrete-time LTI and suitable for control design based on linear system theory.

Model-Based Control Design A variety of feedback control algorithms may be designed based on the model in Eq. (2). However, given the need for enforcing explicit constraints on droplet size and the availability of sufficient time in between layers for computation, model predictive control (MPC) schemes are particularly well-suited for this application. Thus, an MPC problem of the following form can be posed (Rawlings and Mayne 2012):

$$U^* = \arg \min_U J(h_L, U) \quad (3)$$

$$\text{s.t. } \mathcal{E}U^T \leq c,$$

where $U = [u_{0|L} \dots u_{N-1|L}]$, $u_{i|L}$ indicates the i th layer control input in the receding horizon. The cost function J penalizes tracking error:

$$J(h_L, U_L) = (h_{N_c|L} - h_{d,N_c|L})^T P (h_{N_c|L} - h_{d,N_c|L}) + \sum_{i=0}^{N_c-1} \left[(h_{i|L} - h_{d,i|L})^T Q (h_{i|L} - h_{d,i|L}) \right], \quad (4)$$

where $h_{i|L}$ indicates the predicted height profile i layers into the receding horizon, which is obtained by propagating the model (2) i layers starting with $h_{0|L} = h_L$. N_c is the prediction (and control) horizon. P and Q are positive-definite matrices, Q is the state penalty matrix, and P is the terminal penalty matrix that can be designed to guarantee MPC stability. In (3), $\mathcal{E}$ and c are matrix and column vectors defined by

$$\mathcal{E} = \begin{bmatrix} E_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & E_{N-1} \end{bmatrix} \quad c = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{N-1} \end{bmatrix}, \quad (5)$$

where $E_i = [-I \ I]^T$ and $b_i = [-u_{\text{low}} \ u_{\text{high}}]^T$ that defines the upper and lower constraints on the input droplet size. The optimization is performed each layer, and $u_{0|L}^*$ is applied. This MPC prob-

lem can be solved using centralized or distributed methods. We refer the reader to Guo et al. (2018) for a detailed development.

Powder Bed Fusion: Selective Laser Melting (SLM)

Selective laser melting (SLM) is a powder-bed fusion AM process that employs a high-power laser to melt and fuse metal layer by layer to build a 3D part. In the SLM process, a recoater spreads a thin layer of metal powder over the build plate. A laser scans this metal powder, which locally melts and then solidifies. A combination of a lens optics and a computer-controlled galvo scanner is used to control the spot size and drive the focused laser beam in a specified pattern, respectively (shown in yellow arrows in Fig. 2). The build plate is then lowered using a motion stage controller, and the recoater spreads a new layer of powder on top of the recently melted material, and the process is repeated for the next layer.

There are two controllable variables in the typical SLM process, the galvo scan path and the laser power profile. These are operated in a decoupled but coordinated manner: the galvo is driven by internal position control feedback to follow a prescribed path, while the laser power profile is controlled separately (usually in open loop). Typical measurements for feedback from

the process include infrared (IR) cameras (shown in blue) and photodiodes (or pyrometers). There are also safety mechanisms built in, such as an O₂ analyzer that is used to trigger alarms and stop the process in case of loss of seal of the chamber.

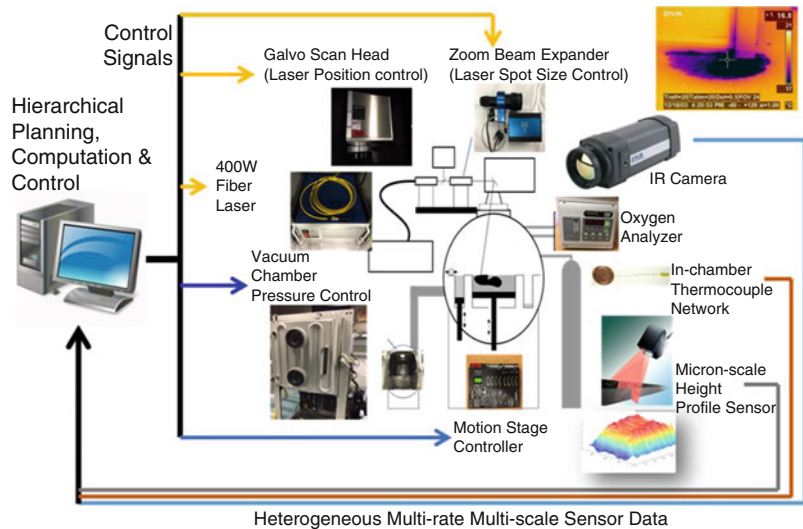
Feedforward Control Design For feedforward design, iterative learning control (ILC) can be used to find power profiles that maintain a constant melt pool geometry (y_d). Such ILC schemes are based on iterative refinement of the power profile $p(\cdot)$ based on error measurements $e(\cdot) = y_d - y(\cdot)$ obtained from prior iterations, based on an update law of the form

$$p_{j+1}(\cdot) = Q(p_j(\cdot) + Le_j(\cdot)) \quad (6)$$

where j represents the iteration number, L is the learning operator (often consisting of a gain and a forward shift), and Q is a low-pass filter for robustness. Repeating this update eventually yields the optimized power profile $p^*(\cdot)$

Model-Free Control Design The seminal experimental demonstration of *feedback* control of SLM to date is the PID controller presented in Craeghs et al. (2010). The control objective is to keep the melt pool size (area) constant. Two different feedback measurement mechanisms can be used to obtain a surrogate for the melt pool size: a high-speed CMOS camera and

Control of Additive Manufacturing, Fig. 2
Schematic of a typical SLM system and its components



a photodiode. The photodiode measurement is proportional to the melt pool area. The relationship between the input power $p(t)$ and the output melt pool area $y(t)$ can be approximated as a first-order transfer function $G(s) = \frac{K}{\tau s + 1}$, where K and τ are identified experimentally from step response measurements. The PID feedback control is

$$p(t) = p^*(t) + K_p e(t) + K_d \dot{e}(t) + K_i \int_0^t e(\tau) d\tau, \quad (7)$$

where $e(t) = y_d(t) - y(t)$ and $p^*(t)$ is the open loop feedforward power profile.

PDE Model-Based Control Design Reported literature on PDE model-based feedback control strategies for laser power control is limited. These approaches are based on modeling the SLM process approximately as a 1D advection-diffusion-reaction equation (a partial differential equation) of the form

$$U_t(x, t) = kU_{xx}(x, t) + v(t)U_x(x, t) - \alpha U(x, t) + p(t)\Pi(x). \quad (8)$$

where $p(\cdot)$ is the power profile, $v(\cdot)$ is the laser velocity, k and α are based on material properties, $\Pi(\cdot)$ is the laser spot shape, and U is the distributed temperature *field* (scaled and shifted by the ambient temperature T_∞). The laser power p and v can both be potentially used as control inputs. The subscripts t and x denote partial derivatives with respect to time and spatial coordinates, respectively. The control objective is to regulate the cooling rate profile at a given location, given by $C_r(\delta) = -\frac{\partial U(x, t)}{\partial t}|_{x=\delta}$. There has been, as yet, no reported experimental demonstration of open- or closed-loop control of cooling rate.

Directed Energy Deposition: Laser Metal Deposition (LMD)

Laser metal deposition (LMD) is an AM process that uses a laser beam to form a melt pool on

a metallic substrate into which metal powder is injected using a gas stream. The absorbed metal powder produces a deposit on the surface, and thus a part is created by layer-upon-layer addition of metal. For these processes, the laser power and velocity are the two controllable variables. Measured variables during the process are the height and temperature field (or melt pool size). The control objective is typically to maintain a desired layer thickness (height) and a specified melt pool size.

Melt Pool Geometry Modeling and Control

Within a single pass of the laser, i.e., a single cladding layer, the dynamics of molten-pool geometry and temperature with respect to process input parameters such as laser power and laser scan speed can be obtained from mass, momentum, and energy balance in the melt pool. This results in a nonlinear implicit ODE model of the form:

$$h^2(t)\dot{h}(t) + K_1 h^2(t)v(t) = K_2 \frac{p(t) - p_c}{T_m - T_\infty} \quad (9)$$

$$\begin{aligned} \frac{d}{dt} \left(h^3(t)e(t) \right) &= K_3 h^2(t)v(t) \\ &+ K_4 p(t) + K_5 T(t) \\ &+ K_6 T_\infty + K_7 T_m \end{aligned} \quad (10)$$

$$\begin{aligned} e(t) &= c_1 T(t) + c_2 T_m \\ &+ c_3 T_\infty, \end{aligned} \quad (11)$$

where K_i, c_i are parameters dependent on the material properties, thermal properties, and metal powder volumetric flow; $T(t)$ is the average temperature of the melt pool; T_m is the melting point of the metal powder; T_∞ is the ambient and metal substrate temperature; $h(t)$ is the clad height; $p(t)$ is the input laser power; while $v(t)$ is the laser velocity.

Based on the multi-input multi-output (MIMO) nonlinear model above, a feedback linearization control that uses laser power and

laser scan speed to simultaneously track melt pool height and temperature reference trajectories can be designed. First, (9) and (10) can be rewritten into

$$\begin{bmatrix} \dot{h}(t) \\ \dot{T}(t) \end{bmatrix} = \mathbf{f}(h(t), T(t)) + \mathbf{G}(h(t), T(t)) \cdot \begin{bmatrix} v(t) \\ \tilde{p}(t) \end{bmatrix} \quad (12)$$

where $\tilde{p} = p - p_c$. We refer the reader to Wang et al. (2016) for the detailed expressions for $\mathbf{f}$ and $\mathbf{G}$. Given measurement (or estimates) of the state, a feedback linearization control of the form

$$\begin{bmatrix} v(t) \\ \tilde{p}(t) \end{bmatrix} = \mathbf{G}^{-1} \cdot \left(-\mathbf{f} + \begin{bmatrix} \lambda_1(h_d(t) - h(t)) \\ \lambda_2(T_d(t) - T(t)) \end{bmatrix} \right) \quad (13)$$

can be used to track a reference trajectory for height and melt pool temperature $[h_d(t) \ T_d(t)]^T$.

Modeling Layer-to-Layer Height Evolution

The model above focuses on dynamics within a single layer and thus ignores layer-to-layer height evolution and the effect of stand-off height on the material catchment efficiency η , which is necessary to describe phenomena observed in LMD such as layer rippling. To capture such phenomena, a 2D model that incorporates both in-layer and layer-to-layer dynamics is necessary. In Sammons et al. (2018), a 2D model of the following form is proposed and validated:

$$\begin{aligned} h(x, k) = & f_\mu(d_s(x, k) - h(x, k-1)) \lambda(x, j) \\ & * f_s(x) + h(x, k-1) * f_r(x) \end{aligned} \quad (14)$$

where k is the layer number; h is the height profile; x is the spatial coordinate; f_μ , f_s , and f_r are functions that model powder catchment, melt dynamics, and remelt dynamics, respectively; and $*$ is the convolution operation in the spatial dimension. The two controlled inputs are

λ , the powder flow rate, and d_s , the standoff height. Given this model, feedback and feedforward control strategies for powder flow rate and stand-off height can be obtained by using design tools for the general class of 2D models, which are presented in Sammons et al. (2019).

Summary and Future Directions

Control of AM systems is still in its early stages, with significant opportunities for improvement. Given the ever-growing variety in the types of processes, feedback and feedforward control strategies will need to be developed to improve throughput and yield. There are several challenges that hinder the widespread adoption of AM control strategies, particularly for closed-loop control. These include:

- *Sensing* is particularly challenging for AM systems. Although there are several modalities for process monitoring, these are not always well-suited for real-time feedback control. Further, relating physical process variables (e.g., temperature fields) to the sensed variables (e.g., IR camera images) is often difficult. From a control standpoint, this translates to designing suitable estimation algorithms for AM processes, which remains an open question.
- Suitable *control-oriented modeling* strategies that exploit both physical models and input-output data will be critical for feedback control. Current strategies are either black box or purely physics-based, which limits their applicability.
- *Control design strategies for distributed parameter systems* (such as PDEs) will need to be developed for both polymer and metal AM processes. Current approaches for AM control rely on discretization of these models, which leads to poor guarantees of performance. Further, system identification and online adaptation of such models on the fly remain an unsolved problem.

Cross-References

- [Iterative Learning Control](#)
- [Model Predictive Control in Practice](#)

Bibliography

- ASTM International (2012) Standard terminology for additive manufacturing technologies. Technical Report F2792-12a
- Cohen DL, Lipson H (2010) Geometric feedback control of discrete-deposition SFF systems. *Rapid Prototyp J* 16(5):377–393
- Craeghs T, Bechmann F, Berumen S, Kruth J-P (2010) Feedback control of layerwise laser melting using optical sensors. *Phys Procedia* 5:505–514; Laser assisted net shape engineering 6, Proceedings of the LANE 2010, Part 2. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1875389210005043>
- Doumanidis C, Skordeli E (2000) Distributed-parameter modeling for geometry control of manufacturing processes with material deposition. *J Dyn Syst Meas Control* 122(1):71–77
- Guo Y, Peters J, Oomen T, Mishra S (2018) Control-oriented models for ink-jet 3D printing. *Mechatronics* 56:211–219. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957415818300618>
- Hoelzle DJ, Barton KL (2016) On spatial iterative learning control via 2-D convolution: stability analysis and computational efficiency. *IEEE Trans Control Syst Technol* 24(4):1504–1512
- Kwon KS, Kim W (2007) A waveform design method for high-speed inkjet printing based on self-sensing measurement. *Sensors Actuators A Phys* 140(1):75–83
- Lu L, Zheng J, Mishra S (2015) A layer-to-layer model and feedback control of ink-jet 3-D printing. *IEEE/ASME Trans Mechatron* 20(3):1056–1068
- Rawlings JB, Mayne DQ (2012) *Model predictive control: theory and design*. Nob Hill Publishing, Madison
- Sammons PM, Bristow DA, Landers RG (2018) Two-dimensional modeling and system identification of the laser metal deposition process. *J Dyn Syst Meas Control* 141(2):021012 (10 pages)
- Sammons PM, Gegel ML, Bristow DA, Landers RG (2019) Repetitive process control of additive manufacturing with application to laser metal deposition. *IEEE Trans Control Syst Technol* 27(2):566–575
- Wang Q, Li J, Gouge PMM, Nassar AR, Reutzel E (2016) Physics-based multivariable modeling and feedback linearization control of melt-pool geometry and temperature in directed energy deposition. *J Manuf Sci Eng* 139(2):021013 (12 pages)
- Wassink MG, Bosch N, Bosgra O, Koekebakker S (2005) Enabling higher jet frequencies for an inkjet printhead using iterative learning control. In: 2005 IEEE conference on control applications (CCA). IEEE, pp 791–796

Control of Anemia in Hemodialysis Patients

Sabrina Rogg¹ and Peter Kotanko^{2,3}

¹Fresenius Medical Care Deutschland GmbH, Bad Homburg, Germany

²Renal Research Institute, New York, NY, USA

³Icahn School of Medicine at Mount Sinai, New York, NY, USA

Abstract

Since the introduction of erythropoiesis-stimulating agents (ESAs) in the late 1980s, anemia in hemodialysis (HD) patients has been managed using standard protocol-based approaches. Aiming for personalized therapy, the development of model-based feedback control algorithms for ESA dosing has become an active research area. Physiology-based models and artificial neural networks have been applied. Both approaches have shown superiority over traditional dosing schemes. In this chapter, we provide control-relevant background information, present the state-of-the-art control algorithms, and outline corresponding clinical study results.

Keywords

Model predictive control · Physiology-based models · Artificial neural networks · Erythropoiesis · Clinical trials

Introduction

In most countries worldwide, in-center hemodialysis (HD) is the predominant form of kidney replacement therapy for patients with end-stage renal disease (ESRD) (USRDS 2019). During HD, a patient's blood is filtered through an external circuit to remove waste, toxins, and excess fluids. Typically, in-center HD patients undergo treatment three times a week for about 3–5 h each. A common complication in patients with ESRD is anemia which is defined as a decreased number of circulating

red blood cells (RBCs) or decreased level of hemoglobin (Hgb). Anemia leads to reduced oxygen transport in the blood, is associated with poor quality of life, and represents an independent risk factor for cardiac disease, hospitalization, and mortality. The main driver of anemia in these patients is insufficient endogenous erythropoietin (EPO) production together with shortened RBC survival and iron deficiency. Exogenous replacement of EPO with erythropoiesis-stimulating agents (ESAs) in combination with iron supplementation has become the primary treatment. As dialysis lines provide an easy access to a patient's circulation, intravenous ESA and iron administrations during HD sessions are standard. Note that potential iron deficiency is usually assessed before initiating ESA treatment as the latter can only be efficient under sufficient iron availability (Fishbane and Spinowitz 2018; KDIGO 2012). Research in the field of personalized anemia management has so far mainly focused on the individualized dosing of ESAs, whereas iron is prescribed following computer-implemented or written protocols. Therefore, the focus of this chapter is on personalized ESA dosing.

The introduction of ESAs in the late 1980s has revolutionized the management of anemia in patients suffering from ESRD. Prior to their market entrance, RBC transfusions were the only treatment option. The currently available ESAs differ significantly in their serum half-life. First-generation drugs, epoetin alfa, are short-acting drugs as their half-life is only 4–10 h, requiring frequent dosing up to three times per week (i.e., each dialysis session). Second-generation ESAs, darbepoetin alfa and methoxy polyethylene glycol-epoetin beta, have a longer half-life of around 24–26 and 120–130 h, respectively. The benefit of the long-acting drugs is a reduction of dosing frequency from per treatment to weekly, biweekly, or even monthly (Fishbane and Spinowitz 2018; Kalantar-Zadeh 2017; Rosati et al. 2018). According to the most recent USRDS report (USRDS 2019), US HD patients were prescribed the following in 2016: 34.4% epoetin alfa, 17.9% darbepoetin alfa, and 24.4% methoxy polyethylene glycol-epoetin beta.

Anemia is assessed by a patient's Hgb level. Conventionally, Hgb concentrations in HD patients are measured between once a week and once a month from pre-dialysis blood samples. Several authors pointed at the potential of (optical) sensors like the Crit-Line® monitor (Fresenius Medical Care, Concord, CA) to provide noninvasive Hgb measurements for every HD treatment (even intra-dialysis measurements) (Chesterton et al. 2005; Thakuria et al. 2011). But until now, these additional measurements have not found their way into anemia management in clinical practice. Solely, Fuertinger et al. (2018a) used such noninvasive Hgb measurements to adapt a physiology-based mathematical model of erythropoiesis to individual patients for prediction of Hgb levels.

During the last 30 years, guidelines for Hgb targets in HD patients have been updated several times (Kalantar-Zadeh 2017; Wish 2018). Consensus exists with respect to only partially correcting anemia. It is not clear to what extent a full correction can further improve quality of life (QoL), whereas an increased risk of death and serious cardiovascular events has been demonstrated (Besarab et al. 1998; Drüeke et al. 2006; Singh et al. 2006; Pfeffer et al. 2009). At present, the recommended Hgb target range is 10–12 g/dl (Locatelli et al. 2013; Mactier et al. 2011). The most recent KDIGO anemia guidelines recommend to exceed the limit of 11.5 g/dl only for patients with further enhancement in QoL and who are prepared to take the associated risks (KDIGO 2012). This recommendation would culminate in an individualized risk versus benefit analysis for each and every patient (Drüeke et al. 2006; Fishbane and Spinowitz 2018). Koulouridis et al. (2013) have shown an association of high ESA doses with all-cause mortality and cardiovascular complications independent of Hgb. The FDA recommends using the lowest ESA dose sufficient to reduce the need for RBC transfusions. Although most patients respond adequately to ESAs, about 10% of them present ESA resistance or hyperresponsiveness (Johnson et al. 2007). These patients do not achieve the desired Hgb concentration despite large ESA

doses or require increasingly higher ESA doses to maintain a certain Hgb concentration. Hence, these patients need to be treated with caution, avoiding ineffective high ESA doses.

To summarize, the objective of anemia management in HD patients is to achieve the narrow Hgb target range while keeping ESA doses low. In addition, the individualized setting of Hgb targets should be possible. To date, there exists no generally accepted dosing algorithm. In the following, the current standard of care in ESA dosing is outlined before the two main streams in research on feedback control-based ESA dosing are presented.

Standard of Care for ESA Dosing

Most dialysis facilities use their own ESA dosing protocols which are built upon the package insert and follow human expert-based rules for initiation and dose adjustment. Usually, the last Hgb measurement and the rate of Hgb change are used to alter ESA dose recommendations between once a week and once a month. Eventually, the patient's iron status and treatment are factored in. Examples of such protocols can be found, for example, in Brier et al. (2018) and McCarthy et al. (2014). By now, these systems are usually computer-implemented which led to a certain standardization of care. So generated ESA doses are not automatically prescribed but assessed first by an anemia manager or physician who accepts or overwrites the recommendation.

The weakness of these protocols is that they are so-called one-size-fits-all and neither account for intraindividual nor interindividual variability. According to the Dialysis Outcomes and Practice Patterns Study, in June 2019, only 63% of US HD patients had a Hgb within 10–11.9g/dl (DOPPS 2019). Moreover, it has been shown that these protocols themselves contribute or even cause Hgb variability (Berns et al. 2003; Chait et al. 2013; Fishbane and Berns 2005; Rogers et al. 2018) and cycling (Lacson Jr et al. 2003), phenomena that are independently associated with all-cause mortality (Zhao et al. 2019). A virtual

anemia trial strategy to compare different dosing protocols *in silico* was presented and realized for methoxy polyethylene glycol-epoetin beta by Fuerterer et al. (2018b).

In the following, we refer to such standard treatment protocols as STPs.

Personalized ESA Dosing Through Closed-Loop Control

Different controller schemes have been developed by various groups over the last years (Barbieri 2016; Brier et al. 2010; Gaweda et al. 2014; Lines et al. 2011; Lobo et al. 2020; McCarthy et al. 2014; Uehlinger et al. 1992). This section covers approaches that have already been tested in clinical trials or even been implemented in clinical routine. As for STPs, generated ESA doses are presented to an anemia manager or physician as suggestion for prescription, turning any of the developed tools into a decision support system.

State-of-the-art feedback control algorithms for ESA dosing are model predictive controllers (MPCs) consisting of two main components: a model to predict a patient's Hgb response to ESA administration and a dose optimizer that determines, using the predictive model, optimal next ESA doses on a finite horizon into the future (called the prediction horizon). New ESA dose recommendations are triggered by Hgb measurements. The measurement frequency differs between once a week and once a month. The dosing frequency also varies between 3 times per week, weekly, biweekly, and monthly for different types of ESA. Depending on these schedules, some of the algorithms are m -step MPC schemes. The developed MPC algorithms can be categorized into two groups based on the predictive model they use: physiology-based models and artificial neural networks (ANN).

MPC Algorithms Based on Physiological Models

The physiological process that determines a patient's Hgb response to ESA administration is erythropoiesis which is the development from

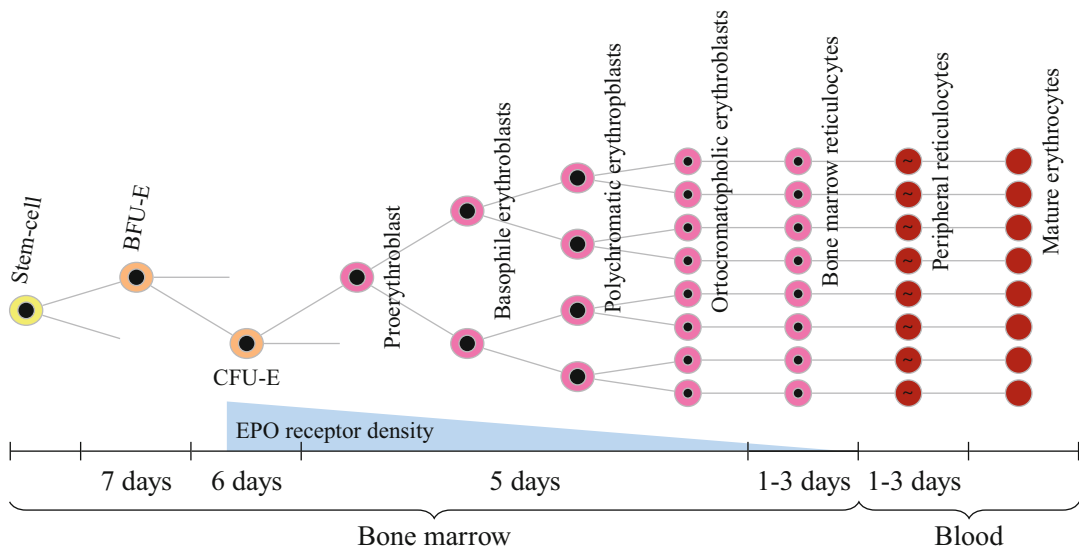
multipotent hematopoietic stem cells to mature RBCs through a series of proliferations and differentiations. A schematic of erythropoiesis is given in Fig. 1. Erythropoiesis occurs mostly in the bone marrow and ends in the blood stream. Starting from hematopoietic stem cells, erythroid cells mature to BFU-Es (burst-forming unit-erythroids) which develop into CFU-Es (colony-forming unit erythroids), pass different stages of erythroblasts, and become reticulocytes. Those are released from the bone into the blood stream and mature into erythrocytes within 1–3 days. The schematic shows the approximate time that cells stay in the respective class. It becomes obvious that an estimation on the RBC response to ESA administration is difficult as the cells stay in the bone marrow for up to three weeks. The shortened RBC lifespan in HD patients was measured by Ma et al. (2017) to be 73.2 ± 17.8 days (range 37.7–115.8 days).

Initially, simplified predictive modeling approaches for ESA dosing were investigated. The very first technique was presented by Uehlinger et al. in 1992. The software package NONMEM was used to estimate the population's RBC lifespan and sensitivity to the administered ESA. Easy calculations were provided to

determine the initial dose for a single patient using the population averages and to estimate the patient's individual values over time.

Using a modification of the model proposed by Uehlinger et al. (1992), Gaweda, Brier, and others developed a multiple MPC algorithm (Gaweda et al. 2014, 2018) that is meanwhile licensed by Dosis Inc. as Strategic Anemia Advisor (SAA). The two corresponding clinical studies are listed in Table 1. The algorithm uses five preset dose-response categories (hyporesponder to extreme hyperresponder) with a representative model for each class. On a monthly basis, optimal ESA doses for the category-specific models are determined. The suggested dose is obtained as the weighted average of these five doses. The weights are calculated using a fuzzy set approach (fuzzy membership functions).

An MPC algorithm that uses an extensive physiology-based model is the Mayo Clinic Anemia Management System (MCAMS) (McCarthy et al. 2014; Rogers et al. 2018). Erythropoiesis is described by means of a two-compartment model with nonlinear first-order differential equations. A conceptual overview of the model is shown in Rogers et al. (2018), but the explicit system



Control of Anemia in Hemodialysis Patients, Fig. 1 Schematic of different cell stages during erythropoiesis; compare Fig. 1 in Fuerstinger et al. (2013)

Control of Anemia in Hemodialysis Patients, Table 1 Summary of clinical trial results related to licensed MPC-based ESA dosing algorithms. STP refers to the respective standard treatment protocol

Study	Year	ESA	No. of subjects	Study design	Main findings	Licensed as
Gaweda et al. (2014)	2014	Epoetin alfa	62	RCT, 12 mos. single-center	Hgb values in target: 72.5% vs. 61.9% (STP) Median ESA dose per patient: no difference	SAA ^b Dosis Inc.
Gaweda et al. (2018) ^a	2018	Epoetin alfa	56	STP (12 mos.) SAA (12 mos.) single-center	Hgb values in target: 69.3% vs. 62.7% (STP) Mean ESA dose per week: −50%	
McCarthy et al. (2014)	2014	Darbepoetin alfa	300–342	STP (2007) transition (01/08–06/09) AMS (till 12/2010) 8 clinics	Hgb values in target: 83% vs. 69% (STP) ESA dose per patient per month: −40%	PhySoft AMS TM ^c
Barbieri et al. (2016b)	2016	Darbepoetin alfa	752	STP (1 year) ACM (1 year) 3 clinics	Hgb values in target: 76.6% vs. 70.6% (STP) Median ESA dose: 0.46 vs. 0.63 (STP) in $\mu\text{g/kg/month}$	ACM ^d integrated in EuCliD [®]
Bucalo et al. (2018)	2018	Darbepoetin alfa	213–219	ACM (6 mos.) STP (6 mos.) ACM (6 mos.) 2 clinics	Hgb values in target: 78.2% – 72.7% – 80.9% Median ESA dose: 20 – 30 – 20 in $\mu\text{g/month}$	

^aThis study also reports 42-months follow-up results for 233 patients

^bStrategic Anemia Advisor

^cPhysician Software Systems Anemia Management SystemTM

^dAnemia Control Model

equations have not been published. The dosing scheme has initially been tested in an observational study encompassing a total of 300–342 patients (see Table 1). The following five model parameters are weekly updated (if necessary) on a per patient level using a modified Monte Carlo procedure: mean daily BFU-E production rate, reference probability of CFU-E survival, effect of fractional EPO receptor binding on CFU-E survival, reticulocyte survival, and RBC lifespan (Rogers et al. 2018). Mayo Clinic licensed its algorithm to Physician Software Systems yielding FDA clearance in December 2013.

The dose optimizer applied in the study by McCarthy et al. (2014) was a simple trial-and-error strategy. So far, only in silico results for an MPC scheme that uses an individualized physiology-based model and an actual optimization algorithm for dose calculation are published (Rogg et al. 2019). The MPC scheme presented in Rogg et al. (2019) is based on a comprehensive age-structured cell population model to describe

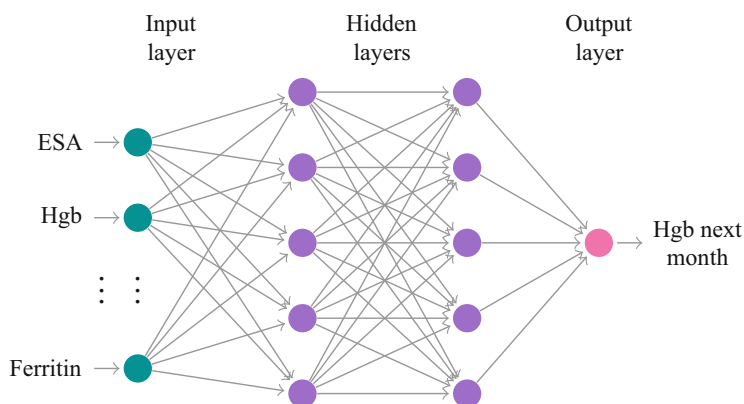
erythropoiesis by means of partial differential equations (Fuerterer et al. 2013, 2018a). The further developed MPC algorithm is currently being tested in a clinical trial for administration of methoxy polyethylene glycol-epoetin beta (Renal Research Institute 2020).

MPC Algorithms Based on Artificial Neural Networks

ANNs are a class of nonlinear machine learning methods inspired by neurons (nerve cells) in the human brain. ANNs map a set of input variables to a corresponding target output through a series of hidden layers. The ANNs used for prediction of Hgb values are mostly so-called feed-forward ones. In Fig. 2, such an ANN is exemplified. It consists of an input layer to bring the initial data into the system, an output layer that produces the output, and one or several hidden layers in between. Each layer consists of a number of artificial neurons which are connected

Control of Anemia in Hemodialysis Patients,

Fig. 2 Example of a feed-forward artificial neural network to predict the Hgb concentration in one month time using two hidden layers



by weighted links (synapses). Each link has a numerical weight associated with it. A neural network learns through iterative adjustment of these weights which is done in the training phase where the ANN is presented with the input-output pairs from a training data set. Once trained, an ANN is tested for accuracy on unseen data from a testing set. The output value of the ANNs is the predicted Hgb value in one month time.

The first clinical results of MPC algorithms that use an artificial neural network as the predictive model have been published by [Gaweda et al. in 2008](#) and by [Brier et al. in 2010](#). While these approaches used solely Hgb and ESA information as inputs to the ANN, Barbieri et al. have developed a model that makes use of a variety of patient information given for HD patients. As these patients are usually treated three times a week, a lot of data is captured even for an individual patient. As of today, the results from two clinical studies using the developed so-called Anemia Control Model (ACM) for dosing of darbepoetin alfa have been published ([Barbieri et al. 2016b](#); [Bucalo et al. 2018](#)) (see Table 1). Results on the predictive performance of the ANN can be found in [Barbieri et al. \(2015, 2016a\)](#). In [Barbieri et al. \(2016a\)](#), the ANN is described to consist of two hidden layers with ten neurons each. The ANN was trained on almost 170,000 and tested on more than 40,000 patient data sets. This data was extracted from Nephro-Care's proprietary clinical management system EuCliD® (European Clinical Database). Today, ACM is integrated into that system. The input

parameters of the ANN include sex, height, dry weight, previous Hgb change, ESA doses from last three months, iron doses, ferritin, and others. Each Hgb measurement, which happens usually once a month, triggers a dose recommendation. According to the authors, the dose optimizer simulates the Hgb response for different ESA doses and chooses a dose that moves the Hgb into the target interval while avoiding excessive Hgb excursion downward or upward.

Summary and Future Directions

Clinical studies on MPC-guided ESA dosing in HD patients have demonstrated its benefit over traditional dosing protocols. Both approaches, MPC based on a physiology-based model of erythropoiesis and based on an ANN to predict Hgb response, have shown a better achievement of Hgb targets, reduced Hgb variability, and reduced ESA consumption. These tools render the individualized setting of Hgb targets possible. So far, none of the different approaches have been tested against each other, and all methods have been developed and tailored to a prevalent Hgb measurement frequency (weekly or monthly). Until now, no clinical results on personalized dosing of methoxy polyethylene glycol-epoetin beta have been published.

In the United States, two web-based clinical decision support tools built upon MPCs that use a physiology-based model are commercially available: Dosis Inc.'s SAA and PhySoft AMS™. The

presented ACM based on an ANN is integrated in the clinical management system EuCliD[®] but not publicly available as stand-alone tool.

Aside from that, hypoxia-inducible factor prolyl hydroxylase inhibitors (HIF-PHIs), also known as HIF stabilizers, are a new class of drugs to treat anemia in HD patients which are about to enter the markets. Roxadustat, being developed by FibroGen, is already approved in China and Japan for treatment of HD patients and was accepted by the FDA for review in February 2020. HIF-PHIs are small molecules administered orally that could change the entire paradigm of anemia management in ESRD patients. HIF-PHIs increase RBC production by stimulating not only the production of endogenous erythropoietin but also by improving iron availability. So far, large-scale clinical trials that address the long-term safety of HIF-PHIs in ESRD patients are pending. Whether HIF-PHIs are associated with less risks than ESAs at comparable target Hgb levels is not yet clear but might again shift optimal Hgb levels in HD patients.

Cross-References

- [Automated Insulin Dosing for Type 1 Diabetes](#)
- [Control of Drug Delivery for Type 1 Diabetes Mellitus](#)
- [Model Predictive Control for Power Networks](#)
- [Model Predictive Control in Practice](#)

Recommended Reading

Fishbane and Spinowitz (2018) provide a comprehensive clinical overview on anemia in ESRD. The use of ESAs to treat anemia is reviewed by Hung et al. (2014). Brier et al. (2018) discuss the transition from paper algorithms to computer-implemented versions and summarize various optimal control-based ESA dosing approaches, including open-loop methods. A detailed review on the development of MCAMS is given by Rogers et al. (2018). Concrete mathematical equations of the physiology-based model of erythropoiesis and how to numerically

solve the optimal control problem for dose optimization are presented by Rogg et al. (2019). Barbieri et al. (2016a) outline the development of an ANN for prediction of Hgb concentrations and analyze its predictive accuracy in a retrospective study. Most relevant clinical studies on actual MPC-guided ESA dosing are listed in Table 1. Beyond that, Lobo et al. (2020) present in silico results for a recurrent neural network showing that the use of future iron dosing information does not enhance model performance. An overview of the different HIF stabilizers under development including conducted clinical studies is provided by Sanghani and Haase (2019).

Bibliography

- Barbieri C (2016) Anemia management in end-stage renal disease patients undergoing dialysis: a comprehensive approach through machine learning techniques and mathematical modeling. PhD Thesis, University of Valencia, Valencia, Spain
- Barbieri C, Mari F, Stopper A, Gatti E, Escandell-Montero P, Martínez-Martínez JM, Martín-Guerrero JD (2015) A new machine learning approach for predicting the response to anemia treatment in a large cohort of end stage renal disease patients undergoing dialysis. *Comput Biol Med* 61:56–61
- Barbieri C, Bolzoni E, Mari F, Cattinelli I, Bellocchio F, Martin JD, Amato C, Stopper A, Gatti E, Macdougall IC, Stuard S, Canaud B (2016a) Performance of a predictive model for long-term hemoglobin response to darbepoetin and iron administration in a large cohort of hemodialysis patients. *PLOS ONE* 11:1–18
- Barbieri C, Molina M, Ponce P, Tothova M, Cattinelli I, Titapiccolo JJ, Mari F, Amato C, Leipold F, Wehmeyer W et al (2016b) An international observational study suggests that artificial intelligence for clinical decision support optimizes anemia management in hemodialysis patients. *Kidney Int* 90(2):422–429
- Berns JS, Elzein H, Lynn RI, Fishbane S, Meisels IS, Deoreo PB (2003) Hemoglobin variability in epoetin-treated hemodialysis patients. *Kidney Int* 64(4): 1514–21
- Besarab A, Bolton WK, Browne JK, Egrie JC, Nissenson AR, Okamoto DM, Schwab SJ, Goodkin DA (1998) The effects of normal as compared with low hematocrit values in patients with cardiac disease who are receiving hemodialysis and epoetin. *N Engl J Med* 339(9):584–590
- Brier ME, Gaweda AE, Dailey A, Aronoff GR, Jacobs AA (2010) Randomized trial of model predictive control for improved anemia management. *Clin J Am Soc Nephrol* 5(5):814–820

- Brier ME, Gaweda AE, Aronoff GR (2018) Personalized anemia management and precision medicine in ESA and iron pharmacology in end-stage kidney disease. In: *Seminars in nephrology*, vol 38. Elsevier, pp 410–417
- Bucalo ML, Barbieri C, Roca S, Titapiccolo JI, Romero MSR, Ramos R, Albaladejo M, Manzano D, Mari F, Molina M (2018) The anaemia control model: does it help nephrologists in therapeutic decision-making in the management of anaemia? *Nefrología (English Edition)* 38(5):491–502
- Chait Y, Horowitz J, Nichols B, Shrestha RP, Hollof CV, Germain MJ (2013) Control-relevant erythropoiesis modeling in end-stage renal disease. *IEEE Trans Biomed Eng* 61(3):658–664
- Chesterton L, Lambie SH, Hulme LJ, Taal M, Fluck RJ, McIntyre CW (2005) Online measurement of haemoglobin concentration. *Nephrol Dial Transplant* 20(9):1951–1955
- DOPPS (2019) Arbor research collaborative for health: DOPPS practice monitor. <http://www.dopps.org/DPM/>. Accessed 5 May 2020
- Drieke TB, Locatelli F, Clyne N, Eckardt KU, Macdougall IC, Tsakiris D, Burger HU, Scherhag A (2006) Normalization of hemoglobin level in patients with chronic kidney disease and anemia. *N Engl J Med* 355(20):2071–2084
- Fishbane S, Berns JS (2005) Hemoglobin cycling in hemodialysis patients treated with recombinant human erythropoietin. *Kidney Int* 68(3):1337–1343
- Fishbane S, Spinowitz B (2018) Update on anemia in ESRD and earlier stages of CKD: core curriculum 2018. *Am J Kidney Dis* 71(3):423–435
- Fuertinger DH, Kappel F, Thijssen S, Levin NW, Kotanko P (2013) A model of erythropoiesis in adults with sufficient iron availability. *J Math Biol* 66(6):1209–1240
- Fuertinger DH, Kappel F, Zhang H, Thijssen S, Kotanko P (2018a) Prediction of hemoglobin levels in individual hemodialysis patients by means of a mathematical model of erythropoiesis. *PLOS ONE* 13(4):1–14
- Fuertinger DH, Topping A, Kappel F, Thijssen S, Kotanko P (2018b) The virtual anemia trial: an assessment of model-based in silico clinical trials of anemia treatment algorithms in patients with hemodialysis. *CPT: Pharmacometrics Syst Pharmacol* 7(4):219–227
- Gaweda A, Jacobs A, Aronoff G, Brier M (2008) Model predictive control of erythropoietin administration in the anemia of ESRD. *Am J Kidney Dis Off J Natl Kidney Found* 51:71–79
- Gaweda AE, Aronoff GR, Jacobs AA, Rai SN, Brier ME (2014) Individualized anemia management reduces hemoglobin variability in hemodialysis patients. *J Am Soc Nephrol* 25(1):159–166
- Gaweda AE, Jacobs AA, Aronoff GR, Brier ME (2018) Individualized anemia management in a dialysis facility—long-term utility as a single-center quality improvement experience. *Clin Nephrol* 90(4):276
- Hung SC, Lin YP, Tarng DC (2014) Erythropoiesis-stimulating agents in chronic kidney disease: what have we learned in 25 years? *J Formos Med Assoc* 113(1):3–10
- Johnson DW, Pollock CA, Macdougall IC (2007) Erythropoiesis-stimulating agent hyporesponsiveness. *Nephrology* 12(4):321–330
- Kalantar-Zadeh K (2017) History of erythropoiesis-stimulating agents, the development of biosimilars, and the future of anemia treatment in nephrology. *Am J Nephrol* 45(3):235–247
- KDIGO (2012) Clinical practice guideline for anemia in chronic kidney disease. *Kidney Int Suppl* 2: 279–335
- Koulouridis I, Alfayez M, Trikalinos TA, Balk EM, Jaber BL (2013) Dose of erythropoiesis-stimulating agents and adverse outcomes in CKD: a meta-regression analysis. *Am J Kidney Dis* 61(1):44–56
- Lacson E Jr, Ofsthun N, Lazarus JM (2003) Effect of variability in anemia management on hemoglobin outcomes in ESRD. *Am J Kidney Dis* 41(1):111–124
- Lines SW, Lindley EJ, Tattersall JE, Wright MJ (2011) A predictive algorithm for the management of anaemia in haemodialysis patients based on ESA pharmacodynamics: better results for less work. *Nephrol Dial Transplant* 27(6):2425–2429
- Lobo B, Abdel-Rahman E, Brown D, Dunn L, Bowman B (2020) A recurrent neural network approach to predicting hemoglobin trajectories in patients with end-stage renal disease. *Artif Intell Med* 104:101823
- Locatelli F, Bárány P, Covic A, De Francisco A, Del Vecchio L, Goldsmith D, Hörl W, London G, Vanholder R, Van Biesen W (2013) Kidney disease: improving global outcomes guidelines on anaemia management in chronic kidney disease: a European renal best practice position statement. *Nephrol Dial Transplant* 28(6):1346–1359
- Ma J, Dou Y, Zhang H, Thijssen S, Williams S, Kuntsevich V, Ouellet G, Wong MMY, Persic V, Kruse A, Rosales L, Wang Y, Levin NW, Kotanko P (2017) Correlation between inflammatory biomarkers and red blood cell life span in chronic hemodialysis patients. *Blood Purif* 43:200–205
- Mactier R, Davies S, Dudley C, Harden P, Jones C, Kanagasundaram S, Lewington A, Richardson D, Taal M, Andrews P, Baker R, Breen C, Duncan N, Farrington K, Fluck R, Geddes C, Goldsmith D, Hoenich N, Holt S, Jardine A, Jenkins S, Kumwenda M, Lindley E, McGregor M, Mikhail A, Sharples E, Shrestha B, Shrivastava R, Stedden S, Warwick G, Wilkie M, Woodrow G, Wright M (2011) Summary of the 5th edition of the renal association clinical practice guidelines (2009–2012). *Nephron Clin Pract* 118(Suppl 1):27–70
- McCarthy JT, Hocum CL, Albright RC, Rogers J, Gallagher EJ, Steensma DP, Gudgell SF, Bergstralh EJ, Dillon JC, Hickson LJ et al (2014) Biomedical system dynamics to improve anemia control with darbepoetin alfa in long-term hemodialysis patients. In: *Mayo clinic proceedings*, vol 89. Elsevier, pp 87–94
- Pfeffer MA, Burdmann EA, Chen CY, Cooper ME, De Zeeuw D, Eckardt KU, Feyzi JM, Ivanovich P, Kewalramani R, Levey AS et al (2009) A trial of darbepoetin alfa in type 2 diabetes and chronic kidney disease. *N Engl J Med* 361(21):2019–2032

- Renal Research Institute (2020) Assessment of an anemia model predictive controller for anemia management in hemodialysis patients. ClinicalTrials.gov. NCT04360902. <https://ClinicalTrials.gov/show/NCT04360902>. Accessed 15 May 2020
- Rogers J, Gallaher EJ, Dingli D (2018) Personalized esa doses for anemia management in hemodialysis patients with end-stage renal disease. *Syst Dyn Rev* 34(1–2):121–153
- Rogg S, Fuerstinger DH, Volkwein S, Kappel F, Kotanko P (2019) Optimal EPO dosing in hemodialysis patients using a non-linear model predictive control approach. *J Math Biol* 79(6–7):2281–2313
- Rosati A, Ravaglia F, Panichi V (2018) Improving erythropoiesis stimulating agent hyporesponsiveness in hemodialysis patients: the role of hepcidin and hemodiafiltration online. *Blood Purif* 45(1–3):139–146
- Sanghani NS, Haase VH (2019) Hypoxia-inducible factor activators in renal anemia: current clinical experience. *Adv Chronic Kidney Dis* 26(4):253–266
- Singh AK, Szczech L, Tang KL, Barnhart H, Sapp S, Wolfson M, Reddan D (2006) Correction of anemia with epoetin alfa in chronic kidney disease. *N Engl J Med* 355(20):2085–2098
- Thakuria M, Ofsthun NJ, Mullon C, Diaz-Buxo JA (2011) Anemia management in patients receiving chronic hemodialysis. In: *Seminars in dialysis*, vol 24. Wiley Online Library, pp 597–602
- Uehlinger DE, Gotch FA, Sheiner LB (1992) A pharmacodynamic model of erythropoietin therapy for uremic anemia. *Clin Pharmacol Ther* 51(1):76–89
- USRDS (2019) United States Renal Data System annual data report: Epidemiology of kidney disease in the United States. National Institutes of Health, National Institute of Diabetes and Digestive and Kidney Diseases, Bethesda
- Wish JB (2018) Perspective: will we ever know the optimal Hgb level in ESRD? *J Am Soc Nephrol* 29(10):2454–2457
- Zhao L, Hu C, Cheng J, Zhang P, Jiang H, Chen J (2019) Haemoglobin variability and all-cause mortality in haemodialysis patients: a systematic review and meta-analysis. *Nephrology* 24(12):1265–1272

Control of Biotechnological Processes

Rudibert King

Technische Universität Berlin, Berlin, Germany

Abstract

Closed-loop control can significantly improve the performance of bioprocesses, e.g., by an

increase of the production rate and yield of a target molecule or by guaranteeing reproducibility of the production together with low by-product formation. In contrast to control of chemical reaction systems, the biological reactions take place inside cells which constitute highly regulated, i.e., internally controlled systems by themselves. As a result, through evolution, the same cell can and will mimic a system of first order in some situations and a high-dimensional, highly nonlinear system in others. This makes supervision, control, and optimization of biosystems very demanding, and a significant amount of control design effort has to be put in the mathematical modeling in the first place.

Keywords

Bioprocess control · Control of uncertain systems · Optimal control · Parameter identification · State estimation · Structure identification · Structured models · Machine learning

Introduction

Biotechnology offers solutions to a broad spectrum of tasks faced today, e.g., in health care, remediation of environmental pollution, new sources for energy supplies, sustainable food production, and the supply of bulk chemicals. To explain the needs for control of bioprocesses, especially for the production of high-value and/or large-volume compounds, it is instructive to have a look on the development of a new process. If a potential strain is found or genetically engineered, the biologist will determine favorable environmental factors for the growth of the cells and the production of the target molecules. These factors typically comprise the kind of nutrients, precursors and so-called trace elements, and the levels of “physical” quantities such as temperature, pH, dissolved oxygen, etc. Moreover, favorable concentration regions for the different chemical compounds are specified. Whereas for the physical variables often fixed

setpoints are suggested, which, at least in smaller scale reactors, can be easily maintained by independent, classically designed controllers, information about the best nutrient supply is incomplete from a control engineering point of view. A dynamic evolution of the nutrient supply, however, offers substantial room for production improvements by control, but is most often not revealed in the biological laboratory. Even if constant concentration levels of the individual nutrients are suggested, the exponential growth of the cells and the highly nonlinear dependencies necessitate constantly changing nutrient feeds into the fermenter, posing challenges for control.

Irrespective whether bacteria, yeasts, fungi, or animal cells are used for production, these cells will consist of thousands of different compounds which react with each other in hundreds or more reactions. All reactions are tightly regulated on a molecular and genetic basis; see “Biosystems and Control”. For so-called unlimited growth conditions, all cellular compartments will be built up with the same specific growth rate, meaning that the cellular composition will not change over time. In a mathematical model describing growth and production, now only one state variable will be needed to describe the biotic phase. This will give rise to unstructured models; see below. Whenever a cell enters a limitation or the environmental conditions are changed significantly, which is often needed for production, the cell will start to dynamically reorganize its internal reaction pathways. Model-based approaches of supervision and control based on unstructured models are now bound to fail. More biotic state variables are needed in the model. However, it is not clear which and how many. As a result, modeling the behavior of the cells when the environment changes drastically is crucial for control of biotechnological processes.

For supervision and control, model-based estimates of the state of the cell and of the environment are a key factor as online measurements of the internal processes inside the cell and of part of the nutrient concentrations are usually impossible. This situation is alleviated though through progress in spectroscopic methods (see Voss et al. 2017), but care has to be taken as very often linear

methods, such as Partial Least Squares (PLS), are applied that might lead to erroneous results. An example will be given below by means of Fig. 1.

Finally, as models used for process control have to be limited in size and thus only give an approximative description, robustness of the methods has to be addressed. Due to these inherent problems, an increasing number of data-driven approaches, e.g., from machine learning, have been proposed. However, data-driven methods suffer from deficiencies with respect to predictions and transfer of control solutions when it comes to the scale-up of fermenters. Hybrid approaches may offer an attractive alternative; see below.

Mathematical Models

In bioprocesses, many up- and downstream unit operations are involved besides the fermentation. As these pose no typical bio-related challenges, we will concentrate here on the cultivation of the organisms only. This is mostly performed in aqueous solutions in special bioreactors/fermentors through which in many cases air is sparged for a supply with oxygen. Disregarding wastewater treatment plants, most cultivations are still performed in a fed-batch mode, meaning that a small amount of cells and part of the nutrients are put into the reactor initially. Then, more nutrients and correcting fluids, e.g., for pH or antifoam control, are added with variable rates leading to an unsteady behavior (see Mears et al. 2017). The system to be modeled consists of the gaseous, the liquid, and the biotic phase inside the reactor. For the former ones, balance equations can be formulated readily. The biotic phase can be modeled in a structured or unstructured way. Moreover, as not all cells behave similarly, this may give rise to a segregated model formulation that is omitted here for brevity.

Unstructured Models

If the biotic phase is represented by just one state variable, m_X , a typical example of a simple unstructured model of the liquid phase derived from balance equations would be

$$\dot{m}_X = \mu_X m_X$$

$$\dot{m}_P = \mu_P m_X$$

$$\dot{m}_S = -a_1 \mu_X m_X - a_2 \mu_P m_X + c_{S,feed} u$$

$$\dot{m}_O = a_3(a_4 - c_O) - a_5 \mu_X m_X - a_6 \mu_P m_X$$

$$\dot{V} = u$$

with the masses m_i with $i = X, P, S, O$, for cells, product, limiting substrate or nutrient, and dissolved oxygen, respectively. The volume is given by V , and the specific growth rate and production rate μ_X and μ_P depend on concentrations $c_i = m_i/V$, e.g., of the substrate S or oxygen O . The rates are described by formal kinetics, such as

$$\mu_X = \frac{a_7 c_S c_O}{(c_S + a_8)(c_O + a_9)},$$

$$\mu_P = \frac{a_{10} c_S}{a_{11} c_S^2 + c_S + a_{12}}$$

The nutrient supply can be changed by the feed rate $u(t)$ as a control input, with inflow concentration $c_{S,feed}$. Very often, just one feed stream is considered in unstructured models. As all parameters a_i have to be identified from noisy and infrequently sampled data, a low-dimensional nonlinear uncertain model results. Implicitly it is often assumed that experiments are reproducible. However, biological variance and all uncertain steps in the so-called precultures add to the uncertainty of the measured data; cf. Fig. 1.

Whereas the balance equations follow from first principles-based modeling, the structure of the kinetics μ_X and μ_P is unknown. Empirical relations are postulated. Many different kinetic expressions can be used here; see Bastin and Dochain (2008) or a small selection shown in Table 1.

It has to be pointed out that, most often, neither c_X , c_P , nor c_S are measured online. As the measurement of c_O might be unreliable, the exhaust gas concentration of the gaseous phase is the main online measurement that can be used employing an additional balance equation for the gaseous phase. Infrequent at-line measurements,

though, are available for X , P , S , especially at the lab-scale during process development.

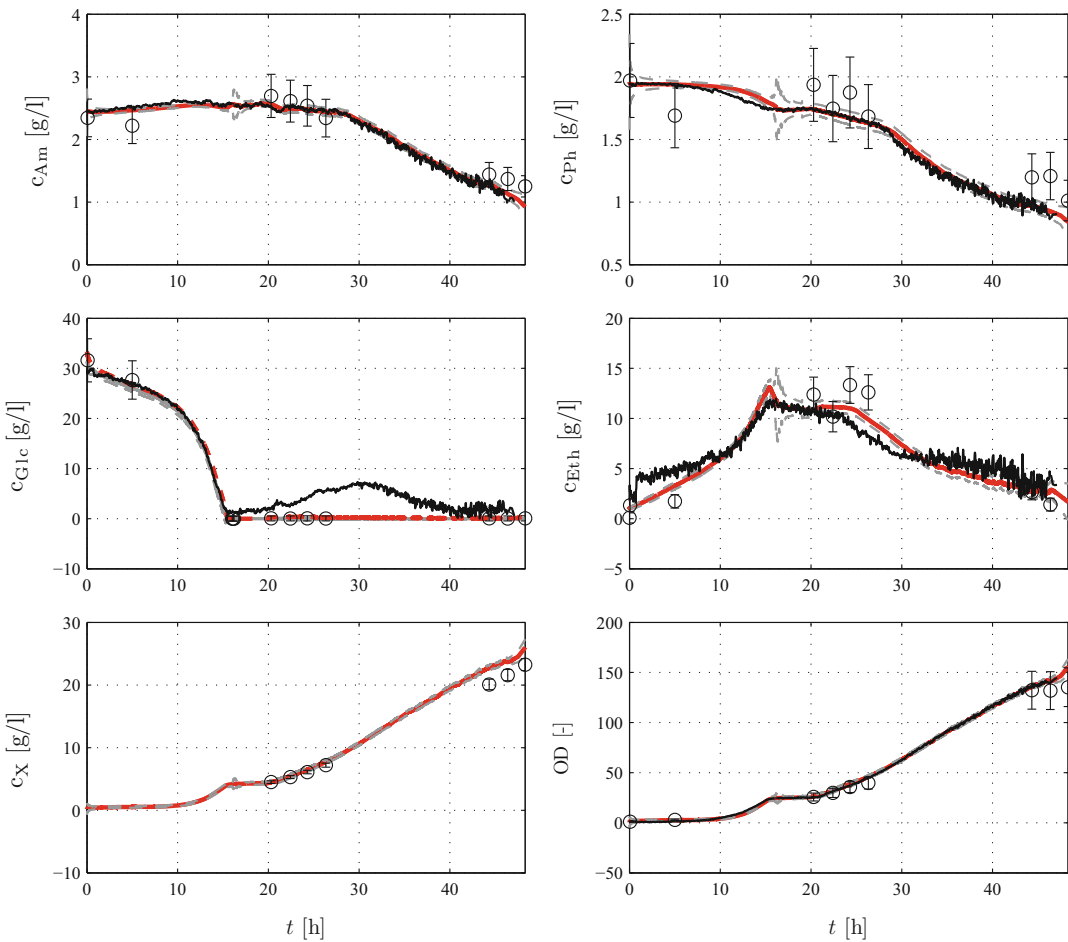
Structured and Hybrid Models

In structured models, the changing composition and reaction pathways of the cell is accounted for. As detailed information about the cell's complete metabolism including all regulations is missing for the majority if not all cells exploited in bioprocesses, an approximative description is used. Examples are models in which a part of the real metabolism is described on a mechanistic level, whereas the rest is lumped together into one or very few states (Goudar et al. 2006) or in so-called cybernetic models (Ramkrishna and Song 2012) or in compartment models (King 1997). As an example, all compartment models can be written down as

$$\dot{\mathbf{m}} = \mathbf{A}\boldsymbol{\mu}(\mathbf{c}) + \mathbf{f}_{in}(\mathbf{u}) + \mathbf{f}_{out}(\mathbf{u})$$

$$\dot{V} = \sum_i u_i$$

with vectors of streams into and out of the reaction mixture, $\mathbf{f}_{in}$ and $\mathbf{f}_{out}$, which depend on control inputs $\mathbf{u}$; a matrix of (stoichiometric) parameters, $\mathbf{A}$; a vector of reaction rates $\boldsymbol{\mu} = \boldsymbol{\mu}(\mathbf{c})$; and, finally, a vector $\mathbf{m}$ comprising the masses of substrates, products, and more than one biotic state. These biotic states can be motivated, for example, by physiological arguments, describing the total amounts of macromolecules in the cell, such as the main building blocks DNA, RNA, and proteins. In very simple compartment models, the cell is only divided up into what is called active and inactive biomass. Again, all coefficients in $\mathbf{A}$ and the structure and the coefficients of all entries in $\boldsymbol{\mu}(\mathbf{c})$ (see Table 1) are unknown and have to be identified based on experimental data. Issues of structural and practical identifiability are of major concern. For models of system biology, algebraic equations are added that describe the dependencies between individual fluxes; however, very often information concerning the regulated dynamics of the fluxes is unknown.



Control of Biotechnological Processes, Fig. 1 EKF prediction (solid red line) of a yeast cultivation based on misleading spectroscopic data (solid black line), 2σ uncertainty interval obtained by the EKF (dotted grey line)

and off-line data (circles with error bars). Nutrients are ammonia, glucose, and phosphate. Products are ethanol and an enzyme (not shown) (Krämer and King 2016)

In hybrid models, models based on balance equations are combined with data-driven approaches, e.g., by replacing unknown parts, such as the formal kinetics of Table 1, with the outcome of neural nets; cf. (Stosch et al. 2016).

Identification

Even if the growth medium initially “only” consists of some 10–20 different, chemically well-defined substances, from which only few are described in the model, this situation will

Control of Biotechnological Processes, Table 1 Multiplicative rates depending on several concentrations $c_1, \dots, c_k$ with possible kinetic terms

$\mu_i = a_{imax} \mu_{i1}(c_1) \cdot \mu_{i2}(c_2) \cdot \dots \cdot \mu_{ik}(c_k)$			
μ_{ij}	$\frac{c_j}{c_j + a_{ij}}$	$\frac{a_{ij}}{c_j + a_{ij}}$	$\frac{c_j}{c_j^2 + a_{ij}}$
	$\frac{c_j}{c_j + a_{ij}} e^{-c_j/a_{ij+1}}$	$\frac{c_j}{a_{ij}c_j^2 + c_j + a_{ij+1}}$	$\dots$

change over the cultivation time as the organisms release further compounds from which only few may be known. If, for economic reasons, complex raw materials are used, even the initial

composition is unknown. For structured models, intracellular substances have to be determined additionally. These are embedded in an even larger matrix of compounds making chemical analysis more difficult. Therefore, the basis for parameter and structure identification or training of data-driven approaches is uncertain and sparse.

As the expensive experiments and chemical analysis tasks are very time-consuming, methods of optimal experimental design should always be considered in biotechnology; see ► [“Experiment Design and Identification for Control”](#).

The models to be built up should possess some predictive capability for a limited range of environmental conditions. This rules out unstructured models for many practical situations. However, for process control, the models should still be of manageable complexity. Medium-sized structured models seem to be well suited for such a situation. The choice of biotic states in **m** and possible structures for the reaction rates μ_i , however, is hardly supported by biological or chemical evidence. As a result, a combined structure and parameter identification problem has to be solved. The choices of possible terms μ_{ij} in all μ_i give rise to a problem that exhibits a combinatorial explosion. Although approaches exist to support this modeling step (see Herold and King 2013 or Mangold et al. 2005), the modeler will finally have to settle with a compromise with respect to the accuracy of the model found versus the number of fully identified model candidates. As a result, all control methods applied should be robust in some sense.

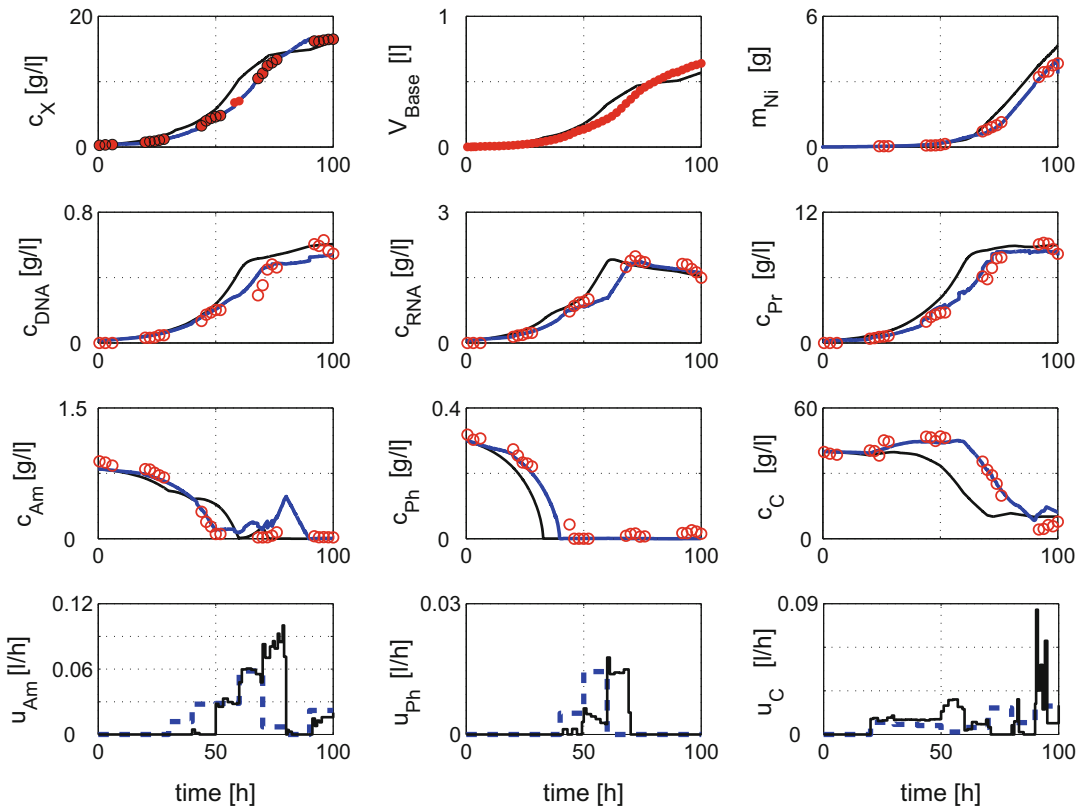
Soft Sensors

Despite advantages in the development of online measurements (see Mandenius and Titchener-Hooker 2013), systems for supervision and control of biotechnical processes often include model-based estimations schemes, such as extended Kalman filters (EKF); see ► [“Kalman Filters”](#) and Fig. 2. Concentration estimates are needed for unmeasured substances and for quantities which depend on these concentrations

like the growth rate of the cells. In real applications, formulations have to be used which account for delays in laboratory analysis of up to several hours and for situations in which results from the laboratory will not be available in the same sequence as the samples were taken. This can be tackled, e.g., with moving horizon estimation. An example from a real cultivation is shown in Fig. 1 where a linear spectroscopic method predicts a non-zero glucose concentration in the second half of the experiment that can be falsified by the outcome of an extended Kalman filter based on a process model. The estimates could be verified later by offline measurements shown as well.

Control

Besides the relatively simple control of physical parameters, such as temperature, pH, dissolved oxygen, and carbon dioxide concentration, only few biotic variables are typically controlled with respect to a setpoint; see Pörtner et al. (2017). The most prominent example is the growth rate of the biomass with the goal to reach a high cell concentration in the reactor as fast as possible. For non-growth-associated products, a high cell mass is desirable as well, as production is proportional to the amount of cells. If the nutrient supply is maintained above a certain level, unlimited growth behavior results, allowing the use of unstructured models for model-based control. An excess of nutrients has to be avoided, though, as some organisms, like baker's yeast, will initiate an overflow metabolism, with products that may be inhibitory in later stages of the cultivation. For some products, such as the antibiotic penicillin, the organism has to grow slowly to obtain a high production rate. For these so-called secondary metabolites, low but not vanishing concentrations for some limiting substrates are needed. If setpoints are given for these concentrations instead, this can pose a rather challenging control problem. As the organisms try to grow exponentially, the controller must be able to increase the feed exponentially as well. The difficulty mainly arises from the inaccurate and infrequent measurements



Control of Biotechnological Processes, Fig. 2 MPC control and state estimation of a cultivation with *S. tendae*. At-line measurement m_X (red filled circles), initially predicted evolution of states (black), online estimated evolution

(blue); off-line data analyzed after the experiment (open red circles). Off-line optimal feeding profiles u_i (blue broken line), MPC-calculated feeds (black, solid). Data obtained by T. Heine

that the soft sensors/controller has to work with and from the danger that an intermediate shortage or oversupply with nutrients may switch the metabolism to an undesired state of low productivity.

For control of biotechnical processes, many methods explained in this Encyclopedia including feedforward, feedback, model-based, optimal, adaptive, fuzzy, neural nets, etc., can be and have been used (cf. Rani and Rao 1999; Dochain 2008; Mears et al. 2017). As in other areas of application, (robust) model-predictive control schemes (MPC) (see ► [Model Predictive Control in Practice](#)) are applied with great success in biotechnology. One of the very first applications in which the production of an antibiotic by a mycelial organism was optimized in a 50 L scale fermenter exploiting a structured model,

an extended Kalman filter and an NMPC was already reported in the year 1993 (see King 1997; Waldruff et al. 1997). For the antibiotic production shown in Fig. 2, optimal feeding profiles u_i for the nutrients ammonia (AM), phosphate (PH), and glucose (C) were calculated before the experiment was performed (see Kawohl et al. 2007). The goal was a maximization of the final mass of the desired antibiotic nikkomycin (Ni). This resulted in the blue broken lines for the feeds u_i . However, due to disturbances and model inaccuracies, an MPC scheme had to significantly change the feeding profiles, to actually obtain this high amount of nikkomycin; see the feeding profiles given in black solid lines. This example shows that, especially in biotechnology, offline trajectory planning has to be complemented by closed-loop concepts.

Summary and Future Directions

Advanced process control including soft sensors can significantly improve biotechnical processes. Improved control in the sense of quality by design (QbD) reduces costly tests of the end product in pharmaceutical applications; cf. (Sommeregger et al. 2017). Using these techniques promotes quality and reproducibility of processes. These methods should, however, not only be exploited in the production scale. For new pharmaceutical products, the time to market is the decisive factor. Methods of (model-based) monitoring and control can help here to speed up process development. Since a few years, a clear trend can be seen in biotechnology to miniaturize and parallelize process development using multi-fermenter systems and robotic technologies. This trend gives rise to new challenges for modeling on the basis of huge data sets and for control in very small scales. At the same time, it is expected that a continued increase of information from bioinformatic tools will be available that has to be utilized for process control as well. Going to large-scale cultivations adds further spatial dimensions to the problem. Now, the assumption of a well-stirred, ideally mixed reactor does not longer hold. Substrate concentrations will be space-dependent. Cells will experience dynamically varying nutrient environments. Thus, mass transfer has to be accounted for, leading to partial differential equations as models for the process.

Cross-References

- [Control and Optimization of Batch Processes](#)
- [Deterministic Description of Biochemical Networks](#)
- [Extended Kalman Filters](#)
- [Experiment Design and Identification for Control](#)
- [Moving Horizon Estimation](#)
- [Nonlinear System Identification: An Overview of Common Approaches](#)

Bibliography

- Bastin G, Dochain D (2008) On-line estimation and adaptive control of bioreactors. Elsevier, Amsterdam
- Dochain D (2008) Bioprocess control. ISTE Ltd, London
- Goudar C, Biener R, Zhang C, Michaels J, Piret J, Konstantinov K (2006) Towards industrial application of quasi real-time metabolic flux analysis for mammalian cell culture. *Adv Biochem Eng Biotechnol* 101:99–118
- Herold S, King R (2013) Automatic identification of structured process models based on biological phenomena detected in (fed-)batch experiments. *Bioproc Biosyst Eng* 37:1289–1304
- Kawohl M, Heine T, King R (2007) Model-based estimation and optimal control of fed-batch fermentation processes for the production of antibiotics. *Chem Eng Process* 11:1223–1241
- King R (1997) A structured mathematical model for a class of organisms: Part 1–2. *J Biotechnol* 52:219–244
- Krämer D, King R (2016) On-line monitoring of substrates and biomass using near-infrared spectroscopy and model-based state estimation for enzyme production by *S. cerevisiae*. *IFAC-PapersOnLine* 49(7): 609–614
- Mandenius CF, Titchener-Hooker NJ (ed)(2013) Measurement, monitoring, modelling and control of bioprocesses. *Advances in biochemical engineering/biotechnology*, 132. Springer, Heidelberg
- Mangold M, Angeles-Palacios O, Ginkel M, Waschler R, Kinele A, Gilles ED (2005) Computer aided modeling of chemical and biological systems – methods, tools, and applications. *Ind Eng Chem Res* 44:2579–2591
- Mears L, Stocks SM, Sin G, Gernaey KV (2017) A review of control strategies for manipulating the feed rate in fed-batch fermentation processes. *J biotechnol* 245: 34–46
- Pörtner R, Barradas O P, Frahm B, Hass V C (2017) Advanced process and control strategies for bioreactors. In: Pandey A et al. (eds) *Current Developments in Biotechnology and Bioengineering*. Elsevier, Amsterdam, pp 463–493
- Ramkrishna D, Song H-S (2012) Dynamic models of metabolism: review of the cybernetic approach. *AIChE J* 58:986–997
- Rani KY, Rao VSR (1999) Control of fermenters. *Bioproc Eng* 31:21–39
- Sommeregger W, Sissolak B, Kandra K, von Stosch M, Mayer M, Striedner G (2017) Quality by control: towards model predictive control of mammalian cell culture bioprocesses. *Biotechnol J* 12:1600546
- von Stosch M, Hamelink J-M, Oliviera R (2016) Hybrid modeling as a QbD/PAT tool in process development: an industrial *E. coli* case study. *Bioprocess Biosyst Eng* 39:773–784
- Voss J-P, Mittelheuser NE, Lemke R, Luttmann R (2017) Advanced monitoring and control of pharmaceutical production processes with *Pichia pastoris* by using

Raman spectroscopy and multivariate calibration methods. *Eng Life Sci* 17(12):1281–1294

Waldraff W, King R, Gilles ED (1997) Optimal feeding strategies by adaptive mesh selection for fed-batch bioprocesses. *Bioprocess Eng* 17:221–227

Control of Circadian Rhythms and Related Processes

A. Agung Julius

Department of Electrical, Computer, and Systems Engineering, Center for Lighting Enabled Systems and Applications, Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract

Circadian rhythms are cyclical behaviors observed in many biological processes, which form a mechanism with which living beings can synchronize with the light and dark pattern of the terrestrial day. In humans, the suprachiasmatic nucleus (SCN) functions as the central circadian pacemaker. Misalignment between the circadian rhythms and the solar day, for example, in the case of jetlag or rotating shift work, leads to negative health consequences and lower productivity. The central circadian pacemaker is driven by light stimuli and coupled with other physiological processes. The sleep process, for example, is known to be modulated by the central circadian rhythm. In this chapter, we discuss the formulation of various control problems related to the regulation of circadian rhythms using light as the control input. We aim at shifting the phase of the circadian oscillation as quickly as possible. In some cases, we include constraints that involve sleep scheduling. In all cases, the problems are formulated as continuous-time optimal control problems.

Keywords

Circadian rhythms · Biological processes · Optimal control · Minimum-time control

Synonyms

Circadian Rhythms Entrainment

Introduction

The circadian rhythms in humans function as a master clock that regulates various biological processes such as sleep, metabolism, and neurobehavioral processes (Foster and Kreitzman 2017) (see illustration in Fig. 1). Disruption of the circadian rhythms is known to have negative impacts on health, ranging from fatigue in travelers with jetlag to an increased risk of cancer in rotating shift workers. Further, malalignment of the circadian phase and related neurobehavioral states, such as alertness, with the timing of critical tasks may lead to lower performance and higher risk of failure. The circadian rhythms are driven by internal pacemakers and externally affected by light stimuli, as well as activity, meals, body temperature, and hormones.

The sleep process in humans is very tightly connected to the circadian rhythm. The sleep drive, for example, is known to be modulated by the circadian rhythm (Achermann and Borbély 2003). Sleep is very important as it allows the body to recuperate and regenerate cells. It is also tied to neurocognitive performance; the lack of or mistiming of sleep have been empirically linked to degeneration of neurocognitive performance and disruption of circadian rhythm regulation.

Regulation of circadian rhythms, from the perspective of control systems, is typically expressed as an optimal control problem of a system with nonlinear dynamics. The control inputs into the system are typically light (Doyle III et al. 2006; Serkh and Forger 2014; Qiao et al. 2017) (which is a strong circadian synchronizer) and pharmaceuticals (Abel et al. 2019). For the remainder of this entry, we shall focus on light as the control input in circadian rhythms control.

Modeling

In the literature, there are mathematical models that capture the dynamics of the circadian rhythm and how light affects it. Generically, the models are ordinary differential equations of the form

$$\frac{dx}{dt} = F(x, u), \quad (1)$$

where x is the state of the circadian oscillator and u is the control input. Because light can only attain nonnegative intensity and its impact to the circadian oscillator has a saturation regime, u is typically constrained as

$$0 \leq u(t) \leq u_{\max}, \quad (2)$$

for all $t \geq 0$.

Circadian rhythms models come in various orders (i.e., levels of complexity). The highest order models, typically based on the biochemical processes in the circadian gene regulation (Leloup and Goldbeter 2003). Lower-order models are typically empirical, such as the third-order model by Jewett et al. (1999), which uses a van der Pol oscillator to represent the circadian

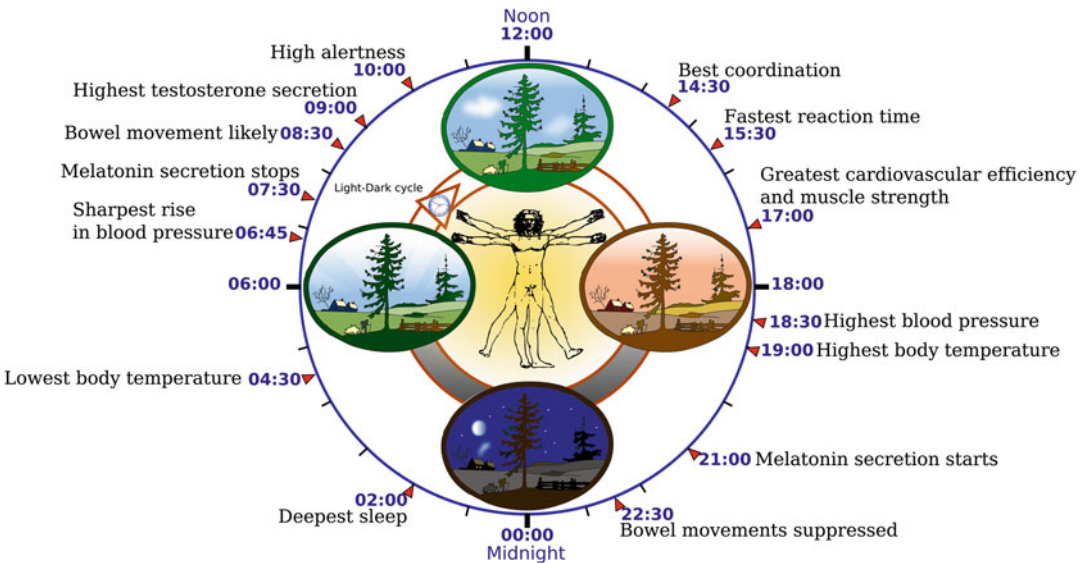
rhythm of the core body temperature. There are also first-order models as described below (Qiao et al. 2017).

First-order models that describe the circadian phase dynamics are of the form

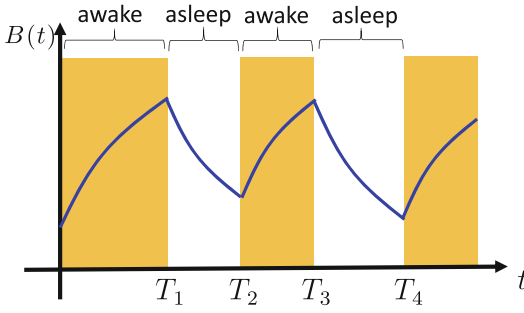
$$\frac{d\theta}{dt} = \omega_0 + f(\theta)u, \quad (3)$$

where the variable θ is the circadian phase, expressed in rad. The parameter ω_0 is the so-called free-running frequency, expressed in rad/h. Nominally its value is around $\frac{2\pi}{24}$ rad/h. The variable u is the light control input. The function $f(\theta)$ is called the *phase response function* or *phase response curve*, which maps the circadian light input to the dynamics of the circadian phase.

The dynamics of the sleep and neurobehavioral states are coupled to that of the circadian rhythm. Their relationship is described in a family of models called the two-process models. These models link the dynamics of sleep drive and alertness to the circadian phase in a hybrid or multimodal dynamical system:



Control of Circadian Rhythms and Related Processes, Fig. 1 The circadian component of various physiological and neurobehavioral (e.g., alertness) processes. (Image source: https://en.wikipedia.org/wiki/Circadian_rhythm)



Control of Circadian Rhythms and Related Processes, Fig. 2 Illustration of the sleepiness trajectory, which is modeled as a hybrid system. The subject goes to sleep at time $T_1, T_3, \dots$ and wakes up at $T_2, T_4, \dots$

$$\frac{ds}{dt} = \begin{cases} G_s(s), & \text{if the subject is asleep,} \\ G_a(s), & \text{if the subject is awake.} \end{cases} \quad (4)$$

Here s denotes the sleep and neurobehavioral states. The transitions between the sleep and awake modes are determined by sleepiness. Sleepiness is quantified as a variable $B(t)$ and modeled as a function of the circadian and sleep states,

$$B(t) = H(x(t), s(t)). \quad (5)$$

The transitions between sleep and wakefulness can be modeled in two ways:

1. Autonomous switching: the occurrence of the transitions purely depends on the states of the systems. For example, we can assume that the subject falls asleep when their sleepiness reaches a certain high level and wakes up spontaneously when their sleepiness hits a certain low level.
2. Scheduled switching: the transitions are externally scheduled, for example, when the subject follows a given sleep schedule.

The hybrid nature of the two process model is illustrated in Fig. 2. Note that the sleep state impacts the circadian state dynamics because the light stimuli cannot act on the subject while asleep.

Alertness, quantified as $A(t)$, is modeled as a function of the sleep and neurobehavioral states (Achermann and Borbely 2003),

$$A(t) = K(x(t), s(t)). \quad (6)$$

The factors that impact alertness in this model are sleep debt (whether the subject is lacking sleep), sleep inertia (whether the subject just recently gain wakefulness), and circadian phase (the subject's biological clock).

Optimal Control of Circadian Rhythms and Related Processes

Quickest Entrainment Problem

One of the main problems in optimal circadian rhythm regulation is the *quickest entrainment problem*, which can be formulated as follows.

Quickest Entrainment Problem (QEP) Given the dynamics of the circadian state in Eqs. (1) and (2), an initial condition $x(0) = x_0$, and a reference circadian state trajectory $x_{\text{ref}}(t)$, determine the input $u(t)$ that minimizes the entrainment time T . Here, the entrainment time T is defined as the earliest time such that $x(T) = x_{\text{ref}}(T)$.

The quickest entrainment problem can capture, e.g., the problem of eliminating jetlag as quickly as possible. In this case, the reference $x_{\text{ref}}(t)$ is the desired circadian state trajectory (without the jetlag). QEP is, in optimal control terminology, a minimum time optimal control problem, which can be solved using, for example:

1. Calculus of variations: this approach includes optimal control techniques (Bryson and Ho 1975), resulting in both *open-loop* optimal lighting strategies (Serkh and Forger 2014), where the optimal light signal is computed offline, and *feedback* optimal lighting strategies, where the optimal light signal is computed as a function of the current circadian state (Qiao et al. 2017).
2. Model predictive control: this approach is based on optimizing the light input in a receding horizon optimization (Mott et al. 2003). However, since minimum time control problems cannot be directly posed in a

receding horizon fashion, the optimization objective is typically formulated as to minimize the error between the circadian states and their respective references.

Examples of the solutions of QEP for various jetlag cases using the third order Kronauer model (Jewett et al. 1999) are shown in Fig. 3. The trajectories of the two circadian oscillator states and their respective references (with the subscript r) are shown in each graph. The optimal light inputs follow a bang-bang control law, as is typical in minimum time control problems. Serkh and Forger (2014) exploited this assumption by computing the optimal light input by parameterizing it by the switching times between light on and light off. On the other hand, Qiao et al. (2017) use Pontryagin's Maximum Principle to compute the optimal light input.

Optimal Control Involving Neurobehavioral Dynamics

The formulation of QEP ignores the dynamics of the sleep and neurobehavioral states ($s(t)$). Therefore, the resulting optimal solutions may be impractical from the sleep perspective. To address this limitation, QEP can be modified to include the sleep dynamics. There are two variations of the new formulation, one where sleep is considered autonomous and one where sleep can be scheduled, as discussed in the previous section.

In the autonomous sleep-wake transitions case, the subject goes to sleep when their sleepiness ($B(t)$) reaches a certain high level ($B_{\max}$) and wakes up when it reaches a certain low level ($B_{\min}$). In the scheduled sleep-wake transitions case, we assume that the transitions can occur any time. However, to add realism to the model, we disallow the situation where the sleep period is too long (which is impossible in reality; the subject would spontaneously wake up), and the situation where the awake period is too long (which leads to sleep deprivation). This leads to two new versions of the quickest entrainment problem.

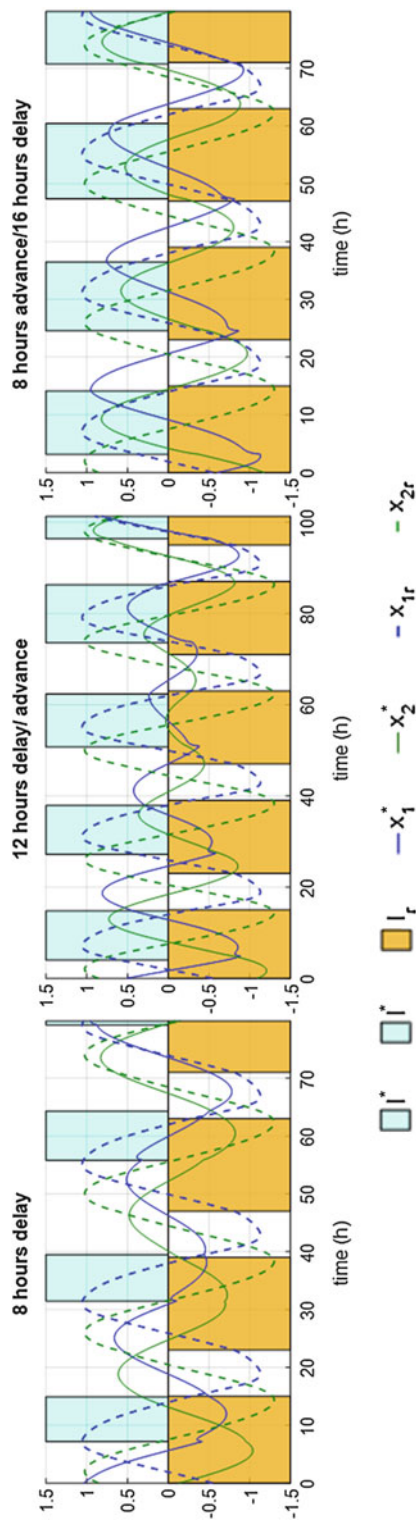
Quickest Entrainment Problem with Autonomous Sleep (QEP-A) Given the dynamics of the circadian, sleep, and neurobehavioral states in Eqs. (1), (2), (3), (4), and (5), where the switching of the sleep dynamics is autonomous, an initial condition $x(0) = x_0$, $s(0) = s_0$, and a reference circadian state trajectory $x_{\text{ref}}(t)$, determine the input $u(t)$ that minimizes the entrainment time T . Here, the entrainment time T is defined as the earliest time such that $x(T) = x_{\text{ref}}(T)$.

Quickest Entrainment Problem with Schedulable Sleep (QEP-S) Given the dynamics of the circadian, sleep, and neurobehavioral states in Eqs. (1), (2), (3), (4), and (5), an initial condition $x(0) = x_0$, $s(0) = s_0$, and a reference circadian state trajectory $x_{\text{ref}}(t)$, determine the input $u(t)$ and the sleep schedule that meets the scheduling constraint, such that the entrainment time T is minimized. Here, the entrainment time T is defined as the earliest time such that $x(T) = x_{\text{ref}}(T)$.

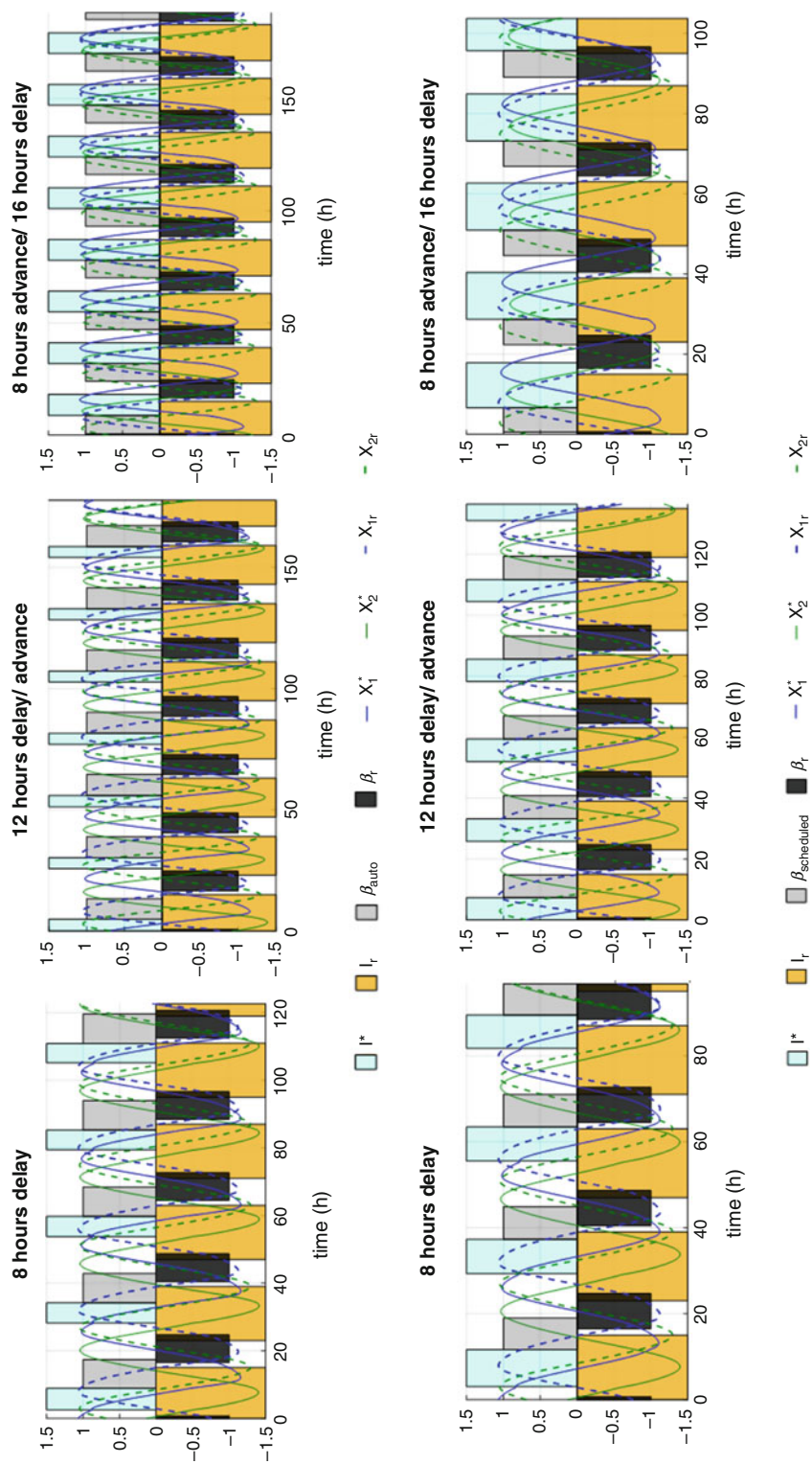
QEP-A and QEP-S are more challenging to solve compared to QEP because the associated systems dynamics (i) have higher orders and (ii) are non-smooth because of the switching. Further, QEP-S is more challenging to solve compared to QEP-A because QEP-S has more optimization variables and constraints (i.e., the sleep schedule). Both QEP-A and QEP-S can be solved using the calculus of variations (Julius et al. 2017). Samples of the solutions can be seen in Fig. 4. The circadian rhythm model used to obtain these results are the same third-order Kronauer model (Jewett et al. 1999) as in the previous subsection.

Another type of optimal control problems that involves the dynamics of the neurobehavioral processes is the alertness optimization problem. The idea is to have the subject be maximally alert during a prescribed time interval, e.g., between given T_{start} and T_{end} . Such problem can be formulated as follows.

Alertness Optimization Problem (AOP) Given the dynamics of the circadian, sleep, and neurobehavioral states in Eqs. (1), (2), (3), (4), (5),



Control of Circadian Rhythms and Related Processes, Fig. 3 Examples of QEP solutions. All simulations are run until entrainment. Cyan bands represent when the optimal control input (lighting) is on. Orange bands represent when the natural (reference) daylight is on (assuming 16 h/8 h light/dark pattern)



Control of Circadian Rhythms and Related Processes, Fig. 4 Solutions of QEP-A (top row) and QEP-S (bottom row), for various jetlag cases. All simulations are run until entrainment. Cyan bands represent when the optimal control input (lighting) is on. Orange bands represent when the natural (reference) daylight is on (assuming 16h/8h light/dark pattern). Light gray bands represent the sleep intervals of the subject. Dark gray bands represent the sleep intervals of the reference

and (6), an initial condition $x(0) = x_0, s(0) = s_0$, determine the input $u(t)$ and the sleep schedule that meets the scheduling constraint, such the minimum alertness $A_{\min}$ is maximized. Here, $A_{\min}$ is defined as

$$A_{\min} \triangleq \min_{T_{\text{start}} \leq t \leq T_{\text{end}}} A(t). \quad (7)$$

AOP can also be solved using the calculus of variations (Zhang et al. 2012). Note that further variations of the OAP can be formulated by, for example, adding the constraints on sleep scheduling.

Related Topics

Spectrally Tunable Lighting

While some of optimal solutions discussed above require turning off the circadian light for some period of time, in practice, the subject is not necessarily in absolute darkness. The circadian rhythms are known to be mostly affected by blue light (Foster and Kreitzman 2017). In practice, the computed optimal control may be implemented in spectrum-variable illumination or via spectral manipulation, e.g., by using blue-blocking filters, leading to smart spaces (e.g., hotel rooms, hospitals) that regulate the occupants' circadian rhythms.

State Estimation

The circadian states $x(t)$ are not directly measurable. Rather, we typically have measurements of indirect markers that are driven by the circadian rhythms, such as actigraphy, heart rate, and body temperature (Foster and Kreitzman 2017). Ideally, we want to estimate the system states from these measurements. A possible approach is to use first-order models that capture only the dynamics of the circadian phase $\theta(t)$. Then, we extract the periodic content of the biometric data stream, for example, by using Fourier analysis or algorithms such as the adaptive notch filter (ANF) (Julius et al. 2016). We can associate the phase shifts of these periodic signals with those of the circadian rhythms and thus use this idea

to identify the phase response curve and estimate the circadian phase.

Cross-References

- [Building Lighting Systems](#)
- [Modeling Hybrid Systems](#)

Bibliography

- Abel JH, Chakrabarty A, Klerman EB, Doyle III FJ (2019) Pharmaceutical-based entrainment of circadian phase via nonlinear model predictive control. *Automatica* 100:336–348
- Achermann P, Borbely AA (2003) Mathematical models of sleep regulation. *Front Biosci* 8:683–693
- Bryson AE, Ho YC (1975) *Applied optimal control*. Taylor & Francis, New York, USA
- Doyle III FJ, Gunawan R, Bagheri N, Mirsky H, To TL (2006) Circadian rhythm: a natural, robust, multi-scale control system. *Comput Chem Eng* 30(10–12):1700–1711
- Foster R, Kreitzman L (2017) *Circadian rhythms*. Oxford University Press, Oxford, UK
- Jewett ME, Forger DB, Kronauer RE (1999) Revised limit cycle oscillator model of human circadian pacemaker. *J Biol Rhythm* 14(6):493–499
- Julius AA, Zhang JX, Qiao W, Wen JT (2016) Multi-input adaptive notch filter and observer for circadian phase estimation. *Int J Adapt Control Signal Process* 30(8–10):1375–1388
- Julius AA, Yin J, Wen JT (2017) Time-optimal control for circadian entrainment for a model with circadian and sleep dynamics. In: *Proceedings of IEEE Conference on Decision and Control*, Melbourne, pp 4709–4714
- Leloup JC, Goldbeter A (2003) Toward a detail computational model for the mammalian circadian clock. *Proc Natl Acad Sci* 100:7051–7056
- Mott C, Mollicone D, van Wollen M, Huzmezan M (2003) Modifying the human circadian pacemaker using model based predictive control. In: *Proceedings of American Control Conference*, Denver, pp 453–458
- Qiao W, Wen JT, Julius AA (2017) Entrainment control of phase dynamics. *IEEE Trans Autom Control* 62(1):445–450
- Serikh K, Forger DB (2014) Optimal schedules of light exposure for rapidly correcting circadian misalignment. *PLOS Comput Biol* 10(4):e1003523
- Zhang JX, Wen JT, Julius AA (2012) Optimal circadian rhythm control with light input for rapid entrainment and improved vigilance. In: *Proceedings of IEEE Conference on Decision and Control*, pp 3007–3012

Control of Drug Delivery for Type 1 Diabetes Mellitus

Kelilah L. Wolkowicz, Francis J.

Doyle III, and Eyal Dassau

Harvard John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA

Abstract

The automated insulin delivery (AID) device known as an artificial pancreas (AP) contains a closed-loop feedback controller that enables glucose management for those with type 1 diabetes mellitus (T1DM). The AP system is comprised of two elements: devices and software. Devices include insulin delivery devices and glucose sensors, while software includes the user interface and the control algorithm. Decision support systems (DSS) also support treatment decisions for those with diabetes mellitus if the user does not use a closed-loop system. This chapter is divided into three sections, beginning with a description of controller components, design objectives, the challenges and the limitations of AID devices. This discussion is followed by recent advances in closed-loop clinical trials, common control algorithms, and clinical challenges. Finally, general expectations for future developments within AID engineering and clinical translation are outlined.

Keywords

Automated insulin delivery · Artificial pancreas · Controller design · Clinical trials · Clinical translation · Decision support system · Qualitative comparison · Healthcare

Introduction

Type 1 diabetes mellitus (T1DM) is a disease that prevents the body from properly producing

insulin, resulting in abnormal metabolism of carbohydrates that generates elevated levels of glucose in the blood and urine (American Diabetes Association 2013). Glucose control is a complex and challenging problem. It includes tracking, rejection of both measured and unmeasured disturbances, uncertainty, and numerous safety constraints. Moreover, the addition of the human-in-the-loop can contribute to system instability, introduce further disturbances, and may initiate signal overrides.

Over 130 clinical studies have demonstrated the safety and efficacy of closed-loop systems for glucose control since 2004. These studies have included different algorithms evaluated across age groups and clinical conditions. The application of advanced control algorithms has led to advanced glycemic control, leading to reduced low-glucose events, as well as achieved tighter control (improved balance in glucose levels) overnight (Doyle et al. 2014; Bertachi et al. 2018; Kowalski 2015). However, only one AID system (Medtronic MiniMed 670G) and two semiautomated systems (Medtronic MiniMed 630G and Tandem t:slim X2 with Basal-IQ) are currently approved for use by the Food and Drug Administration (FDA) (Garg et al. 2017). These semiautomated systems are important stepping stones in the pathway toward hybrid and, ultimately, fully closed-loop systems. The design of control algorithms for a medical application is complex; the user may play an integral part in the intervention (control action) or add unmeasured disturbances. Yet, it is when humans are included in the loop that tight control and performance become even more important to ensure the safety of AID systems.

Engineering Design and Clinical Translation of Type 1 Diabetes Treatment

Preclinical and clinical aspects of AID device development prior to 2014 are summarized in Doyle et al. (2014) and Kudva et al. (2014). Since 2014, significant improvements have been

made with regard to blood glucose (BG) control, particularly in terms of moving AID systems from clinic-based trials toward those that mimic everyday life. This progress relies on identifying and defining the specific factors that pertain to this engineering design problem.

Closed-loop AID devices, such as the AP, provide a route to automate glycemic control so that little to no user intervention is required. The closed-loop system automatically computes and delivers the appropriate insulin dose to the patient, allowing the controller to maintain a desired blood glucose concentration, as illustrated in Fig. 1. First, the desired BG concentration is defined as a specific value (set point) or a target range (zone), which can be fixed or time-varying. The controller compares the desired BG with a measured BG value that is typically received via a single continuous glucose monitor (CGM) attached to the user's abdomen or arm. The controller works to minimize the difference, or error, between the desired and actual BG values. Most CGMs measure the user's actual BG every 5 minutes, resulting in a time delay between the controller's adjustment in insulin delivery and the effect in the user's blood glucose levels. A delay also results due to the time it takes insulin to reach peak serum concentration, which influences the ability to reject disturbances such as inaccuracies in meal bolus dosing, exercise, or stress (Nimri et al. 2019; Dassau et al. 2017a); otherwise the user may request more insulin than is needed (stacking insulin), which in turn may result in dangerously low-glucose levels.

The second element of the AP system is the controller. To date, the most common control algorithms used within AP systems are model predictive control (MPC), proportional-integral-derivative control (PID), and fuzzy logic control (FL). Each algorithm has tuning parameters to adjust how the controller will react to deviations in BG from the desired value. Some control algorithms also integrate insulin feedback and/or insulin-on-board (IOB) constraints to help mitigate insulin stacking. Insulin feedback accelerates the insulin profile, improving the

performance of the controller (Lee et al. 2013). IOB constraints are based on the insulin delivery history (e.g., after a meal bolus) and attempt to approximate the amount of insulin present within the body (Gondhalekar et al. 2016; Dassau et al. 2015). Various models can also be integrated with the AP controller, including physiological (e.g., glucose-insulin dynamics) or empirical models, each of which can either be static or adaptive in nature. Predictive low-glucose suspend (PLGS) is often used in semiautomated systems to suspend basal insulin delivery when low BG is predicted and resume insulin delivery once BG levels begin to rise. Once the closed-loop controller has calculated the error between the actual and desired BG level, the output of the control algorithm communicates the amount of insulin that should be delivered (known as an insulin bolus) to the user via an insulin delivery device, either an insulin pump or insulin pen.

Another important consideration is how closely a user is involved with the AP system, i.e., whether or not the user is directly involved in the control loop (*patient treatment action block*). For example, some AP designs require the user to input a meal announcement, which indicates the size of the meal consumed, allowing the controller to calculate and deliver a full or partial insulin bolus. In other AP designs, the user manually calculates and delivers their meal bolus. Manual meal boluses and/or announcements can often result in tighter control due to the delays in insulin action after delivery and glucose sensing by the controller alone. Users may also provide exercise announcements, allowing the AP to temporarily suspend insulin boluses. Potentially, additional hormones and medications may also be delivered to the user, either manually or through the automated system. These medications include pramlintide, glucagon, and oral dextrose. However, adding user involvement to control performance increases safety concerns, given the unpredictable nature of human behavior (Doyle et al. 2014). The impact of the degree of automation and control performance is critical when considering the design and primary objectives of an AP system.

user is physically active, as well as to improve the controller responsiveness to the activity and glucose trends.

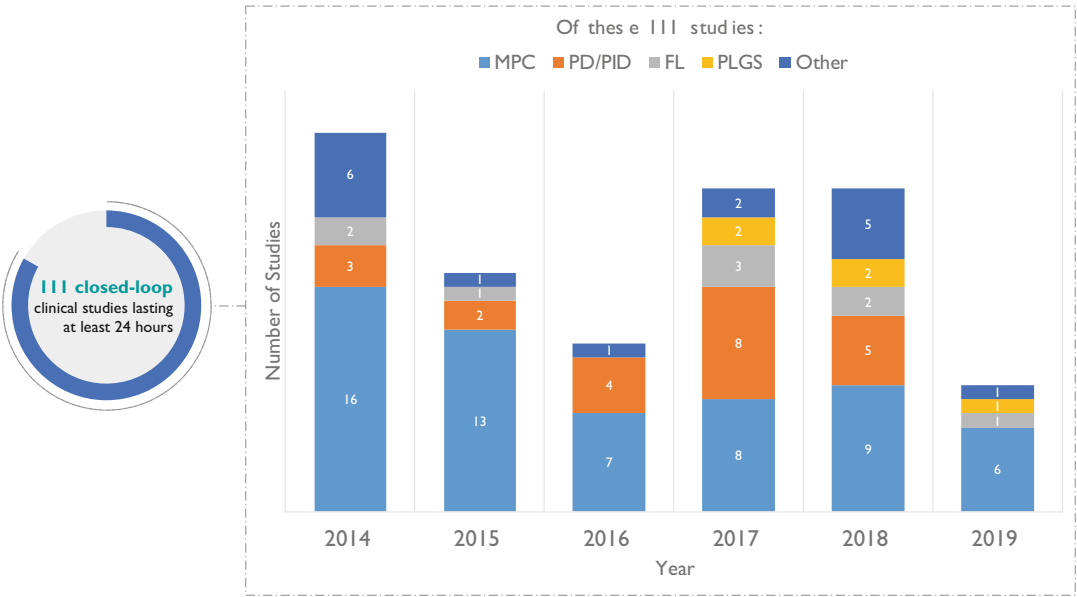
In addition to sensor development, inter-device communication is imperative. Medical devices need to be robust against potential communication and connectivity issues, especially when integrating different hardware platforms. Current clinical trials are testing an AP controller that connects to the user's insulin pump via Bluetooth (Deshpande et al. 2018). This approach brings the technology closer to what the user expects in their everyday life, but frustration ensues when the connection or communication between the two devices is sporadic. Increasing the battery life of these devices is also an important factor, especially when Bluetooth or other wireless communication is needed. Embedded controllers receive CGM data and communicate with a pump and display interface for user interaction (such as a smart phone) via Bluetooth Low Energy (Deshpande et al. 2018; Chakrabarty et al. 2018). The recent push toward integrated devices with embedded controllers may lend to lowering energy costs (Chakrabarty et al. 2018). These system advances get us closer to miniaturized, low-power technology, allowing for wearable and implantable medical devices with long life expectancy (Chakrabarty et al. 2018).

Another limitation facing AP development is human physiology. The high intra- and intersubject variability in insulin kinetics and dynamics affects the estimation of insulin time-to-peak activity, as well as the metabolic clearance rate (Doyle et al. 2014; Haidar et al. 2013). For fully automated AP systems that respond exclusively to CGM feedback, such uncertainties can result in the restricted ability of the controller to respond adequately to unannounced disturbances, leading to over- or under-calculation of insulin dosing. Controllers must be sufficiently robust to these variations. Incorporating additional physiological inputs in control algorithms may result in superior glucose regulation (Doyle et al. 2014; Kudva et al. 2014). Adaptive algorithms that learn how to react to certain user characteristics over time may help to address this intra- and intersubject

variability. For example, some adaptive algorithms optimize the tuning of a user's day- to nighttime basal insulin needs, insulin sensitivity, or meal bolus information for slow inter-day variability. Other algorithms adapt to exercise based on wearable sensors (Castle et al. 2018). As machine learning and artificial intelligence-based technology become more established and suitable for clinical use, these methods may also provide powerful tools for glycemic control and management (Contreras and Vehi 2018). Finally, it must be noted that insulin cannot be removed from the body once delivered to the user. Consequently, controllers must account for IOB (Doyle et al. 2014). While the controller (or the user, manually) can correct with additional carbohydrates or glucagon, this approach may require increased consumption of insulin (Castle et al. 2018; Christiansen et al. 2017; Haidar et al. 2015).

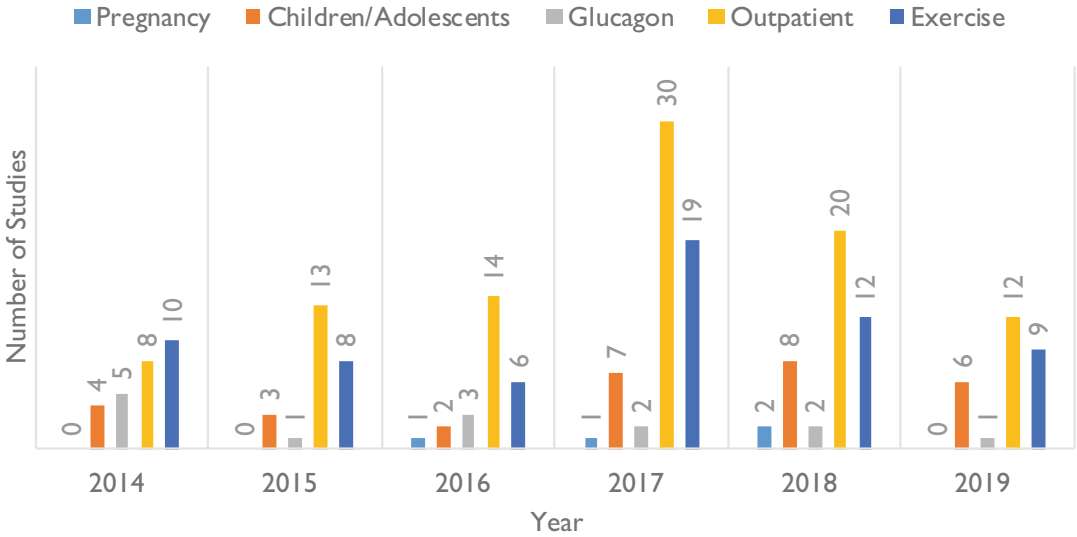
Recent Advances in Clinical Trials

The increasing number of clinical studies lasting over 24 hours reveals the progress in analysis and performance of closed-loop AP systems (Fig. 2). Since 2014, the average percentage of closed-loop time maintained within the euglycemic range (using 70–140 mg/dL) has steadily increased to 72.4%. Additionally, the average percentage of closed-loop time within the hypoglycemic range was 2.29% (using <70 mg/dL). These studies have continued to incorporate more difficult challenges within their protocols (Fig. 3). Yet, there is still inconsistency across those protocols, with very few following the same for closed-loop initiation, meal size and timing, or exercise length and intensity (Doyle et al. 2014). In February 2017, the Advanced Technologies & Treatments for Diabetes (ATTD) Congress convened a panel to address the issue of lack of standardized metrics for analyzing CGM data (Danne et al. 2017). This conference resulted in a consensus report that identified 14 key metrics now recommended to assess and document glycemic control (Danne et al. 2017). These metrics were further updated in 2019 (Battelino et al. 2019).



Control of Drug Delivery for Type 1 Diabetes Mellitus, Fig. 2 Comparison of number of clinical trials per year, including and since 2014 and corresponding types of control algorithms used, where MPC is model predic-

tive control, PD is proportional-derivative control, PID is proportional-integral-derivative control, FL is fuzzy logic, and PLGS is predictive low-glucose suspend



Control of Drug Delivery for Type 1 Diabetes Mellitus, Fig. 3 Engineering and clinical study categories in the AP world by year. Exercise includes studies allowing free-living conditions

The largest and most imminent clinical risk to people with DM is hypoglycemia, which occurs as a result of over delivery or stacking of insulin, exercise, stress, or consumption of alcohol. An increasing amount of AP designs are incorporating safety systems to mitigate this risk, including insulin feedback, IOB calculation and penalization, low-glucose prediction, remote monitoring, and exercise detection (Rossetti et al. 2017; Pinsker et al. 2018; Forlenza et al. 2017). While several research groups have developed algorithms that incorporate insulin estimation within the control loop, no clinical trials have yet included these algorithms within their study protocol (Hajizadeh et al. 2017, 18AD). One of the major limitations is that minimally invasive and portable insulin measurement sensors are still in their infancy. Insulin measurement information has the opportunity to potentially remove the risk of insulin stacking, in addition to removing much of the burden of carbohydrate counting and the need for an insulin bolus calculator (Forlenza 2017).

To manage changes in BG levels, there are two primary routes for insulin delivery: subcutaneous (SC) and intraperitoneal (IP). SC methods receive the insulin bolus from an external insulin pump, and delivery can occur both from multiple daily injections (most common, overall) and continuous subcutaneous insulin infusion (most common for AP). IP methods, on the other hand, can receive the insulin bolus either internally or externally but require users to have an implantable pump or a intraperitoneal port (Dassau et al. 2017b). IP insulin delivery is currently the least common approach due to the requirement that patients must have an implanted catheter, but it provides faster insulin action and elimination than SC, allowing the AP controller to institute changes in glucose levels more quickly. Continuous IP insulin infusion has been demonstrated to improve glycemic control and the quality of life of those with particularly hard to manage (brittle) T1DM (van Dijk et al. 2016, 2015; Logtenberg et al. 2009).

Algorithms have also seen major enhancements. MPC and PID algorithms have continued to undergo the most extensive evaluation in

comparison to others, with 53% and 20% of studies performed, respectively (Fig. 2). Both controllers have demonstrated strong performance across all metrics, including perturbation attenuation when considering challenging disturbances. Fuzzy logic (FL) was the third most evaluated controller, used in 8% of long-term clinical studies. While PLGS algorithms were only used in 5%, they have been used in both the semiautomated systems approved for use by the FDA: Medtronic MiniMed 630G and Tandem t:slim X2 with Basal-IQ. Other control strategies constituted 14%. MPC have shown recent advancement with several specified algorithmic methodologies, including model personalization and asymmetric input cost functions. Model personalization adapts input parameters to be subject-specific, ensuring that controller suggested insulin delivery rates are limited to safety purposes (e.g., avoiding hypoglycemia) due to unique insulin sensitivities (Hajizadeh et al. 18AD; Buckingham et al. 2018). Asymmetric input cost functions decouple the design of the controller's response to hyper- and hypoglycemia because a rise of 30 mg/dL in BG is not equivalent in risk to the same drop in BG, which could be fatal (Gondhalekar et al. 2016; Dassau et al. 2015; Lee et al. 2016). The integration of each of these algorithmic methodologies improves the ability of the AP to more holistically manage an individual's changes in glucose because they compensate for realistic inter- and intra-subject variability.

In an effort to push the performance margins of the AP controller, more studies have incorporated high-impact disturbances within their protocols. Since the timing, size, and composition of meals is variable, compensating for meal disturbances is one of the most difficult challenges in glucose control. Consequently, meal challenges are incorporated to see how the AP handles meals with higher protein and/or fat content, simplified meal insulin boluses, and, most commonly, when the user does not alert the controller of a meal (unannounced meals). Over 48 closed-loop AP studies have also incorporated exercise challenges to aid the controller's ability to mitigate hypoglycemic events. This scenario can prove

especially challenging if a snack is taken pre-exercise. At the same time, an increasing number of studies mimic everyday life through outpatient or free-living assessments. In 2014, about 30% of closed-loop AP studies included outpatient protocols, since then, they have dramatically increased to 52% in 2015, 54% in 2016, 51% in 2017, 45% in 2018, and 43% in 2019.

Despite the progress in AID system control, several limitations still remain that restrict the performance of fully automated systems. User safety is the most important aspect of medical device design, followed by user convenience. Further clinical studies with a variety of participant cohorts (e.g., pediatric, pregnant, renal insufficient, and the elderly) using larger sample sizes may help to improve upon time in range (Bekiari et al. 2018). While the majority of AID systems are evaluated for T1DM, a transition to T2DM is the next step. A complete understanding about the effects of pregnancy in women with T1DM and gestational DM is not known, but these users require tight glucose control to prevent unwanted outcomes (Bertachi et al. 2018). For these users and those in urgent or critical care, adaptive control strategies may be advantageous due to the continuous changes in insulin requirement during pregnancy and illness (Bertachi et al. 2018).

Summary and Future Directions

Recent advances have improved our understanding of patients' interactions with closed-loop AID systems. In the near future, the number of closed-loop clinical AID studies and the of number of potential controller configurations will increase. Introducing advances in sensor technology within AID closed-loop systems will enable the next generation of algorithms to dramatically enhance glycemic control with added physiological information (e.g., physical activity, cortisol, and alcohol levels).

The main limitations of current AID systems are related to variability in outcome reporting, sample size, and short follow-up duration of individual trials (Bekiari et al. 2018). Further

limitations are associated with the accuracy and reliability of available CGM systems, as well as the delayed therapeutic effect of insulin when administered through the SC route. Standardizing controller evaluations and reporting performances in clinical trials will make qualitative analysis more meaningful across the various systems being tested; the adaptation to the International consensus recommended metrics will aid in clearly reporting study outcomes (Danne et al. 2017; Battelino et al. 2019). Additionally, challenges still exist in terms of user acceptance and trust when using an AID system. As stated by Deshpande et al., user trust and the accompanying psychosocial impacts of AP technology are areas that require continued improvement and study (Deshpande et al. 2018). In doing so, AID systems can continue to advance, improving the quality of life of their users.

Cross-References

- [Automated Anesthesia Systems](#)
- [Automated Insulin Dosing for Type 1 Diabetes](#)
- [Control of Anemia in Hemodialysis Patients](#)
- [Control of Drug Delivery for Type 1 Diabetes Mellitus](#)

Recommended Reading

- Battelino T, Danne T, Amiel SA et al (2019) Clinical targets for continuous glucose monitoring data interpretation: recommendations from the international consensus on time-in-range. *Diabetes Care* 42(8):1593–1603
- Danne T, Nimri R, Battelino T et al (2017) International consensus on use of continuous glucose monitoring. *Diabetes Care* 40:1631–1640. <https://doi.org/10.2337/dc17-1600>
- Doyle III FJ, Huyett LM, Lee JB et al (2014) Closed-loop artificial pancreas systems: engineering the algorithms. *Diabetes Care* 37:1191–1197. <https://doi.org/10.2337/dc13-2108>
- Haidar A, Legault L, Messier V et al (2015) Comparison of dual-hormone artificial pancreas, single-hormone artificial pancreas, and conventional insulin pump therapy for glycaemic control in patients with type 1 diabetes: an open-label randomised controlled crossover trial. *Lancet Diabetes Endocrinol* 3:17–26. [https://doi.org/10.1016/S2213-8587\(14\)70226-8](https://doi.org/10.1016/S2213-8587(14)70226-8)
- Kudva YC, Carter RE, Cobelli C et al (2014) Closed-loop artificial pancreas systems: physiological input

to enhance next-generation devices. *Diabetes Care* 37:1184–1190. <https://doi.org/10.2337/dc13-2066>

Nimri R, Ochs AR, Pinsker JE et al (2019) Decision support systems and closed loop. *Diabetes Technol Ther* 21:S-42–S-56. <https://doi.org/10.1089/dia.2019.2504>

Bibliography

- American Diabetes Association (2013) Diagnosis and classification of diabetes mellitus. *Diabetes* 36:S67–S74. <https://doi.org/10.1016/j.autrev.2014.01.020>
- Battelino T, Danne T, Amiel SA et al (2019) Clinical targets for continuous glucose monitoring data interpretation: recommendations from the international consensus on time-in-range. *Diabetes Care* 42(8):1593–1603
- Bekiarı E, Kitsios K, Thabit H et al (2018) Artificial pancreas treatment for outpatients with type 1 diabetes: systematic review and meta-analysis. *BMJ* 361:k1310. <https://doi.org/10.1136/bmj.k1310>
- Bertachi A, Ramkissoon CM, Bondia J, Vehí J (2018) Automated blood glucose control in type 1 diabetes: a review of progress and challenges. *Endocrinol Diabetes y Nutr (English ed)* 65:172–181. <https://doi.org/10.1016/j.endinu.2017.10.011>
- Buckingham BA, Forlenza GP, Pinsker JE et al (2018) Safety and feasibility of the OmniPod hybrid closed-loop system in adult, adolescent, and pediatric patients with type 1 diabetes using a personalized model predictive control algorithm. *Diabetes Technol Ther* 20:dia.2017.0346. <https://doi.org/10.1089/dia.2017.0346>
- Castle JR, El Youssef J, Wilson LM et al (2018) Randomized outpatient trial of single- and dual-hormone closed-loop systems that adapt to exercise using wearable sensors. *Diabetes Care* 41:1471–1477. <https://doi.org/10.2337/dc18-0228>
- Chakrabarty A, Zavitsanou S, Doyle III FJ, Dassau E (2018) Event-triggered model predictive control for embedded artificial pancreas systems. *IEEE Trans Biomed Eng* 65:575–586. <https://doi.org/10.1109/TBME.2017.2707344>
- Christiansen SC, Fougner AL, Stavadahl Ø et al (2017) A review of the current challenges associated with the development of an artificial pancreas by a double subcutaneous approach. *Diabetes Ther* 8:489–506. <https://doi.org/10.1007/s13300-017-0263-6>
- Contreras I, Vehí J (2018) Artificial intelligence for diabetes management and decision support: literature review. *J Med Internet Res* 20:e10775. <https://doi.org/10.2196/10775>
- Danne T, Nimri R, Battelino T et al (2017) International consensus on use of continuous glucose monitoring. *Diabetes Care* 40:1631–1640. <https://doi.org/10.2337/dc17-1600>
- Dassau E, Brown SA, Basu A et al (2015) Adjustment of open-loop settings to improve closed-loop results in type 1 diabetes: a multicenter randomized trial. *J Clin Endocrinol Metab* 100:3878–3886. <https://doi.org/10.1210/jc.2015-2081>
- Dassau E, Pinsker JE, Kudva YC et al (2017a) Twelve-week 24/7 ambulatory artificial pancreas with weekly adaptation of insulin delivery settings: effect on hemoglobin A1c and hypoglycemia. *Diabetes Care* 40:dc171188. <https://doi.org/10.2337/dc17-1188>
- Dassau E, Renard E, Place J et al (2017b) Intraperitoneal insulin delivery provides superior glycaemic regulation to subcutaneous insulin delivery in model predictive control-based fully-automated artificial pancreas in patients with type 1 diabetes: a pilot study. *Diabetes Obes Metab* 19:1698–1705. <https://doi.org/10.1111/dom.12999>
- Deshpande S, Pinsker JE, Zavitsanou S et al (2018) Design and clinical evaluation of the interoperable artificial pancreas system (iAPS) smartphone App: interoperable components with modular design for progressive artificial pancreas research and development. *Diabetes Technol Ther* 21:35–43. <https://doi.org/10.1089/dia.2018.0278>
- Doyle III FJ, Huyett LM, Lee JB et al (2014) Closed-loop artificial pancreas systems: engineering the algorithms. *Diabetes Care* 37:1191–1197. <https://doi.org/10.2337/dc13-2108>
- El-Laboudi A, Oliver NS, Cass A, Johnston D (2012) Use of microneedle array devices for continuous glucose monitoring: a review. *Diabetes Technol Ther* 15:101–115. <https://doi.org/10.1089/dia.2012.0188>
- Forlenza GP (2017) Relevance of bolus calculators in current hybrid closed loop systems. *Diabetes Technol Ther* 19:400–401. <https://doi.org/10.1089/dia.2017.0216>
- Forlenza GP, Raghinaru D, Cameron F et al (2017) Predictive hyperglycemia and hypoglycemia minimization: in-home double-blind randomized controlled evaluation in children and young adolescents. *Pediatr Diabetes* 19:420–428. <https://doi.org/10.1111/pedi.12603>
- Garg SK, Weinzimer SA, Tamborlane WV et al (2017) Glucose outcomes with the in-home use of a hybrid closed-loop insulin delivery system in adolescents and adults with type 1 diabetes. *Diabetes Technol Ther* 19:155–163. <https://doi.org/10.1089/dia.2016.0421>
- Gondhalekar R, Dassau E, Doyle III FJ (2016) Periodic zone-MPC with asymmetric costs for outpatient-ready safety of an artificial pancreas to treat type 1 diabetes. *Automatica* 71:237–246. <https://doi.org/10.1016/j.automatica.2016.04.015>
- Haidar A, Elleri D, Kumareswaran K et al (2013) Pharmacokinetics of insulin aspart in pump-treated subjects with type 1 diabetes: reproducibility and effect of age, weight, and duration of diabetes. *Diabetes Care* 36:173–174. <https://doi.org/10.2337/dc13-0485>
- Haidar A, Legault L, Matteau-Pelletier L et al (2015) Outpatient overnight glucose control with dual-hormone artificial pancreas, single-hormone artificial pancreas, or conventional insulin pump therapy in children and adolescents with type 1 diabetes: an open-

- label, randomised controlled trial. *Lancet Diabetes Endocrinol* 3:595–604. [https://doi.org/10.1016/S2213-8587\(15\)00141-2](https://doi.org/10.1016/S2213-8587(15)00141-2)
- Hajizadeh I, Rashid M, Samadi S et al (18AD) Adaptive and personalized plasma insulin concentration estimation for artificial pancreas systems. *J Diabetes Sci Technol* 12:639–649
- Hajizadeh I, Turksoy K, Cengiz E, Cinar A (2017) Real-time estimation of plasma insulin concentration using continuous subcutaneous glucose measurements in people with type 1 diabetes. *Proc Am Control Conf* 5193–5198. <https://doi.org/10.23919/ACC.2017.7963761>
- Kowalski A (2015) Pathway to artificial pancreas systems revisited: moving downstream. *Diabetes Care* 38:1036–1043. <https://doi.org/10.2337/dc15-0364>
- Kudva YC, Carter RE, Cobelli C et al (2014) Closed-loop artificial pancreas systems: physiological input to enhance next-generation devices. *Diabetes Care* 37:1184–1190. <https://doi.org/10.2337/dc13-2066>
- Lee JB, Dassau E, Seborg DE, Doyle III FJ (2013) Model-based personalization scheme of an artificial pancreas for Type 1 diabetes applications. *Am Control Conf (ACC)* 2013:2911–2916. <https://doi.org/10.1109/ACC.2013.6580276>
- Lee JB, Dassau E, Gondhalekar R et al (2016) Enhanced model predictive control (eMPC) strategy for automated glucose control. *Ind Eng Chem Res* 55:11857–11868. <https://doi.org/10.1021/acs.iecr.6b02718>
- Logtenberg SJJ, Kleefstra N, Houweling S et al (2009) Improved glycemia control with intraperitoneal versus subcutaneous insulin in type 1 diabetes: a randomized controlled trial. *Diabetes Care* 32:1372–1377. <https://doi.org/10.2337/dc08-2340.Clinical>
- Miller PR, Narayan RJ, Polsky R (2016) Microneedle-based sensors for medical diagnosis. *J Mater Chem B* 4:1379–1383. <https://doi.org/10.1039/c5tb02421h>
- Nimri R, Ochs AR, Pinsker JE et al (2019) Decision support systems and closed loop. *Diabetes Technol Ther* 21:S-42–S-56. <https://doi.org/10.1089/dia.2019.2504>
- Pinsker JE, Laguna Sanz AJ, Lee JB et al (2018) Evaluation of an artificial pancreas with enhanced model predictive control and a glucose prediction trust index with unannounced exercise. *Diabetes Technol Ther* 20:455–464. <https://doi.org/10.1089/dia.2018.0031>
- Rossetti P, Quirós C, Moscardó V et al (2017) Closed-loop control of postprandial glycemia using an insulin-on-board limitation through continuous action on glucose target. *Diabetes Technol Ther* 19:355–362. <https://doi.org/10.1089/dia.2016.0443>
- van Dijk PR, Logtenberg SJJ, Hendriks SH et al (2015) Intraperitoneal versus subcutaneous insulin therapy in the treatment of type I diabetes mellitus. *Neth J Med* 73:399–409
- van Dijk PR, Logtenberg SJJ, Chisalita SI et al (2016) Different effects of intraperitoneal and subcutaneous insulin administration on the GH-IGF-1 axis in type 1 diabetes. *J Clin Endocrinol Metab* 101:2493–2501. <https://doi.org/10.1210/jc.2016-1473>

Control of Fluid Flows and Fluid-Structure Models

Jean-Pierre Raymond

Institut de Mathématiques, Université Paul Sabatier Toulouse III and CNRS, Toulouse Cedex, France

Abstract

In this entry, some fluid models are introduced, and corresponding controllability and stabilization results are presented. A short overview on control results for fluid-structure interaction models is then given.

Keywords

Control · Stabilization · Fluid flows · Fluid-structure systems

Some Fluid Models

We consider a fluid flow occupying a bounded domain $\Omega_F \subset \mathbb{R}^N$, with $N = 2$ or $N = 3$, at the initial time $t = 0$ and a domain $\Omega_F(t)$ at time $t > 0$. Let us denote by $\rho(x, t) \in \mathbb{R}^+$ the density of the fluid at time t at the point $x \in \Omega_F(t)$ and by $u(x, t) \in \mathbb{R}^N$ its velocity. The fluid flow equations are derived by writing the mass conservation

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho u) = 0 \quad \text{in } \Omega_F(t), \quad \text{for } t > 0, \quad (1)$$

and the balance of momentum

$$\rho \left(\frac{\partial u}{\partial t} + (u \cdot \nabla) u \right) = \operatorname{div} \sigma + \rho f \quad (2)$$

in $\Omega_F(t)$, for $t > 0$,

where σ is the so-called constraint tensor and f represents a volumic force. For an isothermal fluid, there is no need to complete the system by the balance of energy. The physical nature of the fluid flow is taken into account in the choice of the constraint tensor σ . When the volume is

preserved by the fluid flow transport, the fluid is called incompressible. The incompressibility condition reads as $\operatorname{div} u = 0$ in $\Omega_F(t)$. The incompressible Navier-Stokes equations is the classical model to describe the evolution of isothermal incompressible and Newtonian fluid flows. When in addition the density of the fluid is assumed to be constant, $\rho(x, t) = \rho_0$, the equations reduce to

$$\begin{aligned} \operatorname{div} u &= 0, \\ \rho_0 \left(\frac{\partial u}{\partial t} + (u \cdot \nabla) u \right) &= \nu \Delta u - \nabla p + \rho_0 f \quad \text{in } \Omega_F(t), \quad t > 0, \end{aligned} \quad (3)$$

which are obtained by setting

$$\sigma = \nu(\nabla u + (\nabla u)^T) + \left(\mu - \frac{2\nu}{3}\right) \operatorname{div} u I - pI, \quad (4)$$

in (2). When $\operatorname{div} u = 0$, the expression of σ simplifies. The coefficients $\nu > 0$ and $\mu > 0$ are the viscosity coefficients of the fluid, and $p(x, t)$ is its pressure at the point $x \in \Omega_F(t)$ and at time $t > 0$. This model has to be completed with boundary conditions on $\partial\Omega_F(t)$ and an initial condition at time $t = 0$.

The incompressible Euler equations with constant density are obtained by setting $\nu = 0$ in the above system.

The compressible Navier-Stokes system is obtained by coupling the equation of conservation of mass (1) with the balance of momentum (2), where the tensor σ is defined by (4) and by completing the system with a constitutive law for the pressure.

Control Issues

There are unstable steady states of the Navier-Stokes equations which give rise to interesting control problems (e.g., maximize the lift drag ratio), but cannot be observed in real life because of their unstable nature. In such situations, we would like to maintain the physical model close to an unstable steady state by the action of a

control expressed in feedback form, i.e., as a function depending either on an estimation of the velocity or depending on the velocity itself. The estimation of the velocity of the fluid may be recovered by using some real time measurements. In that case, we speak of a feedback stabilization problem with partial information. Otherwise, if the control is expressed in terms of the velocity itself, we speak of a feedback stabilization problem with full information.

Another interesting issue is to maintain a fluid flow (described by the Navier-Stokes equations) in the neighborhood of a nominal trajectory (not necessarily a steady state) in the presence of perturbations. This is a much more complicated issue which is not yet solved.

In the case of a perturbation in the initial condition of the system (the initial condition at time $t = 0$ is different from the nominal velocity field at time $t = 0$), the exact controllability to the nominal trajectory consists in looking for controls driving the system in finite time to the desired trajectory.

Thus, control issues for fluid flows are those encountered in other fields. However there are specific difficulties which make the corresponding problems challenging. When we deal with the incompressible Navier-Stokes system, the pressure plays the role of a Lagrange multiplier associated with the incompressibility condition. Thus we have to deal with an infinite-dimensional nonlinear differential algebraic system. In the case of a Dirichlet boundary control, the elimination of the pressure, by using the so-called Leray or Helmholtz projector, leads to an unusual form of the corresponding control operator; see Raymond (2006). In the case of an internal control, the estimation of the pressure to prove observability inequalities is also quite tricky; see Fernandez-Cara et al. (2004). From the numerical viewpoint, the approximation of feedback control laws leads to very large size problems, and new strategies have to be found for tackling these issues.

Moreover, the issues that we have described for the incompressible Navier-Stokes equations may be studied for other models like the compressible Navier-Stokes equations and the Euler equations (describing non viscous fluid flows)

both for compressible and incompressible models or even more complicated models.

Feedback Stabilization of Fluid Flows

Let us now describe what are the known results for the incompressible Navier-Stokes equations in 2D or 3D bounded domains, with a control acting locally in a Dirichlet boundary condition. Let us consider a given steady state (u_s, p_s) satisfying the equation

$$- \nu \Delta u_s + (u_s \cdot \nabla) u_s + \nabla p_s = f_s,$$

and $\operatorname{div} u_s = 0$ in Ω_F ,

with some boundary conditions which may be of Dirichlet type or of mixed type (Dirichlet-Neumann-Navier type). For simplicity, we only deal with the case of Dirichlet boundary conditions

$$u_s = g_s \quad \text{on} \quad \partial\Omega_F,$$

where g_s and f_s are time independent functions. In the case $\Omega_F(t) = \Omega_F$, not depending on t , the corresponding instationary model is

$$\begin{aligned} \frac{\partial u}{\partial t} - \nu \Delta u + (u \cdot \nabla) u + \nabla p &= f_s \\ \text{and} \quad \operatorname{div} u &= 0 \text{ in } \Omega_F \times (0, \infty), \\ u &= g_s + \sum_{i=1}^{N_c} f_i(t) g_i, \quad \partial\Omega_F \times (0, \infty), \\ u(0) &= u_0 \quad \text{on } \Omega_F. \end{aligned} \tag{5}$$

In this model, we assume that $u_0 \neq u_s$, g_i are given functions with localized supports in $\partial\Omega_F$ and $f(t) = (f_1(t), \dots, f_{N_c}(t))$ is a finite-dimensional control. Due to the incompressibility condition, the functions g_i have to satisfy

$$\int_{\partial\Omega_F} g_i \cdot n = 0,$$

where n is the unit normal to $\partial\Omega_F$, outward Ω_F .

The stabilization problem, with a prescribed decay rate $-\alpha < 0$, consists in looking for a control f in feedback form, that is, of the form

$$f(t) = K(u(t) - u_s), \tag{6}$$

such that the solution to the Navier-Stokes system (5), with f defined by (6), obeys

$$\|e^{\alpha t}(u(t) - u_s)\|_Z \leq \varphi(\|u_0 - u_s\|_Z),$$

for some norm Z , provided $\|u_0 - u_s\|_Z$ is small enough, and where φ is a nondecreasing function. The mapping K , called the feedback gain, may be chosen linear.

The usual procedure to solve this stabilization problem consists in writing the system satisfied by $u - u_s$, in linearizing this system, and in looking for a feedback control stabilizing this linearized model. The issue is first to study the stabilizability of the linearized model and, when it is stabilizable, to find a stabilizing feedback gain. Among the feedback gains that stabilize the linearized model, we have to find one able to stabilize, at least locally, the nonlinear system too.

The linearized controlled system associated with (5) is

$$\begin{aligned} \frac{\partial v}{\partial t} - \nu \Delta v + (u_s \cdot \nabla) v + (v \cdot \nabla) u_s + \nabla q &= 0 \\ \text{and} \quad \operatorname{div} v &= 0 \text{ in } \Omega_F \times (0, \infty), \\ v &= \sum_{i=1}^{N_c} f_i(t) g_i \text{ on } \partial\Omega_F \times (0, \infty), \\ v(0) &= v_0 = u_0 \text{ on } \Omega_F. \end{aligned} \tag{7}$$

The easiest way for proving the stabilizability of the controlled system (7) is to verify the Hautus criterion. It consists in proving the following unique continuation result. If $(\phi_j, \psi_j, \lambda_j)$ is the solution to the eigenvalue problem

$$\begin{aligned}
& \lambda_j \phi_j - \nu \Delta \phi_j - (u_s \cdot \nabla) \phi_j + (\nabla u_s)^T \phi_j \\
& + \nabla \psi_j = 0 \quad \text{and} \quad \operatorname{div} \phi_j = 0 \text{ in } \Omega_F, \\
& \phi_j = 0 \quad \text{on } \partial \Omega_F, \quad \operatorname{Re} \lambda_j \geq -\alpha,
\end{aligned} \tag{8}$$

and if in addition (ϕ_j, ψ_j) satisfies

$$\int_{\partial \Omega_F} g_i \cdot \sigma(\phi_j, \psi_j) n = 0 \quad \text{for all } 1 \leq i \leq N_c,$$

then $\phi_j = 0$ and $\psi_j = 0$. By using a unique continuation theorem due to Fabre and Lebeau (1996), we can explicitly determine the functions g_i so that this condition is satisfied; see Raymond and Thevenet (2010). For feedback stabilization results of the Navier-Stokes equations in two or three dimensions, we refer to Fursikov (2004), Raymond (2006, 2007), Barbu et al. (2006), Badra (2009), and Vazquez and Krstic (2008).

Controllability to Trajectories of Fluid Flows

If $(\tilde{u}(t), \tilde{p}(t))_{0 \leq t < \infty}$ is a solution to the Navier-Stokes system, the controllability problem to the trajectory $(\tilde{u}(t), \tilde{p}(t))_{0 \leq t < \infty}$, in time $T > 0$, may be rewritten as a null controllability problem satisfied by $(v, q) = (u - \tilde{u}, p - \tilde{p})$. The local null controllability in time $T > 0$ follows from the null controllability of the linearized system and from a fixed point argument. The linearized controlled system is

$$\begin{aligned}
& \frac{\partial v}{\partial t} - \nu \Delta v + (\tilde{u}(t) \cdot \nabla) v \\
& + (v \cdot \nabla) \tilde{u}(t) + \nabla q = 0 \\
& \text{and} \quad \operatorname{div} v = 0 \text{ in } \Omega_F \times (0, T), \\
& v = m_c f \quad \text{on } \partial \Omega_F \times (0, T), \\
& v(0) = v_0 \in L^2(\Omega_F; \mathbb{R}^N), \quad \operatorname{div} v_0 = 0.
\end{aligned} \tag{9}$$

The nonnegative function m_c is used to localize the boundary control f . The control f is assumed to satisfy

$$\int_{\partial \Omega_F} m_c f \cdot n = 0. \tag{10}$$

As for general linear dynamical systems, the null controllability of the linearized system follows from an observability inequality for the solutions to the following adjoint system

$$\begin{aligned}
& -\frac{\partial \phi}{\partial t} - \nu \Delta \phi \\
& - (\tilde{u}(t) \cdot \nabla) \phi + (\nabla \tilde{u}(t))^T \phi + \nabla \psi = 0 \\
& \text{and} \quad \operatorname{div} \phi = 0 \text{ in } \Omega_F \times (0, T), \\
& \phi = 0 \quad \text{on } \partial \Omega_F \times (0, T), \\
& \phi(T) \in L^2(\Omega_F; \mathbb{R}^N), \quad \operatorname{div} \phi(T) = 0.
\end{aligned} \tag{11}$$

Contrarily to the stabilization problem, the null controllability by a control of finite dimension seems to be out of reach, and it will be impossible in general. We look for a control $f \in L^2(\partial \Omega_F; \mathbb{R}^N)$, satisfying (10), driving the solution to system (9) in time T to zero, that is, such that the solution $v_{v_0, f}$ obeys $v_{v_0, f}(T) = 0$. The linearized system (9) is null controllable in time $T > 0$ by a boundary control $f \in L^2(\partial \Omega_F; \mathbb{R}^N)$ obeying (10), if and only if there exists $C > 0$ such that

$$\int_{\Omega_F} |\phi(0)|^2 dx \leq C \int_{\partial \Omega_F} m_c |\sigma(\phi, \psi) n|^2 dx, \tag{12}$$

for all solution (ϕ, ψ) of (11). The observability inequality (12) may be proved by establishing weighted energy estimates called “Carleman type estimates”; see Fernandez-Cara et al. (2004) and Fursikov et al. (1996).

Additional Controllability Results for Other Fluid Flow Models

The null controllability of the 2D incompressible Euler equation has been obtained by J.-M. Coron with the so-called “Return Method” (Coron 1996). See also Coron (2007) for additional references (in particular the 3D case has been treated by O. Glass).

Some null controllability results for the one-dimensional compressible Navier-Stokes equations have been obtained in Ervedoza et al. (2012).

Fluid-Structure Models

Fluid-structure models are obtained by coupling an equation describing the evolution of the fluid flow with an equation describing the evolution of the structure. The coupling comes from the balance of momentum and by writing that, at the fluid-structure interface, the fluid velocity is equal to the displacement velocity of the structure.

The most important difficulty in studying those models comes from the fact that the domain occupied by the fluid at time t evolves and depends on the displacement of the structure. In addition, when the structure is deformable, its evolution is usually written in Lagrangian coordinates, while fluid flows are usually described in Eulerian coordinates.

The structure may be a rigid or a deformable body immersed into the fluid. It may also be a deformable structure located at the boundary of the domain occupied by the fluid.

A Rigid Body Immersed in a Three-Dimensional Incompressible Viscous Fluid

In the case of a 3D rigid body $\Omega_S(t)$ immersed in a fluid flow occupying the domain $\Omega_F(t)$, the motion of the rigid body may be described by the position $h(t) \in \mathbb{R}^3$ of its center of mass and by a matrix of rotation $Q(t) \in \mathbb{R}^{3 \times 3}$. The domain $\Omega_S(t)$ and the flow X_S associated with the motion of the structure obey

$$\begin{aligned} X_S(y, t) &= h(t) + Q(t)Q_0^{-1}(y - h(0)), \\ \text{for } y \in \Omega_S(0) &= \Omega_S, \\ \Omega_S(t) &= X_S(\Omega_S(0), t), \end{aligned} \quad (13)$$

and the matrix $Q(t)$ is related to the angular velocity $\omega : (0, T) \mapsto \mathbb{R}^3$, by the differential equation

$$Q'(t) = \omega(t) \times Q(t), \quad Q(0) = Q_0. \quad (14)$$

We consider the case when the fluid flow satisfies the incompressible Navier-Stokes system

(3) in the domain $\Omega_F(t)$ corresponding to Fig. 1. Denoting by $J(t) \in \mathbb{R}^{3 \times 3}$ the tensor of inertia at time t , and by m the mass of the rigid body, the equations of the structure are obtained by writing the balance of linear and angular momenta

$$\begin{aligned} mh'' &= \int_{\partial\Omega_S(t)} \sigma(u, p)ndx, \\ J\omega' &= J\omega \times \omega + \int_{\partial\Omega_S(t)} (x - h) \times \sigma(u, p)ndx, \\ h(0) &= h_0, \quad h'(0) = h_1, \quad \omega(0) = \omega_0, \end{aligned} \quad (15)$$

where n is the normal to $\partial\Omega_S(t)$ outward $\Omega_F(t)$. The system (3)–(13)–(14)–(15) has to be completed with boundary conditions. At the fluid structure interface, the fluid velocity is equal to the displacement velocity of the rigid solid:

$$u(x, t) = h'(t) + \omega(t) \times (x - h(t)), \quad (16)$$

for all $x \in \partial\Omega_S(t)$, $t > 0$. The exterior boundary of the fluid domain is assumed to be fixed $\Gamma_e = \partial\Omega_F(t) \setminus \partial\Omega_S(t)$. The boundary condition on $\Gamma_e \times (0, T)$ may be of the form

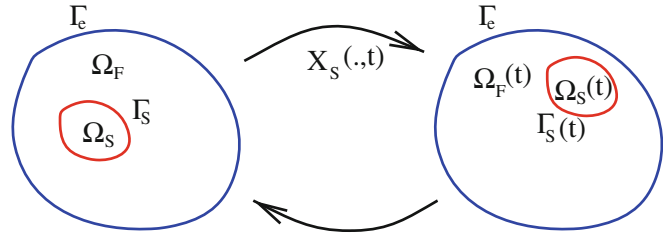
$$u = m_c f \quad \text{on } \Gamma_e \times (0, \infty), \quad (17)$$

with $\int_{\Gamma_e} m_c f \cdot n = 0$, f is a control and m_c a localization function.

An Elastic Beam Located at the Boundary of a Two-Dimensional Domain Filled by an Incompressible Viscous Fluid

When the structure is described by an infinite-dimensional model (a partial differential equation or a system of p.d.e.), there are a few existence results for such systems and mainly existence of weak solutions (Chambolle et al. 2005). But for stabilization and control problems of nonlinear systems, we are usually interested in strong solutions. Let us describe a two-dimensional model in which a one-dimensional structure is located on a flat part $\Gamma_S = (0, L) \times \{y_0\}$ of the boundary of the reference configuration of the fluid domain Ω_F . We assume that the structure is an Euler-

Control of Fluid Flows and Fluid-Structure Models, Fig. 1 Reference and deformed configurations



Bernoulli beam with or without damping. The displacement η of the structure in the direction normal to the boundary Γ_S is described by the partial differential equation

$$\begin{aligned} \eta_{tt} - b\eta_{xx} - c\eta_{txx} + a\eta_{xxxx} &= F, \\ \text{in } \Gamma_S \times (0, \infty), \\ \eta &= 0 \quad \text{and} \quad \eta_x = 0 \quad \text{on } \partial\Gamma_S \times (0, \infty), \\ \eta(0) &= \eta_1^0 \quad \text{and} \quad \eta_t(0) = \eta_2^0 \quad \text{in } \Gamma_S, \end{aligned} \quad (18)$$

where η_x , η_{xx} , and η_{xxxx} stand for the first, the second and the fourth derivative of η with respect to $x \in \Gamma_S$. The other derivatives are defined in a similar way. The coefficients b and c are nonnegative, and $a > 0$. The term $c\eta_{txx}$ is a structural damping term. At time t , the structure occupies the position $\Gamma_S(t) = \{(x, y) \mid x \in (0, L), y = y_0 + \eta(x, t)\}$. When Ω_F is a two-dimensional model, Γ_S is of dimension one, and $\partial\Gamma_S$ is reduced to the two extremities of Γ_S . The momentum balance is obtained by writing that F in (18) is given by $F = -\sqrt{1 + \eta_x^2} \sigma(u, p) \tilde{n} \cdot n$, where $\tilde{n}(x, y)$ is the unit normal at $(x, y) \in \Gamma_S(t)$ to $\Gamma_S(t)$ outward $\Omega_F(t)$, and n is the unit normal to Γ_S outward $\Omega_F(0) = \Omega_F$. If in addition, a control f acts as a distributed control in the beam equation, we shall have

$$F = -\sqrt{1 + \eta_x^2} \sigma(u, p) \tilde{n} \cdot n + f. \quad (19)$$

The equality of velocities on $\Gamma_S(t)$ reads as

$$\begin{aligned} u(x, y_0 + \eta(x, t)) &= (0, \eta_t(x, t)), \\ x &\in (0, L), \quad t > 0. \end{aligned} \quad (20)$$

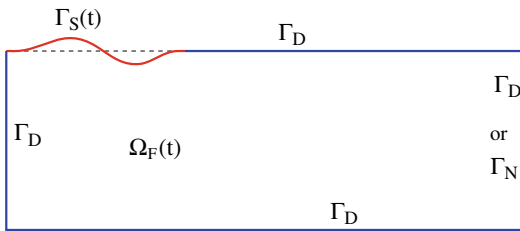
Control of Fluid-Structure Models

To control or to stabilize fluid-structure models, the control may act either in the fluid equation or in the structure equation or in both equations. There are a very few controllability and stabilization results for systems coupling the incompressible Navier-Stokes system with a structure equation. We state below two of those results. Some other results are obtained for simplified one-dimensional models coupling the viscous Burgers equation with the motion of a mass; see Badra and Takahashi (2013) and the references therein.

We have also to mention here recent papers on control problems for systems coupling quasi-stationary Stokes equations with the motion of deformable bodies, modeling micro-organism swimmers at low Reynolds number; see Alouges et al. (2008).

Null Controllability of the Navier-Stokes System Coupled with the Motion of a Rigid Body

The system coupling the incompressible Navier-Stokes system (3) in the domain drawn in Fig. 1, with the motion of a rigid body described by (13)–(14)–(15)–(16), with the boundary control (17), is null controllable locally in a neighborhood of 0. Before linearizing the system in a neighborhood of 0, the fluid equations have to be rewritten in Lagrangian coordinates, that is, in the cylindrical domain $\Omega_F \times (0, \infty)$. The linearized system is the Stokes system coupled with a system of ordinary differential equations. The proof of this null controllability result relies on a Carleman estimate for the adjoint system, see, e.g., Boulakia and Guerrero (2013).



Control of Fluid Flows and Fluid-Structure Models,
Fig. 2 Channel flow with an elastic wall

Feedback Stabilization of the Navier-Stokes System Coupled with a Beam Equation

The system coupling the incompressible Navier-Stokes system (3) in the domain drawn in Fig. 2, with a beam equation (18)–(19)–(20) can be locally stabilized with any prescribed exponential decay rate $-\alpha < 0$, by a feedback control f acting in (18) via (19); see Raymond (2010). The proof consists in showing that the infinitesimal generator of the linearized model is an analytic semigroup (when $c > 0$), that its resolvent is compact, and that the Hautus criterion is satisfied.

When the control acts in the fluid equation, the system coupling (3) in the domain drawn in Fig. 2, with the beam equation (18)–(19)–(20) can be stabilized when $c > 0$. To the best of our knowledge, there is no null controllability result for such systems, even with controls acting both in the structure and fluid equations. The case where the beam equation is approximated by a finite-dimensional model is studied in Lequeurre (2013).

Cross-References

- Boundary Control of 1-D Hyperbolic Systems
- Boundary Control of Korteweg-de Vries and Kuramoto-Sivashinsky PDEs

Bibliography

Alouges F, DeSimone A, Lefebvre A (2008) Optimal strokes for low Reynolds number swimmers: an example. *J Nonlinear Sci* 18:277–302

- Badra M (2009) Lyapunov function and local feedback boundary stabilization of the Navier-Stokes equations. *SIAM J Control Optim* 48:1797–1830
- Badra M, Takahashi T (2013) Feedback stabilization of a simplified 1d fluid-particle system. *Ann Inst H Poincaré Anal Non Linéaire*. <https://doi.org/10.1016/j.anihpc.2013.03.009>
- Barbu V, Lasiecka I, Triggiani R (2006) Tangential boundary stabilization of Navier-Stokes equations. *Memoirs of the American Mathematical Society*, vol 181. American Mathematical Society, Providence
- Boulakia M, Guerrero S (2013) Local null controllability of a fluid-solid interaction problem in dimension 3. *J EMS* 15:825–856
- Chambolle A, Desjardins B, Esteban MJ, Grandmont C (2005) Existence of weak solutions for unsteady fluid-plate interaction problem. *JMFM* 7:368–404
- Coron J-M (1996) On the controllability of 2-D incompressible perfect fluids. *J Math Pures Appl* (9) 75: 155–188
- Coron J-M (2007) Control and nonlinearity. American Mathematical Society, Providence
- Ervedoza S, Glass O, Guerrero S, Puel J-P (2012) Local exact controllability for the one-dimensional compressible Navier-Stokes equation. *Arch Ration Mech Anal* 206:189–238
- Fabre C, Lebeau G (1996) Prolongement unique des solutions de l'équation de Stokes. *Commun Partial Differ Equs* 21:573–596
- Fernandez-Cara E, Guerrero S, Imanuvilov OY, Puel J-P (2004) Local exact controllability of the Navier-Stokes system. *J Math Pures Appl* 83:1501–1542
- Fursikov AV (2004) Stabilization for the 3D Navier-Stokes system by feedback boundary control. *Partial differential equations and applications. Discrete Contin Dyn Syst* 10:289–314
- Fursikov AV, Imanuvilov OY (1996) Controllability of evolution equations. *Lecture Notes Series*, 34. Seoul National University, Research Institute of Mathematics, Global Analysis Research Center, Seoul
- Lequeurre J (2013) Null controllability of a fluid-structure system. *SIAM J Control Optim* 51:1841–1872
- Raymond J-P (2006) Feedback boundary stabilization of the two dimensional Navier-Stokes equations. *SIAM J Control Optim* 45:790–828
- Raymond J-P (2007) Feedback boundary stabilization of the three-dimensional incompressible Navier-Stokes equations. *J Math Pures Appl* 87:627–669
- Raymond J-P (2010) Feedback stabilization of a fluid-structure model. *SIAM J Control and Optim* 48(8):5398–5443
- Raymond J-P, Thevenet L (2010) Boundary feedback stabilization of the two dimensional Navier-Stokes equations with finite dimensional controllers. *DCDS-A* 27:1159–1187
- Vazquez R, Krstic M (2008) Control of turbulent and magnetohydrodynamic channel flows: boundary Stabilization and estimation. Birkhäuser, Boston

Control of Left Ventricular Assist Devices

Marwan A. Simaan

Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, USA

Keywords

Advanced heart failure · Heart transplantation · Left ventricle · LVAD physiological controller · Rotary blood pump · Automatic pump speed controller

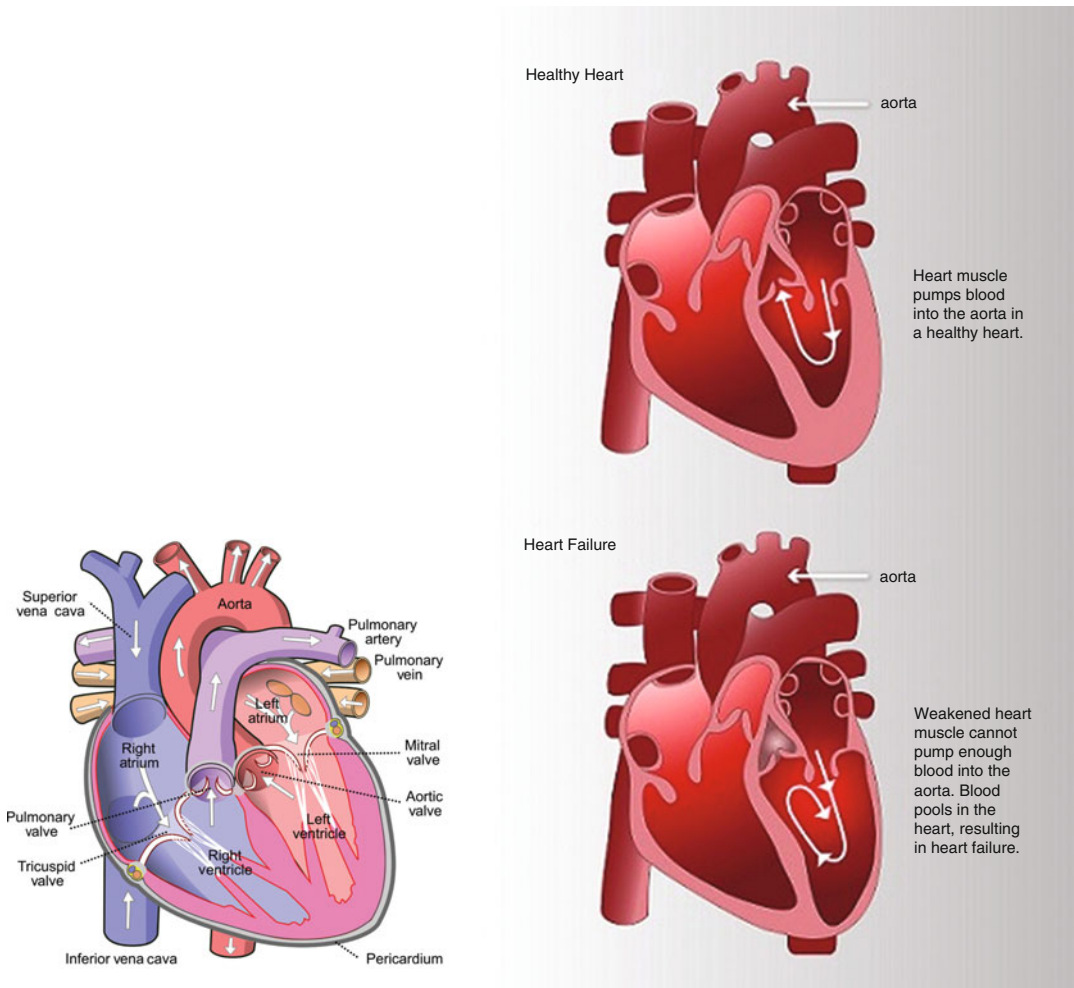
Abstract

Heart failure is a disease that happens when the heart cannot pump enough oxygenated blood into the circulatory system to support other organs in the body. A left ventricular assist device (or LVAD) is a mechanical pump that may be implanted in patients who suffer from advanced heart failure to assist their heart in its pumping function. The pump is connected from the left ventricle to the aorta and forms a parallel path of blood flow into the circulatory system. The latest generation of these devices consists of rotary pumps, which are generally much smaller, lighter, and quieter than the first-generation pulsatile-type pumps. The LVAD is controlled by varying the pump rotor speed to adjust the amount of blood flow contributed by the device. If the patient is in a healthcare facility, the rotor speed can be adjusted manually by a trained clinician. An important limitation facing the increased use of these devices is the desire to allow the patient to return home and resume a normal lifestyle. This limitation may be overcome if a physiological controller can be developed to automatically adjust the pump speed based on the patient's need for blood flow. Although several possible experimental controllers have recently been developed and tested in laboratory settings, the current practice remains limited to the pump speed being adjusted by the physician and fixed at a constant setting before allowing the patient to leave the healthcare facility. The development of a physiological controller for the continuous flow rotary LVAD remains an important challenge that has occupied LVAD researchers and engineers for the past several decades.

Introduction

According to the latest statistics, more than 5.7 million adults in the United States suffer from heart failure, and more than 650,000 new cases of heart failure are diagnosed every year (Mozzafarian et al. 2016). This disease remains the number one killer of Americans every year, in spite of remarkable progress in medical therapies over the past several decades. Heart failure occurs when the heart muscle becomes unable to generate the necessary force to push enough oxygenated blood from the left ventricle through the aortic valve into the circulatory system (Fig. 1). In its initial stages, the disease is often managed through lifestyle changes and medications (Yancy et al. 2013). However, as the disease progresses, more advanced treatments become necessary. The continuous flow left ventricular assist device (LVAD), sometimes also referred to as an implantable rotary blood pump (IRBP), is an option for treating patients with advanced heart failure disease.

The LVAD is a mechanical pump whose function is to assist the heart in pumping blood. It consists of an inflow cannula, the pump, an outflow cannula, a controller, and a battery pack (Fig. 2). The latest generation of these devices (For example the HeartMate 3, the HeartWare, and the Jarvik 2000) consists of rotary pumps, which are generally much smaller, lighter, and quieter than the first-generation pulsatile-type pumps (Prinzing et al. 2016). The pump is surgically implanted into the patient's chest with the inflow cannula attached to the left ventricle and the outlet cannula inserted into the ascending aorta. Control of the pump speed is performed by the controller unit which is a small box located outside the patient's body. The LVAD is powered by the batteries connected to the controller unit. Blood flows from the left ventricle through the



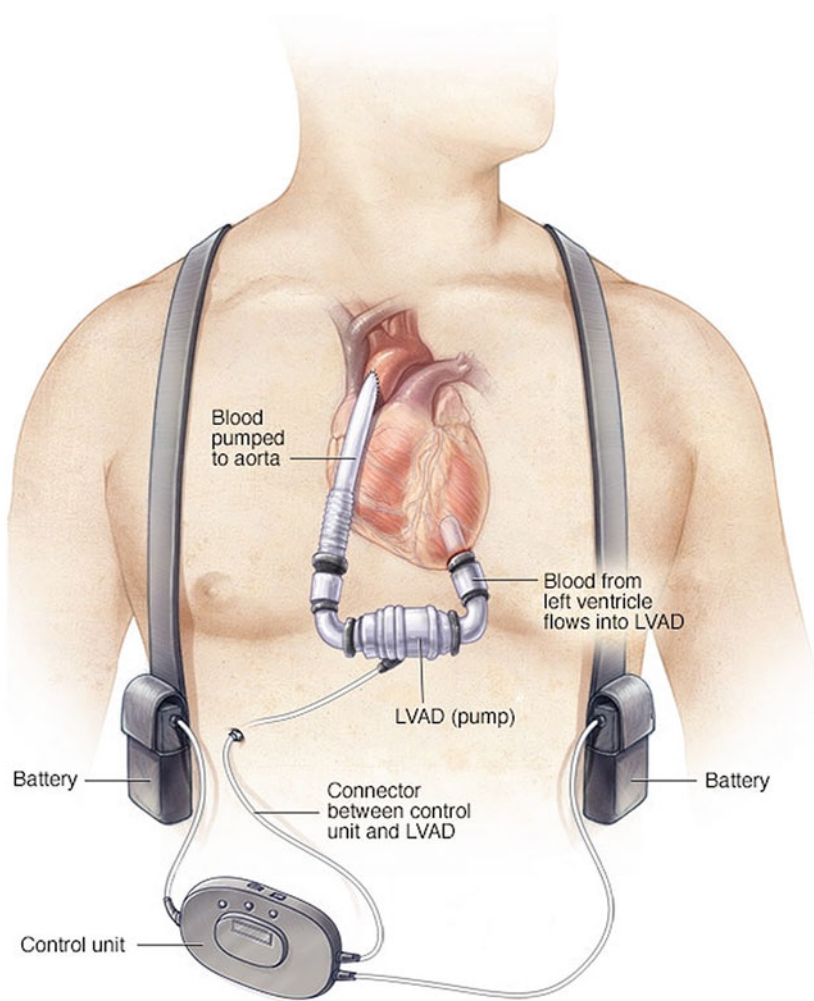
Control of Left Ventricular Assist Devices, Fig. 1 (Left) Cross section of the heart. <https://en.wikipedia.org/wiki/Heart>. (Right) Healthy heart and heart failure.

https://www.cdc.gov/dhbsp/data_statistics/fact_sheets/fs_heart_failure.htm

pump to the aorta forming a parallel path to the natural flow from the left ventricle to the aorta through the aortic valve. The LVAD is controlled by varying the pump's rotational speed to adjust the amount of blood flow through the device. Some LVADs are capable of producing blood flow of up to 10 L/min, which can provide complete circulatory support. In normal LVAD operation, however, the heart and LVAD work together cooperatively, providing the amount of blood needed by the body based on the patient's activity level, as well as his/her physiological and emotional conditions. The heart provides blood during every heartbeat by pumping from the left

ventricle through the aortic valve into the circulatory system, while the LVAD provides blood continuously from the left ventricle through the pump into the aorta and the circulatory system.

Heart transplantation is a viable option of care for heart failure patients who fail to respond to lifestyle changes and medication. At any given time, around 3,500 to 4,000 patients in the United States are waiting for a heart transplant, and many will die before receiving a heart. The wait time for a heart transplantation can be several months and often exceeds 300 days on the average. The LVAD is typically used as a bridge to transplantation for patients on the transplant



© MAYO FOUNDATION FOR MEDICAL EDUCATION AND RESEARCH. ALL RIGHTS RESERVED.

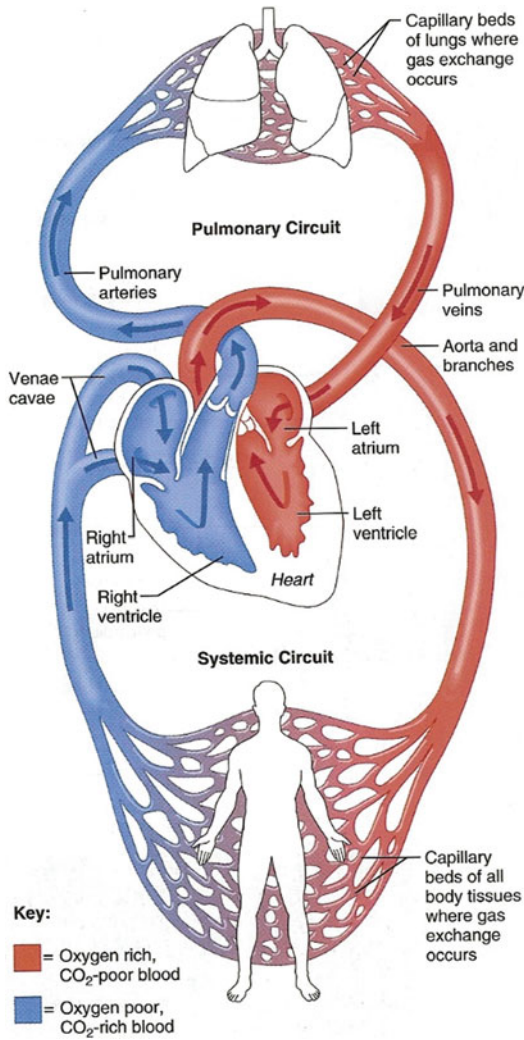
Control of Left Ventricular Assist Devices, Fig. 2 Schematic of an adult patient with a left ventricular assist device. (Used with permission of Mayo Foundation for Medical Education and Research, all rights reserved)

waiting list. Its main role is to support, or even in some cases substitute, the function of the natural heart while the patient is waiting for a donor heart (Poirier 1997; Frazier and Myers 1999; Olsen 2000). As a last resort for patients with refractory heart failure who are not eligible for heart transplantation, the LVAD can be implanted permanently as a destination therapy. In recent years, studies have also shown that for a very small subset of patients, the LVAD is able to restore the muscles of the failing heart, eliminating the need for a transplant (Simon et al. 2005;

Jakovljevic et al. 2017). An increasing number of these bridge-to-recovery patients will have their LVADs successfully explanted following sufficient recovery of the heart's ability to pump on its own.

The Left Ventricle and Systemic Circulation Model

The heart pumps blood through two different circulation paths as illustrated in Fig. 3 . The left



Control of Left Ventricular Assist Devices,
Fig. 3 Diagram of the systemic and pulmonary
 circulatory systems (Silverthorn 2000)

ventricle pumps oxygen-rich blood to the rest of the body through the systemic circulation path, and the right ventricle pumps oxygen-depleted blood to the lungs through the pulmonary circulation path (Marieb 1994). Due to the different nature of the two circulation paths, each side of the heart has a different workload. The systemic circulation is much longer than the pulmonary one and encounters considerable more resistance. To handle the longer path, the left ventricle is more elastic than the right ventricle, giving it

the strength to pump blood at a higher pressure (Marieb 1994).

The ability of the left ventricle to pump blood is expressed in terms of its elastance function $E(t)$. The elastance function is periodic of period $t_c = 60/HR$ where HR is the heart rate in beats per minute (bpm). During each cardiac cycle t_c , $E(t)$ relates the left ventricular pressure $LVP(t)$ to the left ventricular volume $LVV(t)$ according to the expression (Suga and Sagawa 1974):

$$E(t) = \frac{LVP(t)}{LVV(t) - V_0} \quad (1)$$

where V_0 is a reference volume which corresponds to the theoretical volume in the ventricle at zero pressure. Several mathematical expressions have been derived to model the elastance of a healthy heart over one cardiac cycle. In this article, we use the standard expression (Stergiopoulos et al. 1996):

$$E(t) = (E_{\max} - E_{\min})E_n(t_n) + E_{\min} \quad (2)$$

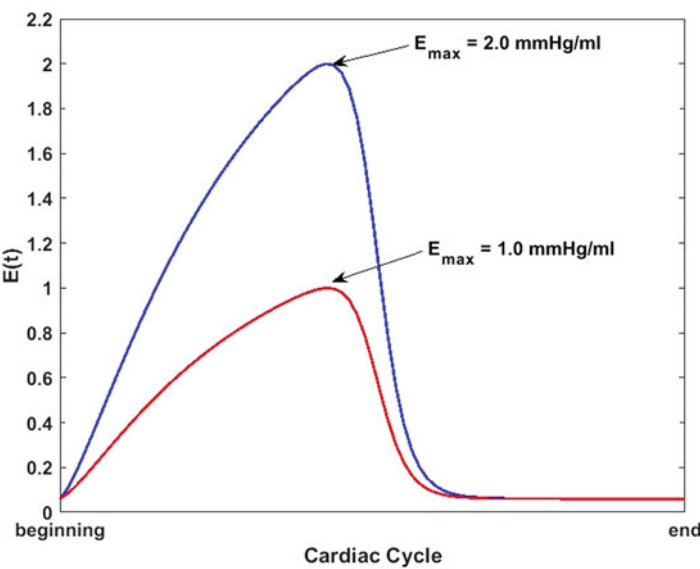
where $E_n(t_n)$ is the so-called double hill function represented by the expression:

$$E_n(t_n) = 1.55 * \left[\frac{\left(\frac{t_n}{0.7}\right)^{1.9}}{1 + \left(\frac{t_n}{0.7}\right)^{1.9}} \right] * \left[\frac{1}{1 + \left(\frac{t_n}{1.17}\right)^{21.9}} \right] \quad (3)$$

In the above expression, $E_n(t_n)$ is the normalized elastance and $t_n = t/(0.2 + 0.15t_c)$. The constants $E_{\max}$ and $E_{\min}$ in (2) are related to the end-systolic pressure-volume relationship (ESPVR) and the end-diastolic pressure-volume relationship (EDPVR), respectively. Figure 4 shows a plot of $E(t)$ for a healthy heart with $E_{\max} = 2$ mmHg/ml, $E_{\min} = 0.06$ mmHg/ml, and a heart rate of 60 bpm (cardiac cycle $t_c = 1$ s). For a patient with heart failure disease, the elastance expression (2) is modified according to $E_\delta(t) = \delta E(t)$ where δ is a factor such that $0 < \delta \leq 1$ and is used to represent the degree of heart failure. The value of $\delta = 1$ represents a healthy ventricle, and the smaller the value of δ , the more severe is the disease. A correlation between $E_{\max}$ and the degrees of heart failure, as defined according to the New York Heart Association

Control of Left Ventricular Assist Devices,

Fig. 4 Ventricular elastance function $E(t)$ (Blue is for a healthy heart and red is for a sick heart with $\delta = 0.5$)



Control of Left Ventricular Assist Devices, Table 1 Map of NYHA classification and condition of heart failure^a

Condition of heart	$\delta = E_{\max}/2.0$	$E_{\max}$ (mmHg/ml)	NYHA class
Normal	1.0	2.0	–
Mild dysfunction	0.6–1.0	1.2–2.0	I
Moderate dysfunction	0.4–0.6	0.8–1.2	II
Severe dysfunction	<0.4	<0.8	III–IV

^a<https://www.acc.org/tools-and-practice-support/clinicaltoolkits/~media/1E4D8F9B69D14F55821BFE642FBFA221.ashx>

(NYHA) classification, is noted in Table 1 (Levin et al. 1994).

The heart and its circulation paths are a very complex system that is very difficult to model mathematically. Numerous models of varying degrees of complexity have been developed over the past 25 years or so. An eighth-order hybrid model of the heart, which subdivides the human circulatory system into a number of lumped parameter blocks that include the pulmonary arterial and venous systems as well as both ventricles, has been developed in Giridharan et al. (2002). A different eleventh-order model which includes the resistance of peripheral blood vessels has been developed in Wu et al. (2007). Our interest in this article is a dynamic state-space-type model of the systemic circulation system supported by a rotary LVAD. The model

should be comprehensive enough to capture the important relationships that characterize the patient’s hemodynamic variables and LVAD blood flow while at the same time simple enough to be mathematically tractable and useful for control and optimization purposes. A fifth-order lumped parameter, time-varying, nonlinear state-space dynamic model for the left ventricle supported by a first-order generic continuous flow rotary LVAD, has been developed and validated in Simaan et al. (2009). The model consists of five state variables describing the hemodynamics of the left ventricle and one state variable describing the blood flow through the LVAD. The six state variables in the model are summarized in Table 2, and their relationships over one cardiac cycle are governed by the time-varying nonlinear matrix differential equation (Simaan et al. 2009):

$$\begin{aligned}
\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \\ \frac{dx_4}{dt} \\ \frac{dx_5}{dt} \\ \frac{dx_6}{dt} \end{bmatrix} &= \begin{bmatrix} \frac{1}{E(t)} \frac{dE(t)}{dt} & 0 & 0 & 0 & 0 & -E(t) \\ 0 & \frac{-1}{R_S C_R} & \frac{1}{R_S C_R} & 0 & 0 & 0 \\ 0 & \frac{1}{R_S C_R} & \frac{-1}{R_S C_R} & 0 & \frac{1}{C_S} & 0 \\ 0 & 0 & 0 & 0 & \frac{-1}{C_A} & \frac{1}{C_A} \\ 0 & 0 & \frac{-1}{L_S} & \frac{1}{L_S} & \frac{-R_C}{L_S} & 0 \\ \frac{1}{L^*} & 0 & 0 & \frac{-1}{L^*} & 0 & \frac{-R^*}{L^*} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \\
&+ \begin{bmatrix} E(t) - E(t) \\ \frac{-1}{C_R} & 0 \\ 0 & 0 \\ 0 & \frac{-1}{C_A} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{R_M} r(x_2 - x_1) \\ \frac{1}{R_A} r(x_1 - x_4) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{-\beta}{L^*} \end{bmatrix} u(t) \quad (4)
\end{aligned}$$

In the above expression, the control variable is $u(t) = \omega^2(t)$ where $\omega(t)$ is the rotational speed of the LVAD in RPM. The entries in the various matrices are the model parameters described in Table 3. The first five rows of (4) are parameters associated with the left ventricle, and in the 6th row are parameters associated with the LVAD. The terms R^* and L^* in the 6th row represent the total resistance and total inductance of the LVAD system (pump + cannulas):

$$R^* = R_i + R_P + R_o + R_{su} \quad (5a)$$

$$L^* = L_i + L_P + L_o \quad (5b)$$

The nonlinear time-varying resistance R_{su} in (5a) is included in the model to characterize a phenomenon called suction. It has the form:

$$R_{su} = \begin{cases} 0 & \text{if } x_1(t) > \bar{x}_1 \\ \alpha(x_1(t) - \bar{x}_1) & \text{if } x_1(t) \leq \bar{x}_1 \end{cases} \quad (6)$$

R_{su} is zero when the pump is operating at normal speeds and is activated when $LVP(t)$ (i.e., state variable $x_1(t)$) becomes less than a pre-determined small threshold $\bar{x}_1$, a condition that occurs at high pump speeds. The value of R_{su} when activated increases linearly as a function of the difference between $x_1(t)$ and $\bar{x}_1$. This increase in R_{su} represents a phenomenon called *ventricular suction* which occurs when the pump is attempting to draw more blood from the ventricle than available. The parameter α in (6) is a cannula-dependent scaling factor. In the model, the values used for the suction parameters are $\alpha = -3.5$ s/ml and $\bar{x}_1 = 1$ mmHg.

The term $r(x)$ in (4) represents the ramp function:

$$r(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (7)$$

It is used to model the status of the aortic and mitral valves during each cardiac cycle as summarized in Table 4. Note that the condition $x_2 < x_1$ and $x_1 < x_4$ occurs twice and represents the phase when the ventricle is undergoing isovolumic contraction prior to ejection

Control of Left Ventricular Assist Devices, Table 2 State variables in the cardiovascular-LVAD model

Variables	Name	Physiological meaning (units)
$x_1(t)$	LVP(t)	Left ventricular pressure (mmHg)
$x_2(t)$	LAP(t)	Left atrial pressure (mmHg)
$x_3(t)$	AP(t)	Arterial pressure (mmHg)
$x_4(t)$	AoP(t)	Aortic pressure (mmHg)
$x_5(t)$	$Q_T(t)$	Total flow (ml/s)
$x_6(t)$	$Q_P(t)$	Pump flow (ml/s)

Control of Left Ventricular Assist Devices, Table 3 Parameter values in the cardiovascular-LVAD model (4)

Parameters	Value	Physiological meaning
Resistances (mmHg.s/ml)		
R_S	1.0000	Systemic vascular resistance (SVR)
R_M	0.0050	Mitral valve resistance
R_A	0.0010	Aortic valve resistance
R_C	0.0398	Characteristic resistance
R_i	0.0677	Resistance of LVAD inflow cannula
R_P	0.1707	Resistance of LVAD pump
R_o	0.0677	Resistance of LVAD outflow cannula
R_{su}	Refer to Eq. (6) for value	Suction resistance
Elastance (mmHg/ml)	TIME-VARYING	Left ventricle elastance
$E(t)$	Eq. (2), Fig. 4	
Compliances (ml/mmHg)		
C_R	4.4000	Left atrial compliance
C_S	1.3300	Systemic compliance
C_A	0.0800	Aortic compliance
Inertances (mmHg.s ² /ml)		
L_S	0.0005	Inertance of blood in aorta
L_i	0.0127	Inertance of inflow cannula
L_P	0.0218	Pump inertance
L_o	0.0127	Inertance of outflow cannula
Other pump parameter	$\beta = 9.9025 \cdot 10^{-7} \text{ mmHg}/(\text{rpm})^2$	Pump-dependent constant

Control of Left Ventricular Assist Devices, Table 4 Phases of the left ventricle within the cardiac cycle

Conditions of x_1 , x_2 , and x_4 in (4)		Valves status		Phases of the left ventricle
		Mitral	Aortic	
$x_2 < x_1$	$x_1 < x_4$	Closed	Closed	Isovolumic contraction
$x_2 < x_1$	$x_1 < x_4$	Closed	Open	Ejection
$x_2 < x_1$	$x_1 < x_4$	Closed	Closed	Isovolumic relaxation
$x_2 > x_1$	$x_1 < x_4$	Open	Closed	Filling

and isovolumic relaxation after ejection and prior to filling. Finally, the condition $x_2 > x_1$ and $x_1 > x_4$ is not feasible because it represents both mitral and aortic valves being simultaneously open. The four different phases of operation of the left ventricle over four consecutive time intervals within each cardiac cycle are outlined in Table 4. The overall model (4) over each cardiac cycle therefore consists of three different differential equations consecutively representing the four phases of the cardiac cycle: the filling phase, followed by isovolumic contraction, followed by the ejection phase, and followed again by isovolumic relaxation.

Control of the LVAD

Irrespective of how the LVAD is used, whether as bridge to transplantation, destination therapy, or bridge to recovery, its speed needs to be properly controlled for the safety and comfort of the patient. The total amount of blood pumped by the LVAD into the circulatory system (state variable x_5) depends on the LVAD rotational speed ω , which is directly dependent on the power P_E supplied to the LVAD. The relationship between the LVAD speed ω and power P_E is given by the nonlinear expression (Fargallah et al. 2011; Fargallah and Simaan 2013):

$$\omega = \sqrt{\frac{\gamma P_E}{x_6}} \quad (8)$$

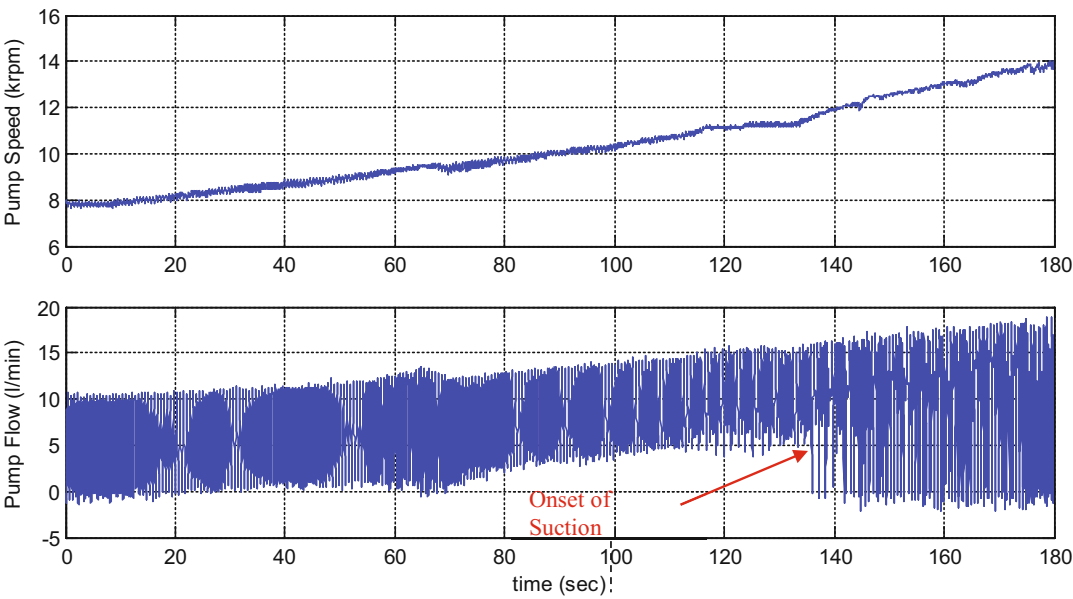
where $\gamma = 7.5688 \times 10^9$ and x_6 is the rate of flow in the LVAD. The higher the power supplied to the LVAD, the higher its speed will be, and the more blood will be pumped into the circulatory system. An important engineering challenge facing the increased use of this device is the development of an appropriate controller that automatically adjusts its rotational speed so that the amount of blood pumped by the LVAD is sufficient to meet the patient's blood needs at all times and as the patient's activity level changes. Patients are typically either sleeping or engaged in some type of activity (walking or exercising) which requires different levels of cardiac support. While the heart may be contributing a part of this support through the aortic valve, the LVAD controller should be able to detect the need for additional support and adjust the pump speed accordingly. Unfortunately, such a controller does not exist at this time. The development of such a controller remains an important challenge that has occupied LVAD researchers for the past several decades. At this point, we know that in addition to being robust and reliable, the controller must ensure that the speed ω remains within two patient-dependent extremes, ω_L and ω_H , at all times:

- (a) At the lower end, if the speed is too low ($\omega < \omega_L$), blood may regurgitate back from the aorta to the left ventricle through the pump resulting in what is known as *backflow*, a condition that causes congestion of the pulmonary circulation.
- (b) At the upper end, if the pump speed is too high ($\omega > \omega_H$), the pump will attempt to draw more blood from the ventricle than available. This may cause collapse of the ventricle resulting in a dangerous phenomenon called *ventricular suction*. The occurrence of ventricular suction must be detected quickly and the pump speed reduced back to the normal range $\omega \in [\omega_L, \omega_H]$ before the heart muscle is permanently

damaged. Several algorithms specifically for detecting ventricular suction have been developed over the past 20 years (Yuhki et al. 1999; Vollkron et al. 2006; Simaan et al. 2009; Wang and Simaan 2013; Wang et al. 2015, and others). Figure 5 shows a plot of an LVAD flow signal measured in an in vivo study (Simaan et al. 2009) in which the LVAD speed is increased linearly until suction is reached. This figure shows the drastic change in the characteristics of the LVAD flow signal before and after the onset of suction.

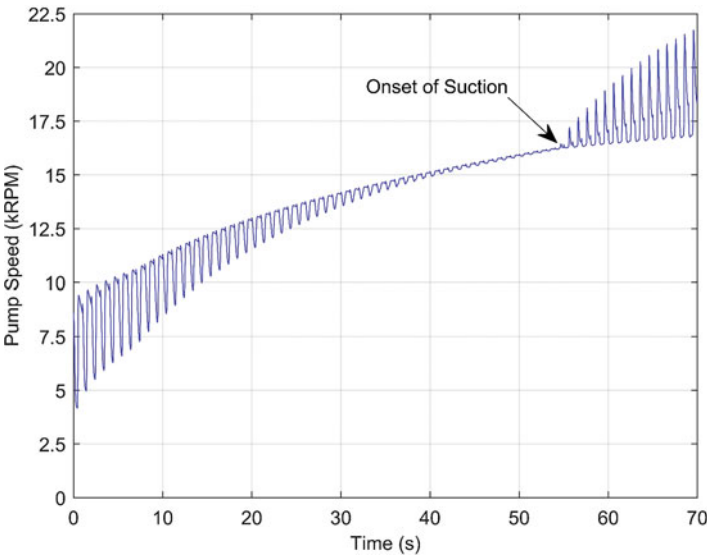
While avoiding these two extremes, the LVAD controller should be able to adjust the speed continuously to adapt to the daily activities and physiological and emotional changes of the patient. In general, controlling the LVAD speed ω is done by varying the power P_E supplied to the LVAD. Figure 6 shows a plot of ω for values of P_E increasing from low to high values. Three important observations can be derived from this plot. First, as expected the relationship between ω and P_E is highly nonlinear. Second, ω has a superposed oscillatory component that has the same pulsatility as the heart rate (60 bpm in this case) which indicates that the beating of the heart seems to modulate the pump speed around its mean value. This is a very interesting and an extremely important phenomenon that has recently been observed in data obtained through clinical studies of intensive care patients implanted with LVADs (Mason et al. 2008). Third, the amplitude of the oscillatory component in ω decreases as P_E increases up to a point when it becomes close to zero when a breakdown occurs. From this point on, the amplitude increases as the pump power is increased. The value of ω at the breakpoint corresponds to the onset of ventricular suction which represents the high end ω_H of the normal operating range of the LVAD.

The controller's function is to regulate the amount of blood pumped by the LVAD into the circulatory system in order to meet the body's requirements for cardiac output (CO) and mean arterial pressure (MAP) (Olsen 2000; Boston



Control of Left Ventricular Assist Devices, Fig. 5 In vivo LVAD flow data as LVAD pump speed is increased (Obtained from animal implanted with rotary blood pump)

Control of Left Ventricular Assist Devices, Fig. 6 Pump speed signal ω as a function of time derived from a linearly increasing pump power P_E



et al. 2003; Shima et al. 1992; Marieb 1994). The eventual goal of the LVAD controller is to enable the patient to leave the healthcare facility and return home to a normal lifestyle. The development of an LVAD controller that is capable of accomplishing this goal is a very difficult problem that has challenged the LVAD research community for more than two decades.

The current practice for most eligible patients is to be sent home with an LVAD running at a fixed constant speed, which cannot be changed manually. This “constant speed control” is determined and set by physicians at the time of discharge from the healthcare facility. The rationale is that the LVAD, continuously running at the set speed, together with the native heart will be

able to provide sufficient blood flow for a limited range of patient movements and activities. While this is not optimal, it is the only viable option available at this time. The hope is that in the near future, a physiological feedback controller based on modern optimal control theory can be developed so that sensor measurements of the left ventricle hemodynamic variables can be the main drivers in deciding how the pump speed should be automatically adjusted for optimal operation of the LVAD in meeting the patient's demands of blood flow.

In the meantime, several conceptual physiological controllers have been developed over the past 20 years or so and tested either in a laboratory using a mock circulatory loop or by computer simulation (for a review of these, see Alomari et al. 2013). Some of these were based on noninvasive data such as estimates of the pump flow rate, while others were based on invasive data such as measurements of some hemodynamic variables (state variables x_1 , x_2 , and/or x_6). Unfortunately, until now none of these have been implemented in a clinical environment. Several of these controllers have been tested and compared (Petrou et al. 2018). A selected summary is given below:

1. Inlet-outlet pressure controller: This controller developed by Bullister et al. (2002) consists of a cascaded arrangement in which measured aortic pressure and heart rate are used to define a set point that corresponds to the minimum of the left ventricular diastolic pressure. This set point is then used to control the pump speed based on the measured minimum left ventricular diastolic pressure.
2. Minimum pump flow controller: This feedback controller developed by Simaan et al. (2009) adjusts the pump speed based on the slope of the minimum pump flow signal. The objective of the controller is to increase the speed until the envelope of the minimum pump flow signal reaches an extreme point indicating the onset of suction and then maintain it below that level afterward.
3. Afterload-impedance controller: This controller developed by Moscato et al. (2010) uses the left ventricular pressure signal to calculate a reference mean pump flow. A proportional-integral controller is then used to control the pump speed by matching the pump flow to the reference pump flow.
4. Preload responsive speed controller: This controller developed by Ochsner et al. (2013) uses measurement of the end-diastolic volume to calculate the desired power output of the LVAD which is then used to determine the pump speed.
5. Pump flow and end-diastolic pressure controller: This controller developed by Mansouri et al. (2015) consists of a cascaded control loop that uses the end-diastolic pressure to compute a desired pump flow. A proportional-integral controller is then used to control the pump speed to match the pump flow to the desired pump flow.
6. A suction prevention and physiological control algorithm was developed by Wang et al. (2015). This algorithm uses two gain-scheduled, proportional-integral controllers that maintain a differential pump speed (ΔRPM) above a user-defined threshold to prevent left ventricular suction while maintaining an average reference differential pressure between the left ventricle and aorta to provide physiologic perfusion.

Summary and Future Directions

The rotary left ventricular assist device has become one of the most viable long-term treatment option for patients awaiting heart transplantation and patients with refractory heart failure. More than 22,000 devices have been implanted to date in the United States, and more than 2,500 new implants occur annually (Kirklin et al. 2017). Many of the recipient patients are confined to a healthcare facility, and a small number have been allowed to go home with an LVAD running at a set constant speed. Many more patients could be allowed to go home if a physiological controller for the LVAD speed were available. Without such a controller, many LVAD patients remain confined to a hospital for

months awaiting a donor heart, an option that is extremely costly and may not be necessary.

Several factors need to be overcome before a physiological controller becomes feasible in a clinical setting. The controller must rely on sensor measurements of hemodynamic variables to detect changes in the body's demand for blood flow as the patient's activity level changes and to close the loop by adjusting the LVAD speed accordingly. The controller must be patient-adaptive, robust, and reliable. To this date, implanting additional sensors in the circulatory system is not recommended and is highly undesirable due to the low reliability and high measurement errors of currently available sensors. Another reason is the high possibility of thrombus formation around the sensors. Estimating the hemodynamic variables, on the other hand, has also been unsuccessful so far due to the lack of robustness and accuracy of the estimates.

Ever since the late 1990s, when the LVAD technology evolved from the pulsatile to the rotary type pumps, one of the most important engineering problems that has preoccupied the LVAD research community has been the development of a physiological controller for the rotational speed of the LVAD. It is expected that this same problem will remain a central focus of the LVAD research activity and innovation in the years to come.

Cross-References

- [Automated Anesthesia Systems](#)
- [Control of Anemia in Hemodialysis Patients](#)

Bibliography

- Alomari AHH, Savkin AV, Stevens M, et al (2013) Developments in control systems for rotary left ventricular assist devices for heart failure patients: a review. *Physiol Meas* 34:R1–R27
- Antaki JF, Boston JR, Simaan MA (2003) Control of heart assist devices. In: *Proceedings of IEEE conference on decision and control*, Maui, 9–12 Dec, pp 4084–4089
- Baloe LA, Liu D, Boston JR, Simaan MA, Antaki JF (2000) Control of rotary heart assist devices. In: *Proceedings of American control conference*, Chicago, 28–30 June, pp 2982–2986
- Boston JR, Simaan MA, Antaki JF, Yu Y-C, Choi S (1998) Intelligent control design of heart assist devices. In: *Proceedings of the IEEE 1998 international symposium on intelligent control, computational intelligence in robotics and automation, and intelligent systems and semiotics: a joint conference on the science and technology of intelligent systems*. National Institute of Standards and Technology, Gaithersburg, 14–17 Sept, pp 497–502
- Boston JR, Baloe LA, Liu D, Simaan MA, Choi S, Antaki JF (2000a) Combination of data approaches to heuristic control and fault detection. In: *Proceedings of the IEEE conference on control applications and international symposium on computer-aided control systems design*, Anchorage, 25–27 Sept, pp 98–103
- Boston JR, Simaan MA, Antaki JF, Yu Y-C (2000b) Control issues in rotary heart assist devices. In: *Proceedings of the American control conference*, Chicago, 28–30 June, pp 3473–3477
- Boston JR, Antaki JF, Simaan MA (2003) Hierarchical control for hearts assist devices. *IEEE Robot Autom Mag* 10(1):54–64
- Bullister E, Reich S, Sluetz J (2002) Physiologic control algorithms for rotary blood pumps using pressure sensor input. *Artif Organs* 26:931–938
- Faragallah G, Simaan MA (2013) An engineering analysis of the aortic valve dynamics in patients with left ventricular assist devices. *J Healthcare Eng* 4(3):307–327
- Faragallah G, Wang Y, Divo E, Simaan MA (2011) A new current-based control model of the combined cardiovascular and rotary left ventricular assist device. In: *Proceedings of the 2011 American control conference*, San Francisco, 29 June – 1 July, pp 4776–4780
- Faragallah G, Wang Y, Divo E, Simaan MA (2012) A new control system for left ventricular assist devices based on patient-specific physiological demand. *Inverse Prob Sci Eng* 20(5):721–34
- Frazier H, Myers TJ (1999) Left ventricular assist systems as a bridge to myocardial recovery. *Ann Thoracic Surg* 68:734–741
- Giridharan GA, Skliar M, Olsen DB, Pantalos GM (2002) Modeling and control of a brushless DC axial flow ventricular assist device. *Am Soc Artif Int Organ (ASAIO) J* 48:272–289
- Jakovljevic DG, Yacoub MH, Schueler S et al (2017) Left ventricular assist device as a bridge to recovery for patients with advanced heart failure. *J Am Coll Cardiol* 69(15):1924–1933
- Kirklin JK, Pagani FD, Kormos RL, Stevenson LW, Blume ED, Myers SL, Miller MA, Baldwin JT, Young JB, Naftel DC (2017) Eighth annual intermacs report: special focus on framing the impact of adverse events. *J Heart Lung Transplant* 36:1080–1086
- Levin R, Dolgin M, Fox C, Gorlin R (1994) The criteria committee of the New York heart association. In: *Nomenclature and criteria for diagnosis of diseases of the heart and great vessels*, LWW handbooks 9:253–256

- Mansouri M, Salamonsen RF, Lim E, Akmeliawati R, Lovell NH (2015) Preload-based starling-like control for rotary blood pumps: numerical comparison with pulsatility control and constant speed operation. *PLoS One* 10:e0121413
- Marieb E (1994) Human anatomy and physiology. Pearson Education, San Francisco
- Mason D, Hilton A, Salamonsen R (2008) Reliable suction detection for patients with rotary blood pumps. *Am Soc Artif Int Organ (ASAIO) J* 54:359–366
- Moscato F, Arabia M, Colacino FM, Naiyanetr P, Danieli GA, Schima H (2010) Left ventricle afterload impedance control by an axial flow ventricular assist device: a potential tool for ventricular recovery. *Artif Organ* 34:736–744
- Mozzafarian D, Benjamin EJ, Go AS et al (2016) On behalf of the American heart association statistics committee and stroke statistics subcommittee. Heart disease and stroke statistics—2016 update: a report from the American Heart Association. *Circulation* 133:e38–e360
- Ochsner G, Amacher R, Wilhelm MJ et al (2013) A physiological controller for turbodynamic ventricular assist devices based on a measurement of the left ventricular volume. *Artif Organ* 38:527–538
- Olsen D (2000) The history of continuous-flow blood pumps. *Artif Organ* 24(6):401–404
- Petrou A, Lee J, Dual S, Ochsner G, Meboldt M, Daners MS (2018) Standardized comparison of selected physiological controllers for rotary blood pumps: in vitro study. *Artif Organ* 42(3):E29–E42
- Poirier VL (1997) The LVAD: a case study. *The Bridge* 27:14–20
- Prinzing P, Herold ULF, Berkefeld A, Krane M, Lange R, Voss B (2016) Left ventricular assist devices—current state and perspectives. *J Thorac Dis* 8(8):E660–E666
- Shima H, Trubel W, Moritz A, Wieselthaler G, Stohr H, Thomas H, Losert U, Wolner E (1992) Noninvasive monitoring of rotary blood pumps: necessity, possibilities, and limitations. *Artif Organ* 14(2):195–202
- Silverthorn DU (2000) Human physiology: an integrated approach, 2nd edn. Benjamin/Cummings Publishing, New York
- Simaan MA, Ferreira A, Chen S, Antaki JF, Galati D (2009) A dynamical state-space representation and performance analysis of a feedback-controlled rotary left ventricular assist device. *IEEE Trans Control Syst Technol* 1(17):15–28
- Simaan MA, Faragallah G, Wang Y, Divo E (2011) Left ventricular assist devices: engineering design considerations, chap. 2. In: Reyes G (ed) *New aspects of ventricular assist devices*. Intech Publishers, pp 21–42
- Simon MA, Kormos RL, Murali S et al (2005) Myocardial recovery using ventricular assist devices: prevalence, clinical characteristics, and outcomes. *Circulation* 112(9 suppl):I-32–I-36
- Stergiopulos N, Meister J, Westerhof N (1996) Determinants of stroke volume and systolic and diastolic aortic pressure. *Am J Physiol* 270(6):H2050–59
- Suga H, Sagawa K (1974) Instantaneous pressure volume relationships and their ratio in the excised, supported canine left ventricle. *Circ Res* 35(1):117–126
- Vollkron M, Chima H, Huber L, Benkowski R, Morello G, Wieselthaler G (2006) Advanced suction detection for an axial flow pump. *Artif Organ* 30(9):665–670
- Wang Y, Simaan MA (2013) A suction detection system for rotary blood pumps based on Lagrangian support vector machine algorithm. *IEEE J Biomed Health Inform* 17(3):654–663
- Wang Y, Koenig SC, Slaughter MS, Giridharan GA (2015) Rotary blood pump control strategy for preventing left ventricular suction. *Am Soc Artif Int Organ ASAIO J* 61:21–30
- Wu Y, Allaire PE, Tao G, Olsen DB (2007) Modeling, estimation, and control of human circulatory system with a left ventricular assist device. *IEEE Trans Control Syst Technol* 15(17):754–767
- Yancy CW, Jessup M, Bozkurt B et al (2013) ACCF/AHA guideline for the management of heart failure: a report of the American College of Cardiology Foundation/American Heart Association task force on practice guidelines. *J Am Coll Cardiol* 62(16):e147–e239
- Yuhki Y, Hatoh E, Nogawa M, Miura M, Shimazaki Y, Takatani S (1999) Detection of suction and regurgitation of the implantable centrifugal pump based on the motor current waveform analysis and its application to optimization of the pump flow. *Artif Organ* 23:532–537

Control of Linear Systems with Delays

Wim Michiels

Numerical Analysis and Applied Mathematics Section, KU Leuven, Leuven (Heverlee), Belgium

Abstract

The presence of time delays in dynamical systems may induce complex behavior, and this behavior is not always intuitive. Even if a system's equation is scalar, oscillations may occur. Time delays in control loops are usually associated with degradation of performance and robustness, but, at the same time, there are situations where time delays are used as controller parameters. In the chapter, basis properties of time-delay systems are described, and an overview of delay effects in control systems is presented.

Keywords

Delay differential equations · Delays as controller parameters · Functional differential equation

Introduction

Time delays are important components of many systems from engineering, economics, and the life sciences, due to the fact that the transfer of material, energy, and information is mostly not instantaneous. They appear, for instance, as computation and communication lags, they model transport phenomena and heredity, and they arise as feedback delays in control loops. An overview of applications, ranging from traffic flow control and lasers with phase-conjugate feedback, over (bio)chemical reactors and cancer modeling, to control of communication networks and control via networks, is included in Sipahi et al. (2011).

The aim of this contribution is to describe some fundamental properties of linear control systems subjected to time delays and to outline principles behind analysis and synthesis methods. Throughout the text, the results will be illustrated by means of the scalar system

$$\dot{x}(t) = u(t - \tau), \quad (1)$$

which, controlled with instantaneous state feedback, $u(t) = -kx(t)$, leads to the closed-loop system

$$\dot{x}(t) = -kx(t - \tau). \quad (2)$$

Although this didactic example is extremely simple, we shall see that its dynamics are already very rich and shed a light on delay effects in control loops.

In some works, the analysis of (2) is called the *hot shower problem*, as it can be interpreted as a (over)simplified model for a human adjusting the temperature in a shower: $x(t)$ then denotes the difference between the water temperature and the desired temperature as felt by the person taking the shower, the term $-kx(t)$ models the reaction of the person by further opening or closing taps, and the delay is due to the propagation with finite speed of the water in the ducts.

Basis Properties of Time-Delay Systems

Functional Differential Equation

We focus on a model for a time-delay system described by

$$\dot{x}(t) = A_0x(t) + A_1x(t - \tau), \quad x(t) \in \mathbb{R}^n. \quad (3)$$

This is an example of a *functional differential equation* (FDE) of *retarded type*. The term FDE stems from the property that the right-hand side can be interpreted as a functional evaluated at a piece of trajectory. The term retarded expresses that the right-hand side does not explicitly depend on $\dot{x}$.

As a first difference with an ordinary differential equation, the initial condition of (3) at $t = 0$ is a function ϕ from $[-\tau, 0]$ to $\mathbb{R}^n$. For all $\phi \in \mathcal{C}([-\tau, 0], \mathbb{R}^n)$, where $\mathcal{C}([-\tau, 0], \mathbb{R}^n)$ is the space of continuous functions mapping the interval $[-\tau, 0]$ into $\mathbb{R}^n$, a forward solution $x(\phi)$ exists and is uniquely defined.

In Fig. 1, a solution of the scalar system (2) is shown. The discontinuity in the derivative at $t=0$ stems from $A_0\phi(0) + A_1\phi(-\tau) \neq \lim_{\theta \rightarrow 0} \dot{\phi}$. Due to the smoothing property of an integrator, however, at $t = n\tau$, with $n \in \mathbb{N}$, the discontinuity will only be present in the derivative of order $(n + 1)$. This illustrates a second property of functional differential equations of retarded type: solutions become smoother as time evolves. We note that the presence of discontinuities plays an important role in the development of time-integration methods (Bellen 2013). Finally, as a third major difference with ODEs, backward continuation of solutions is not always possible (Michiels and Niculescu 2014).

Reformulation in a First-Order Form

The state of system (3) at time t is the minimal information needed to continue the solution, which, once again, boils down to a function segment $x_t(\phi)$ where $x_t(\phi)(\theta) = x(\phi)(t + \theta)$, $\theta \in [-\tau, 0]$ (in Fig. 1, the function x_t is shown in red for $t = 5$). This suggests that (3) can be reformulated as a standard ordinary differen-

tial equation over the infinite-dimensional space $\mathcal{C}([-\tau, 0], \mathbb{R}^n)$. This equation takes the form

$$\frac{d}{dt}z(t) = \mathcal{A}z(t), \quad z(t) \in \mathcal{C}([-\tau, 0], \mathbb{R}^n) \quad (4)$$

where operator $\mathcal{A}$ is given by

$$\mathcal{D}(\mathcal{A}) = \left\{ \phi \in \mathcal{C}([-\tau_m, 0], \mathbb{R}^n) : \begin{array}{l} \dot{\phi} \in \mathcal{C}([-\tau_m, 0], \mathbb{R}^n) \\ \dot{\phi}(0) = A_0\phi(0) + A_1\phi(-\tau) \end{array} \right\},$$

$$\mathcal{A}\phi = \frac{d\phi}{d\theta}. \quad (5)$$

The relation between solutions of (3) and (4) is given by $z(t)(\theta) = x(t + \theta)$, $\theta \in [-\tau, 0]$. Note that all system information is concentrated in the nonlocal boundary condition describing the domain of $\mathcal{A}$. The representation (4) is closely related to a description by an advection PDE with a nonlocal boundary condition (Krstic 2009).

Asymptotic Growth Rate of Solutions and Stability

The reformulation of (3) into the standard form (4) allows us to define stability notions and to generalize the stability theory for ordinary differential equations in a straightforward way, with the main change that the state space is $\mathcal{C}([-\tau, 0], \mathbb{R}^n)$. For example, the null solution of (3) is exponentially stable if and only if there

exist constants $C > 0$ and $\gamma > 0$ such that

$$\forall \phi \in \mathcal{C}([-\tau_m, 0], \mathbb{R}^n) \quad \|x_t(\phi)\|_s \leq Ce^{-\gamma t} \|\phi\|_s,$$

where $\|\cdot\|_s$ is the supremum norm and $\|\phi\|_s = \sup_{\theta \in [-\tau, 0]} \|\phi(\theta)\|_2$. Furthermore, asymptotic stability and exponential stability are equivalent. A direct generalization of Lyapunov's second method is described by the following theorem.

Theorem 1 *The null solution of linear system (3) is asymptotically stable if there exist a continuous functional $V : \mathcal{C}([-\tau, 0], \mathbb{R}^n) \rightarrow \mathbb{R}$ (a so-called Lyapunov-Krasovskii functional) and continuous nondecreasing functions $u, v, w : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ with*

$$\begin{aligned} u(0) = v(0) = w(0) = 0 \text{ and } u(s) > 0, \\ v(s) > 0, w(s) > 0 \text{ for } s > 0, \end{aligned}$$

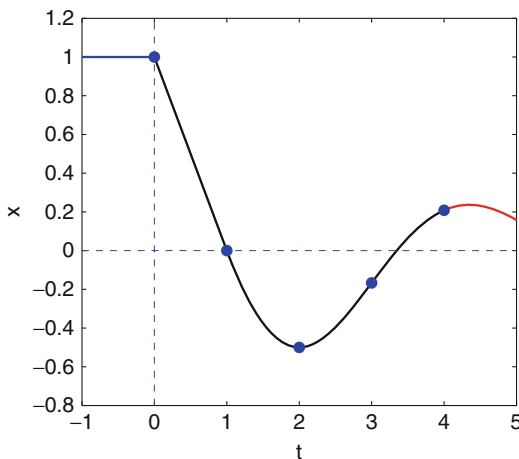
such that for all $\phi \in \mathcal{C}([-\tau, 0], \mathbb{R}^n)$

$$\begin{aligned} u(\|\phi\|_s) \leq V(\phi) \leq v(\|\phi\|_2), \\ \dot{V}(\phi) \leq -w(\|\phi\|_2), \end{aligned}$$

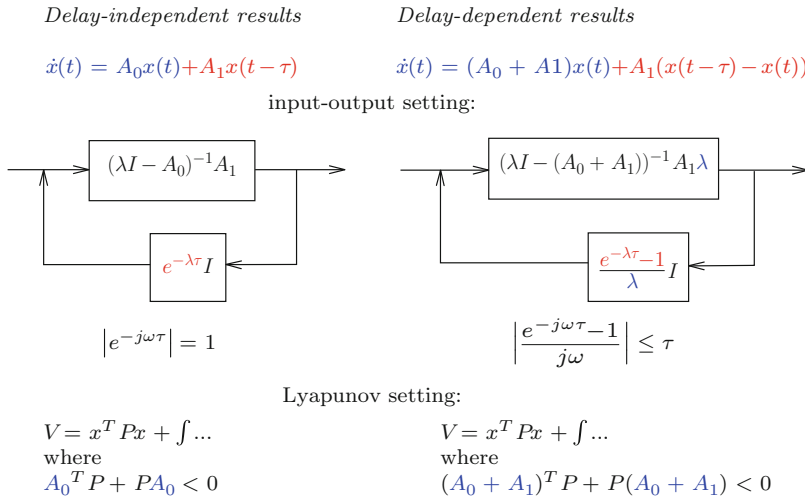
where

$$\dot{V}(\phi) = \limsup_{h \rightarrow 0^+} \frac{1}{h} [V(x_h(\phi)) - V(\phi)].$$

Converse Lyapunov theorems and the construction of the so-called complete-type Lyapunov-Krasovskii functionals are discussed in Kharitonov (2013). Imposing a particular structure on the functional, e.g., a form depending only on a finite number of free parameters, often leads



Control of Linear Systems with Delays, Fig. 1 Solution of (2) for $\tau = 1, k = 1$, and initial condition $\phi \equiv 1$



Control of Linear Systems with Delays, Fig. 2 The classification of stability criteria in delay-independent results and delay-dependent results stems from two differ-

ent perturbation viewpoints. Here, perturbation terms are displayed in *red* color

to easy-to-check stability criteria (for instance, in the form of LMIs), yet as price to pay, the obtained results may be conservative in the sense that the sufficient stability conditions might not be close to necessary conditions. As an alternative to Lyapunov functionals, Lyapunov functions can be used as well, provided that the condition $\dot{V} < 0$ is relaxed (the so-called Lyapunov-Razumikhin approach); see, for example, Gu et al. (2003) and Fridman (2014).

Delay Differential Equations as Perturbation of ODEs

Many results on stability, robust stability, and control of time-delay systems are explicitly or implicitly based on a perturbation point of view, where delay differential equations are seen as perturbations of ordinary differential equations. For instance, in the literature, a classification of stability criteria is often presented in terms of *delay-independent* criteria (conditions holding for all values of the delays) and *delay-dependent* criteria (usually holding for all delays smaller than a bound). This classification has its origin at two different ways of seeing (3) as a perturbation of an ODE, with as nominal system $\dot{x}(t) = A_0 x(t)$ and $\dot{x}(t) = (A_0 + A_1)x(t)$ (system for zero delay), respectively. This observation is

illustrated in Fig. 2 for results based on input-output and Lyapunov-based approaches.

The Spectrum of Linear Time-Delay Systems

Two Eigenvalue Problems

The substitution of an exponential solution in functional differential equation (3) leads us to the *nonlinear eigenvalue problem*

$$(\lambda I - A_0 - A_1 e^{-\lambda\tau})v = 0, \lambda \in \mathbb{C}, v \in \mathbb{C}^n, v \neq 0. \quad (6)$$

The solutions of the equation $\det(\lambda I - A_0 - A_1 e^{-\lambda\tau}) = 0$ are called characteristic roots. Similarly, formulation (4) leads to the equivalent *infinite-dimensional linear eigenvalue problem*

$$(\lambda I - \mathcal{A})u = 0, \lambda \in \mathbb{C}, u \in \mathcal{C}([-\tau, 0], \mathbb{C}^n), u \neq 0. \quad (7)$$

The combination of these two viewpoints lays at the basis of most methods for computing characteristic roots; see Michiels (2012). On the one hand, discretizing (7), i.e., approximating $\mathcal{A}$ with a matrix, and solving the resulting standard eigenvalue problems allow to obtain

global information, for example, estimates of *all* characteristic roots in a given compact set or in a given right half plane. On the other hand, the (finitely many) nonlinear equations (6) allow to make *local corrections* on characteristic root approximations up to the desired accuracy, e.g., using Newton's method or inverse residual iteration. Linear time-delay systems satisfy spectrum-determined growth properties of solutions. For instance, the zero solution of (3) is asymptotically stable if and only if all characteristic roots are in the open left half plane.

In Fig. 3 (left), the rightmost characteristic roots of (2) are depicted for $k\tau = 1$. Note that since the characteristic equation can be written as $\lambda\tau + k\tau e^{-\lambda\tau} = 0$, k and τ can be combined into one parameter. In Fig. 3 (right), we show the real parts of the characteristic roots as a function of $k\tau$. The plots illustrate some important spectral properties of retarded-type FDEs. First, even though there are in general infinitely many characteristic roots, the number of them in *any* right half plane is always finite. Second, the individual characteristic roots, as well as the *spectral abscissa*, i.e., the supremum of the real parts of all characteristic roots, continuously depend on parameters. Related to this, a loss or gain of stability is always associated with characteristic roots crossing the imaginary axis. Figure 3 (right)

also illustrates the transition to a delay-free system as $k\tau \rightarrow 0^+$.

Critical Delays: A Finite-Dimensional Characterization

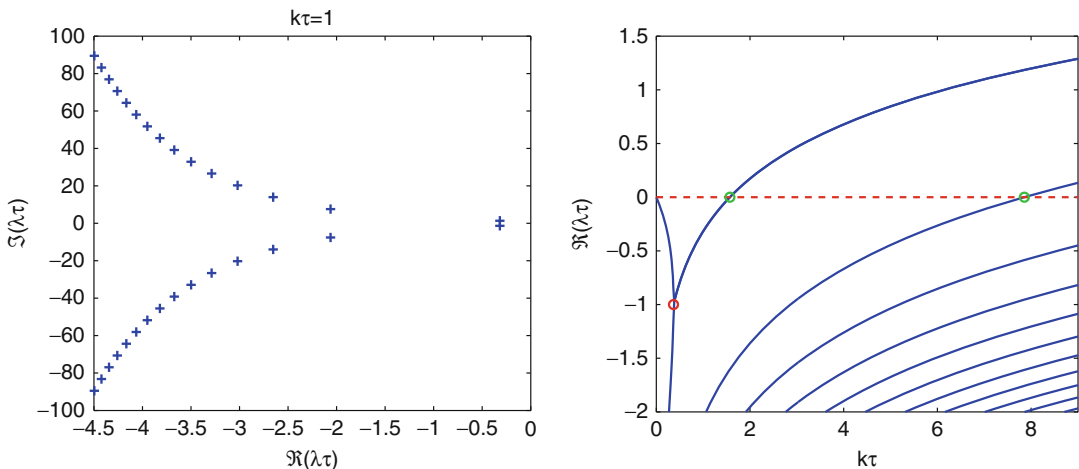
Assume that for a given value of k , we are looking for values of the delay τ_c for which (2) has a characteristic root $j\omega_c$ on the imaginary axis. From $j\omega = -ke^{-j\omega\tau}$, we get

$$\omega_c = k, \quad \tau_c = \frac{\frac{\pi}{2} + l2\pi}{\omega_c}, \quad l = 0, 1, \dots, \quad \Re \left\{ \frac{d\lambda}{d\tau} \Big|_{(\tau_c, j\omega_c)} \right\}^{-1} = \frac{1}{\omega_c^2}. \quad (8)$$

Critical delay values τ_c are indicated with green circles on Fig. 3 (right). The above formulas first illustrate an *invariance property* of imaginary axis roots and their crossing direction with respect to delay shifts of $2\pi/\omega_c$. Second, the number of possible values of ω_c is one and thus *finite*. More generally, substituting $\lambda = j\omega$ in (6) and treating τ as a free parameter lead to a *two-parameter eigenvalue problem*

$$(j\omega I - A_0 - A_1 z)v = 0, \quad (9)$$

with ω on the real axis and $z := \exp(-j\omega\tau)$ on the unit circle. Most methods to solve such a



Control of Linear Systems with Delays, Fig. 3 (Left) Rightmost characteristic roots of (2) for $k\tau = 1$. (Right) Real parts of rightmost characteristic roots as a function of $k\tau$

problem boil down to an elimination of one of the independent variables ω or z . As an example of an elimination technique, we directly get from (9)

$$\begin{aligned} j\omega \in \sigma(A_0 + A_1 z), \quad -j\omega \in \sigma(A_0^* + A_1^* z^{-1}) \\ \Rightarrow \det\left((A_0 + A_1 z) \oplus (A_0^* + A_1^* z^{-1})\right) = 0, \end{aligned}$$

where $\sigma(\cdot)$ denotes the spectrum and $\oplus$ the Kronecker sum. Clearly, the resulting eigenvalue problem in z is finite dimensional.

Control of Linear Time-Delay System

Limitations Induced by Delays

It is well-known that delays in control loop may lead to a significant degradation of performance and robustness and even to instability (Fridman 2014; Niculescu 2001; Richard 2003). Let us return to example (2). As illustrated with Fig. 3 and expressions (8), the system loses stability if τ reaches the value $\pi/2k$, while stability cannot be recovered for larger delays. The maximum achievable exponential decay rate of the solutions, which corresponds to the minimum of the spectral abscissa, is given by $-1/\tau$; hence, large delays can only be tolerated at the price of a degradation of the rate of convergence. It should be noted that the limitations induced by delays are even more stringent if the uncontrolled systems are exponentially unstable, which is not the case for (2).

The analysis in the previous sections gives a hint why control is difficult in the presence of delays: the system is inherently infinite dimensional. As a consequence, most control design problems which involve determining a finite number of parameters can be interpreted as reduced-order control design problems or as control design problems for under-actuated systems, which both are known to be hard problems.

Fixed-Order Control

Most standard control design techniques lead to controllers whose dimension is larger or equal to the dimension of the system. For infinite-dimensional time-delay system, such

controllers might have a disadvantage of being complicated and hard to implement. To see this, for a system with delay in the state, the generalization of static state feedback, $u(t) = k(x)$, is given by $u(t) = \int_{-\tau}^0 x(t + \theta) d\mu(\theta)$, where μ is a function of bounded variation. However, in the context of large-scale systems, it is known that reduced-order controllers often perform relatively well compared to full-order controllers, while they are much easier to implement.

Recently, methods for the design of controllers with a prescribed order (dimension) or structure have been proposed (Michiels 2012). These methods rely on a direct optimization of appropriately defined cost functions (spectral abscissa, $\mathcal{H}_2/\mathcal{H}_\infty$ criteria). While $\mathcal{H}_2$ criteria can be addressed within a derivative-based optimization framework, $\mathcal{H}_\infty$ criteria and the spectral abscissa require targeted methods for *non-smooth optimization problems*. To illustrate the need for such methods, consider again Fig. 3 (right): minimizing the spectral abscissa for a given value of τ as a function of the controller gain k leads to an optimum where the objective function is not differentiable, even not locally Lipschitz, as shown by the red circle. In case of multiple controller parameters, the path of steepest descent in the parameter space typically has phases along a manifold characterized by the non-differentiability of the objective function.

Using Delays as Controller Parameters

In contrast to the detrimental effects of delays, there are situations where delays have a beneficial effect and are even used as controller parameters; see Sipahi et al. (2011). For instance, delayed feedback can be used to stabilize oscillatory systems where the delay serves to adjust the phase in the control loop. Delayed terms in control laws can also be used to approximate derivatives in the control action. Control laws which depend on the difference $x(t) - x(t - \tau)$, the so-called Pyragas-type feedback, have the property that the position of equilibria and the shape of periodic orbits with period τ are not affected, in contrary to their stability properties. Last but not least, delays can be used in control schemes to generate predictions or to stabilize predictors, which allow

to compensate delays and improve performance (Krstic 2009). Let us illustrate the main idea once more with system (1), considered as model for the shower problem. Its special structure, in the sense that the delay is only present in the input, can be fully exploited.

The person who is taking a shower is – possibly after some bad experiences – aware about the delay and will take into account his/her prediction of the system's reaction when adjusting the cold and hot water supply. Let us formalize this. The uncontrolled system can be rewritten as $\dot{x}(t) = v(t)$, where $v(t) = u(t - \tau)$. We know u up to the current time t ; thus, we know v up to time $t + \tau$, and if $x(t)$ is also known, we can predict the value of x at time $t + \tau$,

$$\begin{aligned} x_p(t + \tau) &= x(t) + \int_t^{t+\tau} v(s) ds \\ &= x(t) + \int_{t-\tau}^t u(s) ds, \end{aligned}$$

and use the predicted state for feedback. With the control law $u(t) = -kx_p(t + \tau)$, there is only one closed-loop characteristic root at $\lambda = -k$, i.e., as long as the model used in the predictor is exact, the delay in the loop is compensated by the prediction. For further reading on prediction-based controllers, see, e.g., Krstic (2009) and the references therein.

Summary and Future Directions

Time-delay systems, which appear in a large number of applications, are a class of infinite-dimensional systems, resulting in rich dynamics and challenges from a control point of view. The different representations and interpretations and, in particular, the combination of viewpoints lead to a wide variety of analysis and synthesis tools.

It is important to mention that many discussed properties do not (necessarily) carry over to functional differential equations of neutral type. In general solutions of neutral type systems do not become smoother as time evolves. Linear time-invariant systems may have infinitely many characteristic roots in a right half plane, while

asymptotic stability does not always imply exponential stability. In the multiple delays case asymptotic stability might even be sensitive with respect to infinitesimal delay perturbations, as discussed in Michiels and Niculescu (2014).

Active research topics include improved structure exploiting methods for robust control, control of complex and networked systems, model reduction approaches for delay systems, as well as applications in traffic, machining and vibration control.

Cross-References

- [Control of Nonlinear Systems with Delays](#)
- [H-Infinity Control](#)
- [H₂ Optimal Control](#)
- [Optimization-Based Control Design Techniques and Tools](#)

Acknowledgments The author received funding from the project C14/17/072 of the KU Leuven Research Council, the project G0A5317N of the Research Foundation-Flanders (FWO - Vlaanderen), and from the project UCoCoS, funded by the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No 675080.

Bibliography

- Bellen E, Zennaro M (2013) Numerical methods for delay differential equations. Oxford University Press, Oxford
- Fridman E (2014) Introduction to time-delay systems. Analysis and control. Birkhäuser, Basel
- Gu K, Kharitonov VL, Chen J (2003) Stability of time-delay systems. Birkhäuser, Basel
- Kharitonov VL (2013) Time-delay systems. Lyapunov functionals and matrices. Birkhäuser, Basel
- Krstic M (2009) Delay compensation for nonlinear, adaptive, and PDE systems. Birkhäuser, Basel
- Michiels W (2012) Design of fixed-order stabilizing and $\mathcal{H}_2 - \mathcal{H}_\infty$ optimal controllers: an eigenvalue optimization approach. In: Time-delay systems: methods, applications and new trends. Lecture notes in control and information sciences, vol 423. Springer, Berlin/Heidelberg, pp 201–216
- Michiels W, Niculescu S-I (2014) Stability, control, and computation for time-delay systems. An eigenvalue based approach, 2nd edn. SIAM, Philadelphia
- Niculescu S-I (2001) Delay effects on stability: a robust control approach. Lecture notes in control and information sciences, vol 269. Springer, Berlin/New York

Richard J-P (2003) Time-delay systems: an overview of recent advances and open problems. *Automatica* 39(10):1667–1694

Sipahi R, Niculescu S, Abdallah C, Michiels W, Gu K (2011) Stability and stabilization of systems with time-delay. *IEEE Control Syst Mag* 31(1):38–65

Control of Machining Processes

Kaan Erkorkmaz

Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON, Canada

Abstract

Control of machining processes encompasses a broad range of technologies and innovations, ranging from optimized motion planning and servo drive loop design to on-the-fly regulation of cutting forces and power consumption to applying control strategies for damping out chatter vibrations caused by the interaction of the chip generation mechanism with the machine tool structural dynamics. This article provides a brief introduction to some of the concepts and technologies associated with machining process control.

Keywords

Adaptive control · Chatter vibrations · Feed drive control · Machining · Trajectory planning

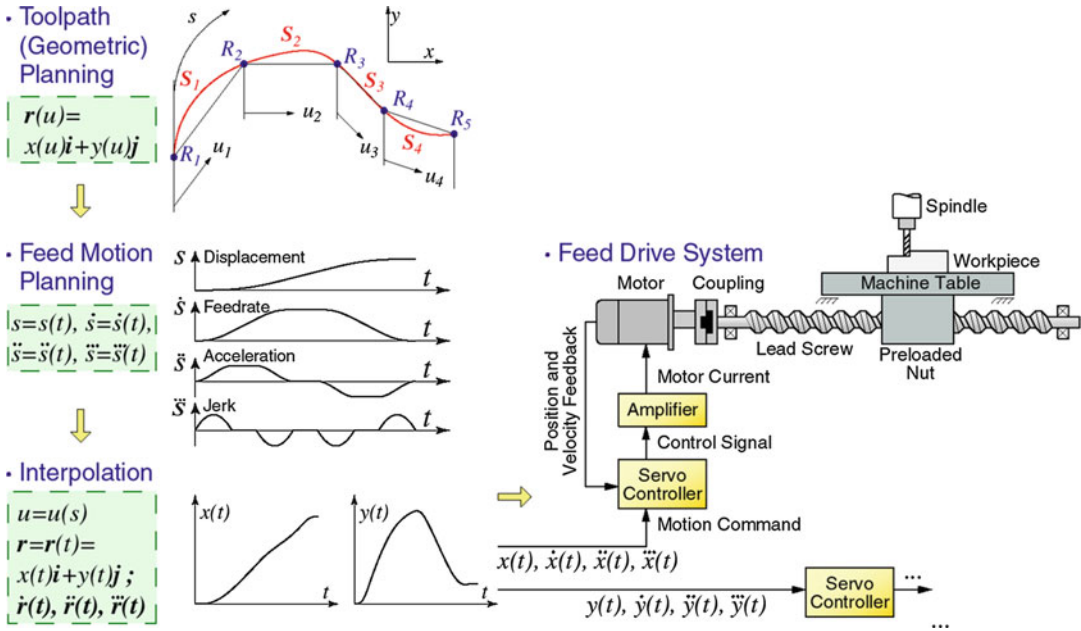
Introduction

Machining is used extensively in the manufacturing industry as a shaping process, where high product accuracy, quality, and strength are required. From automotive and aerospace components, to dies and molds, to biomedical implants, and even mobile device chassis,

many manufactured products rely on the use of machining.

Machining is carried out on machine tools, which are multi-axis mechatronic systems designed to provide the relative motion between the tool and workpiece, in order to facilitate the desired cutting operation. Figure 1 illustrates a single axis of a ball screw-driven machine tool, performing a milling operation. Here, the cutting process is influenced by the motion of the servo drive. The faster the part is fed in towards the rotating cutter, the larger the cutting forces become, following a typically proportional relationship that holds for a large class of milling operations (Altintas 2012). The generated cutting forces, in turn, are absorbed by the machine tool and feed drive structure. They cause mechanical deformation and may also excite the vibration modes, if their harmonic content is near the structural natural frequencies. This may, depending on the cutting speed and tool and workpiece engagement conditions, lead to forced vibrations or chatter (Altintas 2012).

The disturbance effect of cutting forces is also felt by the servo control loop, consisting of mechanical, electrical, and digital components. This disturbance may result in the degradation of tool positioning accuracy, thereby leading to part errors. Another input that influences the quality achieved in a machining operation is the commanded trajectory. Discontinuous or poorly designed motion commands, with acceleration discontinuity, lead typically to jerky motion, vibrations, and poor surface finish. Beyond motion controller design and trajectory planning, emerging trends in machining process control include regulating, by feedback, various outcomes of the machining process, such as peak resultant cutting force, spindle power consumption, and amplitude of vibrations caused by the machining process. In addition to using actuators and instrumentation already available on a machine tool, such as feed and spindle drives and current sensors, additional devices, such as dynamometers, accelerometers, as well as inertial or piezoelectric actuators, may need to be used in order to achieve the required level of feedback and control injection capability.



Control of Machining Processes, Fig. 1 Single axis of a ball screw-driven machine tool performing milling

Servo Drive Control

Stringent requirements for part quality, typically specified in microns, coupled with disturbance force inputs coming from the machining process, which can be in the order of tens to thousands of Newtons, require that the disturbance rejection of feed drives, which act as dynamic (i.e., frequency dependent) “stiffness” elements, be kept as strong as possible. In traditional machine design, this is achieved by optimizing the mechanical structure for maximum rigidity. Afterwards, the motion control loop is tuned to yield the highest possible bandwidth (i.e., responsive frequency range), without interfering with the vibratory modes of the machine tool in a way that can cause instability. The P-PI position velocity cascade control structure, shown in Fig. 2, is the most widely used technique in machine tool drives. Its tuning guidelines have been well established in the literature (Ellis 2004). To augment the command following accuracy, velocity and acceleration feedforward, and friction compensation terms are added. Increasing the closed-loop bandwidth yields better disturbance rejection and more accurate

tracking of the commanded trajectory (Pritschow 1996), which is especially important in high-speed machining applications where elevated cutting speeds necessitate faster feed motion.

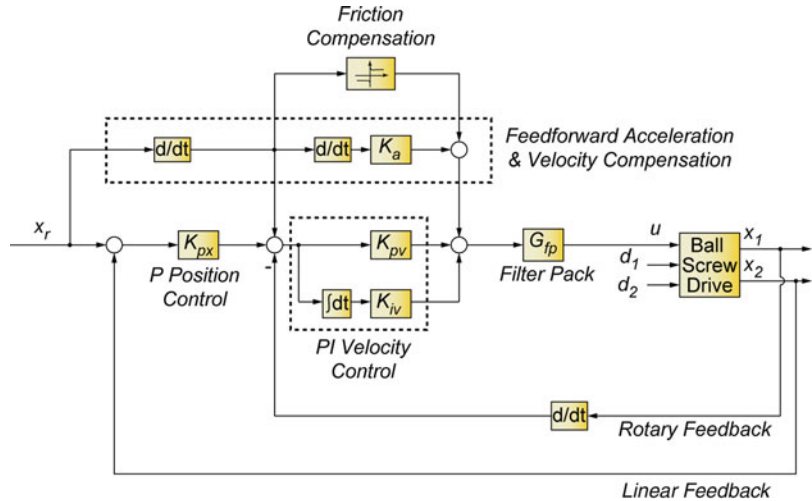
It can be seen in Fig. 2 that increased axis tracking errors (e_x and e_y) may result in increased contour error (ε). A practical solution to mitigate this problem, in machine tool engineering, is to also match the dynamics of different motion axes, so that the tracking errors always assume an instantaneous proportion that brings the actual tool position as close as possible to the desired toolpath (Koren 1983). Sometimes, the control action can be designed to directly reduce the contour error as well, which leads to the structure known as “cross-coupling control” (Koren 1980).

Trajectory Planning

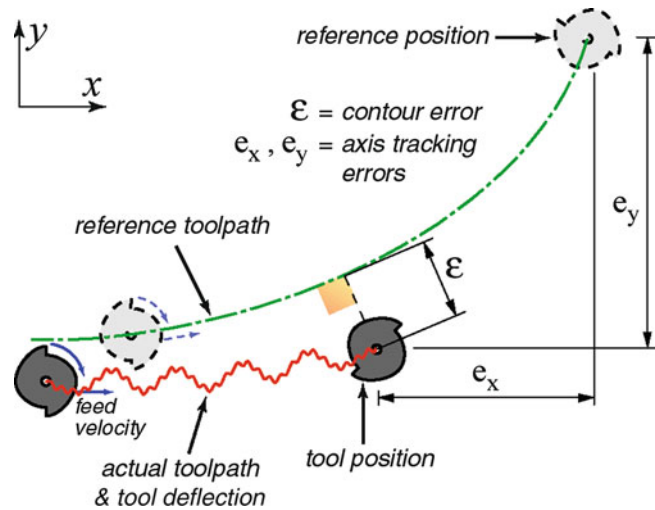
Smooth trajectory planning with at least acceleration level continuity is required in machine tool control, in order to avoid inducing unwanted vibration or excessive tracking error during the machining process. For this purpose, computer numerical control (CNC) systems are equipped

Control of Machining

Processes, Fig. 2 P-Pi position velocity cascade control used in machine tool drives

**Control of Machining**

Processes, Fig. 3 Formation of contour error (ϵ), as a result of servo errors (e_x and e_y) in the individual axes

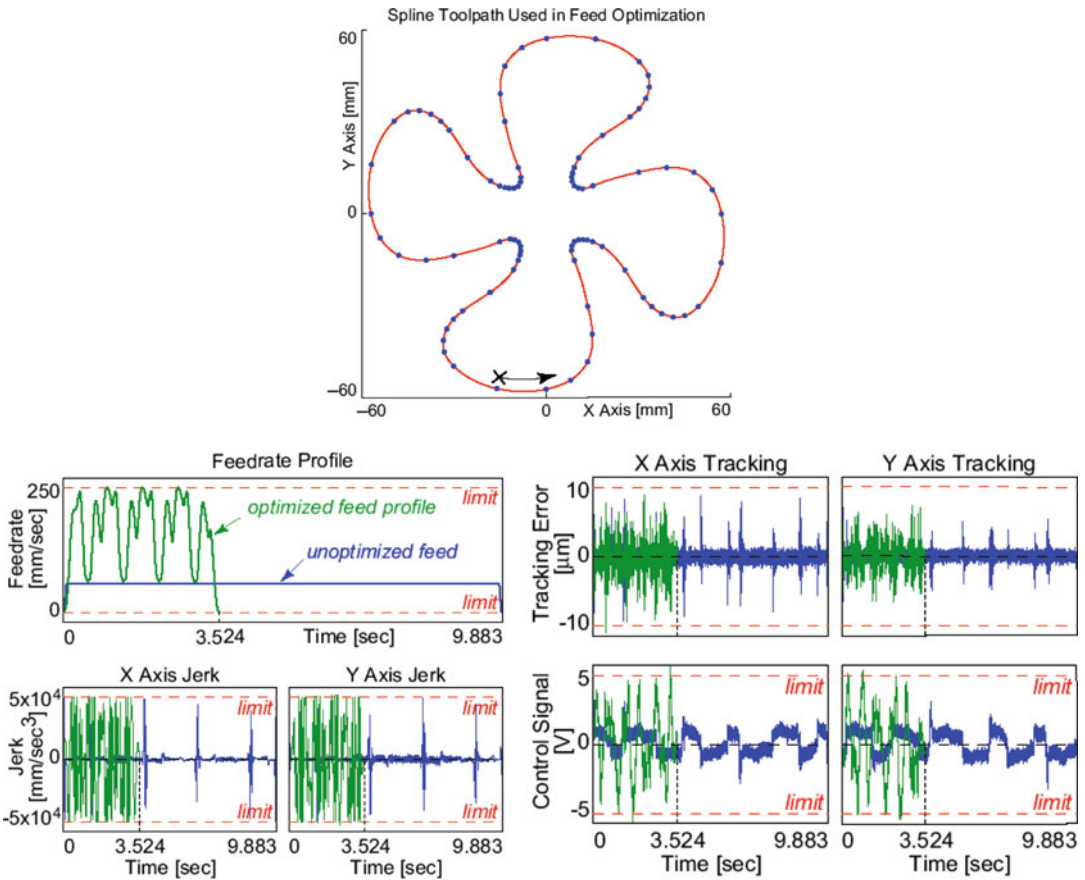


with various spline toolpath interpolation functions, such as B-splines, and NURBS. The feedrate (i.e., progression speed along the toolpath) is planned in the “look-ahead” function of the CNC so that the total machining cycle time is reduced as much as possible. This has to be done without violating the position-dependent feedrate limits already programmed into the numerical control (NC) code, which are specified by considering various constraints coming from the machining process.

In feedrate optimization, axis level trajectories have to stay within the velocity and torque limits of the drives, in order to avoid damaging

the machine tool or causing actuator saturation. Moreover, as an indirect way of containing tracking errors, the practice of limiting axis level jerk (i.e., rate of change of acceleration) is applied (Gordon and Erkorkmaz 2013). This results in reduced machining cycle time, while avoiding excessive vibration or positioning error due to “jerky” motion.

An example of trajectory planning using quintic (5th degree) polynomials for toolpath parameterization is shown in Fig. 4. Here, comparison is provided between unoptimized and optimized feedrate profiles subject to the same axis velocity, torque (i.e., control signal), and



Control of Machining Processes, Fig. 4 Example of quintic spline trajectory planning without and with feedrate optimization

jerk limits. As can be seen, significant machining time reduction can be achieved through trajectory optimization, while retaining the dynamic tool position accuracy. While Fig. 4 shows the result of an elaborate nonlinear optimization approach (Altintas and Erkorkmaz 2003), practical look-ahead algorithms have also been proposed which lead to more conservative cycle times but are much better suited for real-time implementation inside a CNC (Weck et al. 1999).

Adaptive Control of Machining

There are established mathematical methods for predicting cutting forces, torque, power, and even surface finish for a variety of machining operations like turning, boring,

drilling, and milling (Altintas 2012). However, when machining complex components, such as gas turbine impellers, or dies and molds, the tool and workpiece engagement and workpiece geometry undergo continuous change. Hence, it may be difficult to apply such prediction models efficiently, unless they are fully integrated inside a computer-aided process planning environment, as reported for 3-axis machining by Altintas and Merdol (2007).

An alternative approach, which allows the machining process to take place within safe and efficient operating bounds, is to use feedback from the machine tool during the cutting process. This measurement can be of the cutting forces using a dynamometer or the spindle power consumption. This measurement is then used inside a feedback control loop

to override the commanded feedrate value, which has direct impact on the cutting forces and power consumption. This scheme can be used to ensure that the cutting forces do not exceed a certain limit for process safety or to increase the feed when the machining capacity is underutilized, thus boosting productivity. Since the geometry and tool engagement are generally continuously varying, the coefficients of a model that relates the cutting force (or power) to the feed command are also time-varying. Furthermore, in CNC controllers, depending on the trajectory generation architecture, the execution latency of a feed override command may not always be deterministic. Due to these sources of variability, rather than using classical fixed gain feedback, machining control research has evolved around adaptive control techniques (Masory and Koren 1980; Spence and Altintas 1991), where changes in the cutting process dynamics are continuously tracked and the control law, which computes the proceeding feedrate override, is updated accordingly. This approach has produced significant cycle time reduction in 5-axis machining of gas turbine impellers, as reported in Budak and Kops (2000) and shown in Fig. 5.

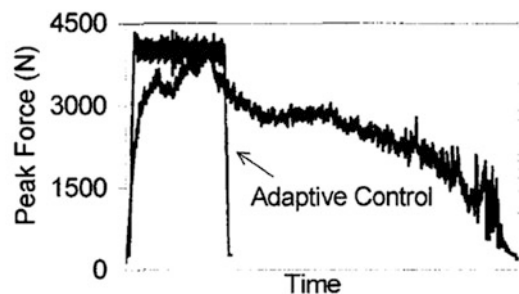
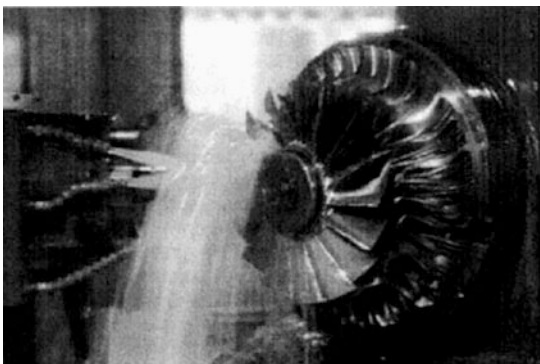
Control of Chatter Vibrations

Chatter vibrations are caused by the interaction of the chip generation mechanism with the

structural dynamics of the machine, tool, and workpiece assembly (see Fig. 6). The relative vibration between the tool and workpiece generates a wavy surface finish. In the consecutive tool pass, a new wave pattern, caused by the current instantaneous vibration, is generated on top of the earlier one. If the formed chip, which has an undulated geometry, displays a steady average thickness, then the resulting cutting forces and vibrations also remain bounded. This leads to a stable steady-state cutting regime, known as “forced vibration.” On the other hand, if the chip thickness keeps increasing at every tool pass, resulting in increased cutting forces and vibrations, then chatter vibration is encountered. Chatter can be extremely detrimental to the machined part quality, tool life, and the machine tool.

Chatter has been reported in literature to be caused by two main phenomena: self-excitation through regeneration and mode coupling. For further information on chatter theory, the reader is referred to Altintas (2012) as an excellent starting point.

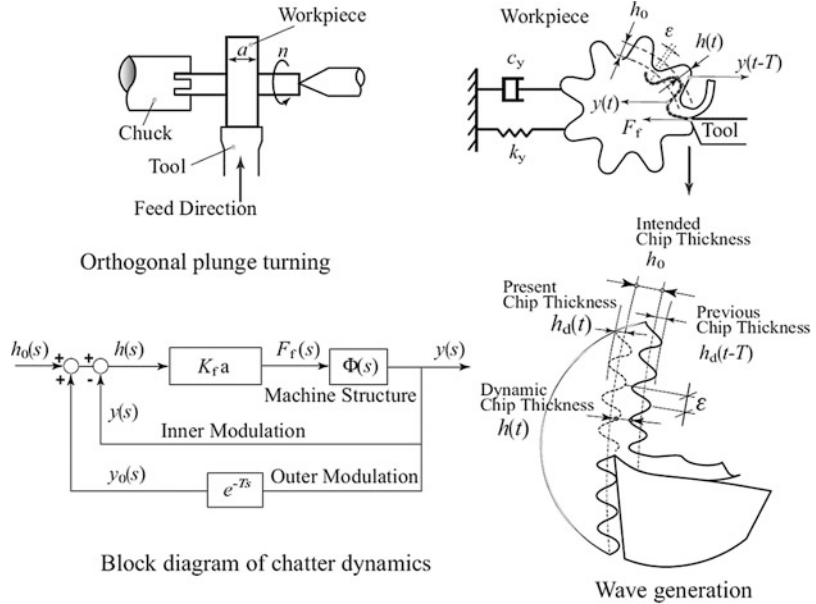
Various mitigation measures have been investigated and proposed in order to avoid and control chatter. One widespread approach is to select chatter-free cutting conditions through detailed modal testing and stability analyses. Recently, to achieve higher material removal rates, the application of active damping has started to receive interest. This has been realized through specially designed tools and actuators



Control of Machining Processes, Fig. 5 Example of 5-axis impeller machining with adaptive force control (Source: Budak and Kops (2000), courtesy of Elsevier)

Control of Machining Processes, Fig. 6

Schematic of the chatter vibration mechanism for one degree of freedom (From: Altintas (2012), courtesy of Cambridge University Press)



(Pratt and Nayfeh 2001; Munoa et al. 2013) and demonstrated productivity improvement in boring and milling operations. As another method for chatter suppression, modulation of the cutting (i.e., spindle) speed has been successfully applied as a means of interrupting the regeneration mechanism (Soliman and Ismail 1997; Zatarain et al. 2008).

levels of rigor, so that these new technologies can be utilized reliably and at their full potential.

Cross-References

- [Adaptive Control: Overview](#)
- [PID Control](#)
- [Robot Motion Control](#)

Summary and Future Directions

This article has presented an overview of various concepts and emerging technologies in the area of machining process control. The new generation of machine tools, designed to meet the ever-growing productivity and efficiency demands, will likely utilize advanced forms of these ideas and technologies in an integrated manner. As more computational power and better sensors become available at lower cost, one can expect to see new features, such as more elaborate trajectory planning algorithms, active vibration damping techniques, and real-time process and machine simulation and control capability, beginning to appear in CNC units. No doubt that the dynamic analysis and controller design for such complicated systems will require higher

Bibliography

- Altintas Y (2012) Manufacturing automation: metal cutting mechanics, machine tool vibrations, and CNC design, 2nd edn. Cambridge University Press, Cambridge
- Altintas Y, Erkorkmaz K (2003) Feedrate optimization for spline interpolation in high speed machine tools. Ann CIRP 52(1):297–302
- Altintas Y, Merdol DS (2007) Virtual High performance milling. Ann CIRP 55(1):81–84
- Budak E, Kops L (2000) Improving productivity and part quality in milling of titanium based impellers by chatter suppression and force control. Ann CIRP 49(1): 31–36
- Ellis GH (2004) Control system design guide, 3rd edn. Elsevier Academic, New York
- Gordon DJ, Erkorkmaz K (2013) Accurate control of ball screw drives using pole-placement vibration damping and a novel trajectory prefilter. Precis Eng 37(2):308–322

- Koren Y (1980) Cross-coupled biaxial computer control for manufacturing systems. *ASME J Dyn Syst Meas Control* 102:265–272
- Koren Y (1983) *Computer control of manufacturing systems*. McGraw-Hill, New York
- Masory O, Koren Y (1980) Adaptive control system for turning. *Ann CIRP* 29(1):281–284
- Munoa J, Mancisidor I, Loix N, Uriarte LG, Barcena R, Zatarain M (2013) Chatter suppression in ram type travelling column milling machines using a biaxial inertial actuator. *Ann CIRP* 62(1):407–410
- Pratt JR, Nayfeh AH (2001) Chatter control and stability analysis of a cantilever boring bar under regenerative cutting conditions. *Philos Trans R Soc* 359:759–792
- Pritschow G (1996) On the influence of the velocity gain factor on the path deviation. *Ann CIRP* 45/1:367–371
- Soliman E, Ismail F (1997) Chatter suppression by adaptive speed modulation. *Int J Mach Tools Manuf* 37(3):355–369
- Spence A, Altintas Y (1991) CAD assisted adaptive control for milling. *ASME J Dyn Syst Meas Control* 113(3):444–450
- Weck M, Meylahn A, Hardebusch C (1999) Innovative algorithms for spline-based CNC controller. *Ann Ger Acad Soc Prod Eng* VI(1):83–86
- Zatarain M, Bediaga I, Munoa J, Lizarralde R (2008) Stability of milling processes with continuous spindle speed variation: analysis in the frequency and time domains, and experimental correlation. *Ann CIRP* 57(1):379–384

Control of Networks of Underwater Vehicles

Naomi Ehrich Leonard
Department of Mechanical and Aerospace
Engineering, Princeton University, Princeton,
NJ, USA

Abstract

Control of networks of underwater vehicles is critical to underwater exploration, mapping, search, and surveillance in the multiscale, spatiotemporal dynamics of oceans, lakes, and rivers. Control methodologies have been derived for tasks including feature

tracking and adaptive sampling and have been successfully demonstrated in the field despite the severe challenges of underwater operations.

Keywords

Adaptive sampling · Feature tracking · Gliders · Mobile sensor arrays · Underwater exploration

Introduction

The development of theory and methodology for control of networks of underwater vehicles is motivated by a multitude of underwater applications and by the unique challenges associated with operating in the oceans, lakes, and rivers. Tasks include underwater exploration, mapping, search, and surveillance, associated with problems that include pollution monitoring, human safety, resource seeking, ocean science, and marine archeology. Vehicle networks collect data on underwater physics, biology, chemistry, and geology for improving the understanding and predictive modeling of natural dynamics and human-influenced changes in marine environments. Because the underwater environment is opaque, inhospitable, uncertain, and dynamic, control is critical to the performance of vehicle networks.

Underwater vehicles typically carry sensors to measure external environmental signals and fields, and thus a vehicle network can be regarded as a mobile sensor array. The underlying principle of control of networks of underwater vehicles leverages their mobility and uses an interacting dynamic among the vehicles to yield a high-performing collective behavior. If the vehicles can communicate their state or measure the relative state of others, then they can cooperate and coordinate their motion.

One of the major drivers of control of underwater mobile sensor networks is the multiscale, spatiotemporal dynamics of the environmental fields and signals. In Curtin et al. (1993), the concept of the autonomous oceanographic sampling network (AOSN),

featuring a network of underwater vehicles, was introduced for dynamic measurement of the ocean environment and resolution of spatial and temporal gradients in the sampled fields. For example, to understand the coupled biological and physical dynamics of the ocean, data are required both on the small-scale dynamics of phytoplankton, which are major actors in the marine ecosystem and the global climate, and on the large-scale dynamics of the flow field, temperature, and salinity.

Accordingly, control laws are needed to coordinate the motion of networks of underwater vehicles to match the many relevant spatial and temporal scales. And for a network of underwater vehicles to perform complex missions reliably and efficiently, the control must address the many uncertainties and real-world constraints including the influence of currents on the motion of the vehicles and the limitations on underwater communication.

Vehicles

Control of networks of underwater vehicles is made possible with the availability of small (e.g., 1.5–2 m long), relatively inexpensive autonomous underwater vehicles (AUVs). Propelled AUVs such as the REMUS provide maneuverability and speed. These kinds of AUVs respond quickly and agilely to the needs of the network, and because of their speed, they can often power through strong ocean flows. However, propelled AUVs are limited by their batteries; for extended missions, they need docking stations or other means to recharge their batteries.

Buoyancy-driven autonomous underwater gliders, including the Slocum, the Spray, and the Seaglider, are a class of endurance AUVs designed explicitly for collecting data over large three-dimensional volumes continuously over periods of weeks or even months (Rudnick et al. 2004). They move slowly and steadily, and, as a result, they are particularly well suited to network missions of long duration.

Gliders propel themselves by alternately increasing and decreasing their buoyancy using either a hydraulic or a mechanical buoyancy engine. Lift generated by flow over fixed wings converts the vertical ascent/descent induced by the change in buoyancy into forward motion, resulting in a sawtooth-like trajectory in the vertical plane. Gliders can actively redistribute internal mass to control attitude, for example, they pitch by sliding their battery pack forward and aft. For heading control, they shift mass to roll, bank, and turn or deflect a rudder. Some gliders are designed for deep water, e.g., to 1,500 m, while others for shallower water, e.g., to 200 m.

Gliders are typically operated at their maximum speed and thus they move at approximately constant speed relative to the flow. Because this is relatively slow, on the order of 0.3–0.5 m/s in the horizontal direction and 0.2 m/s in the vertical, ocean currents can sometimes reach or even exceed the speed of the gliders. Unlike a propelled AUV, which typically has sufficient thrust to maintain course despite currents, a glider trying to move in the direction of a strong current will make no forward progress. This makes coordinated control of gliders challenging; for instance, two sensors that should stay sufficiently far apart may be pushed toward each other leading to less than ideal sampling conditions.

Communication and Sensing

Underwater communication is one of the biggest challenges to the control of networks of underwater vehicles and one that distinguishes it from control of vehicles on land or in the air. Radio-frequency communication is not typically available underwater, and acoustic data telemetry has limitations including sensitivity to ambient noise, unpredictable propagation, limited bandwidth, and latency.

When acoustic communication is too limiting, vehicles can surface periodically and communicate via satellite. This method may be bandwidth limited and will require time and energy. However, in the case of profiling propelled AUVs

or underwater gliders, they already move in the vertical plane in a sawtooth pattern and thus regularly come closer to the surface. When on the surface, vehicles can also get a GPS fix whereas there is no access to GPS underwater. The GPS fix is used for correcting onboard dead reckoning of the vehicle's absolute position and for updating onboard estimation of the underwater currents, both helpful for control.

Vehicles are typically equipped with conductivity-temperature-density (CTD) sensors to measure temperature, salinity, and density. From this pressure can be computed and thus depth and vertical speed. Attitude sensors provide measurements of pitch, roll, and heading. Position and velocity in the plane is estimated using dead reckoning. Many sensors for measuring the environment have been developed for use on underwater vehicles; these include chlorophyll fluorometers to estimate phytoplankton abundance, acoustic Doppler profilers (ADPs) to measure variations in water velocity, and sensors to measure pH, dissolved oxygen, and carbon dioxide.

Control

Described here are a selection of control methodologies designed to serve a variety of underwater applications and to address many of the challenges described above for both propelled AUVs and underwater gliders. Some of these methodologies have been successfully field tested in the ocean.

Formations for Tracking Gradients, Boundaries, and Level Sets in Sampled Fields

While a small underwater vehicle can take only single-point measurements of a field, a network of N vehicles employing cooperative control laws can move as a formation and estimate or track a gradient in the field. This can be done in a straightforward way in 2D with three vehicles and can be extended to 3D with additional vehicles. Consider $N = 3$ vehicles moving together in an equilateral triangular formation and sampling a

2D field $T : \mathbb{R}^2 \rightarrow \mathbb{R}$. The formation serves as a sensor array and the triangle side length defines the resolution of the array.

Let the position of the i th vehicle be $\mathbf{x}_i \in \mathbb{R}^2$. Consider double integrator dynamics $\ddot{\mathbf{x}}_i = \mathbf{u}_i$, where $\mathbf{u}_i \in \mathbb{R}^2$ is the control force on the i th vehicle. Suppose that each vehicle can measure the relative position of each of its neighbors, $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. Decentralized control that derives from an artificial potential is a popular method for each of the three vehicles to stay in the triangular formation of prescribed resolution d_0 . Consider the nonlinear interaction potential $V_I : \mathbb{R}^2 \rightarrow \mathbb{R}$ defined as

$$V_I(\mathbf{x}_{ij}) = k_s \left(\ln \|\mathbf{x}_{ij}\| + \frac{d_0}{\|\mathbf{x}_{ij}\|^2} \right)$$

where $k_s > 0$ is a scalar gain. The control law for the i th vehicle derives as the gradient of this potential with respect to $\mathbf{x}_i$ as follows:

$$\ddot{\mathbf{x}}_i = \mathbf{u}_i = - \sum_{j=1, j \neq i}^N \nabla V_I(\mathbf{x}_{ij}) - k_d \dot{\mathbf{x}}_i$$

where a damping term is added with scalar gain $k_d > 0$. Stability of the triangle of resolution d_0 is proved with the Lyapunov function

$$V = \frac{1}{2} \sum_{i=1}^N \|\dot{\mathbf{x}}_i\|^2 + \sum_{i=1}^{N-1} \sum_{j=i+1}^N V_I(\mathbf{x}_{ij}).$$

Now let each vehicle use the sequence of single-point measurements it takes along its path to compute the projection of the spatial gradient onto its normalized velocity, $\mathbf{e}_{\dot{\mathbf{x}}} = \dot{\mathbf{x}}_i / \|\dot{\mathbf{x}}_i\|$, i.e., $\nabla T_P(\mathbf{x}, \dot{\mathbf{x}}_i) = (\nabla T(\mathbf{x}) \cdot \mathbf{e}_{\dot{\mathbf{x}}}) \mathbf{e}_{\dot{\mathbf{x}}}$. Following Bachmayer and Leonard (2002), let

$$\ddot{\mathbf{x}}_i = \mathbf{u}_i = \kappa \nabla T_P(\mathbf{x}, \dot{\mathbf{x}}_i) - \sum_{j=1, j \neq i}^N \nabla V_I(\mathbf{x}_{ij}) - k_d \dot{\mathbf{x}}_i,$$

where κ is a scalar gain. For $\kappa > 0$, each vehicle will accelerate along its path when it measures an increasing T and decelerates for a decreasing T . Each vehicle will also turn to keep up with the

others so that the formation will climb the spatial gradient of T to find a local maximum.

Alternative control strategies have been developed that add versatility in feature tracking. The virtual body and artificial potential (VBAP) multivehicle control methodology (Ögren et al. 2004) was demonstrated with a network of Slocum autonomous underwater gliders in the AOSN II field experiment in Monterey Bay, California, in August 2003 (Fiorelli et al. 2006). VBAP is well suited to the operational scenario described above in which vehicles surface asynchronously to establish communication with a base.

VBAP is a control methodology for coordinating the translation, rotation, and dilation of a group of vehicles. A virtual body is defined by a set of reference points that move according to dynamics that are computed centrally and made available to the vehicles in the group. Artificial potentials are used to couple the dynamics of vehicles and a virtual body so that control laws can be derived that stabilize desired formations of vehicles and a virtual body. When sampled measurements of a scalar field can be communicated, the local gradients can be estimated. Gradient climbing algorithms prescribe virtual body direction, so that, for example, the vehicle network can be directed to head for the coldest water or the highest concentration of phytoplankton. Further, the formation can be dilated so that the resolution can be adapted to minimize error in estimates. Control of the speed of the virtual body ensures stability and convergence of the vehicle formation.

These ideas have been extended further to design provable control laws for cooperative level set tracking, whereby small vehicle groups cooperate to generate contour plots of noisy, unknown fields, adjusting their formation shape to provide optimal filtering of their noisy measurements (Zhang and Leonard 2010).

Motion Patterns for Adaptive Sampling

A central objective in many underwater applications is to design provable and reliable mobile sensor networks for collecting the richest data set in an uncertain environment given limited resources. Consider the sampling of a single

time- and space-varying scalar field, like temperature T , using a network of vehicles, where the control problem is to coordinate the motion of the network to maximize information on this field over a given area or volume.

The definition of the information metric will depend on the application. If the data are to be assimilated into a high-resolution dynamical ocean model, then the metric would be defined by uncertainty as computed by the model. A general-purpose metric, based on objective analysis (linear statistical estimation from given field statistics), specifies the statistical uncertainty of the field model as a function of where and when the data were taken (Bennett 2002). The posteriori error $A(\mathbf{r}, t)$ is the variance of T about its estimate at location $\mathbf{r}$ and time t . Entropic information over a spatial domain of area $\mathcal{A}$ is

$$\mathcal{I}(t) = -\log \left(\frac{1}{\sigma_0 \mathcal{A}} \int d\mathbf{r} A(\mathbf{r}, t) \right),$$

where σ_0 is a scaling factor (Grocholsky 2002).

Computing coordinated trajectories to maximize $\mathcal{I}(t)$ can in principle be addressed using optimal coverage control methods. However, this coverage problem is especially challenging since the uncertainty field is spatially nonuniform and it changes with time and with the motion of the sampling vehicles. Furthermore, the optimal trajectories may become quite complex so that controlling vehicles to them in the presence of dynamic disturbances and uncertainty may lead to suboptimal performance.

An alternative approach decouples the design of motion patterns to optimize the entropic information metric from the decentralized control laws that stabilize the network onto the motion patterns (see Leonard et al. 2007). This approach was demonstrated with a network of 6 Slocum autonomous underwater gliders in a 24-day-long field experiment in Monterey Bay, California, in August 2006 (see Leonard et al. 2010). The coordinating feedback laws for the individual vehicles derive systematically from a control methodology that provides provable stabilization of a parameterized family of collective motion patterns (Sepulchre et al. 2008). These patterns con-

sist of vehicles moving on a finite set of closed curves with spacing between vehicles defined by a small number of “synchrony” parameters. The feedback laws that stabilize a given motion pattern use the same synchrony parameters that distinguish the desired pattern.

Each vehicle moves in response to the relative position and direction of its neighbors so that it keeps moving, it maintains the desired spacing, and it stays close to its assigned curve. It has been observed in the ocean, for vehicles carrying out this coordinated control law, that “when a vehicle on a curve is slowed down by a strong opposing flow field, it will cut inside a curve to make up distance and its neighbor on the same curve will cut outside the curve so that it does not overtake the slower vehicle and compromise the desired spacing” (Leonard et al. 2010). The approach is robust to vehicle failure since there are no leaders in the network, and it is scalable since the control law for each vehicle can be defined in terms of the state of a few other vehicles, independent of the total number of vehicles.

The control methodology prescribes steering laws for vehicles operated at a constant speed. Assume that the i th vehicle moves at unit speed in the plane in the direction $\theta_i(t)$ at time t . Then, the velocity of the i th vehicle is $\dot{\mathbf{x}}_i = (\cos \theta_i, \sin \theta_i)$. The steering control u_i is the component of the force in the direction normal to velocity, such that $\dot{\theta}_i = u_i$ for $i = 1, \dots, N$. Define

$$U(\theta_1, \dots, \theta_N) = \frac{N}{2} \|\mathbf{p}_\theta\|^2, \quad \mathbf{p}_\theta = \frac{1}{N} \sum_{j=1}^N \dot{\mathbf{x}}_j.$$

U is a potential function that is maximal at 1 when all vehicle directions are synchronized and minimal at 0 when all vehicle directions are perfectly anti-synchronized. Let $\tilde{\mathbf{x}}_i = (\tilde{x}_i, \tilde{y}_i) = (1/N) \sum_{j=1}^N \mathbf{x}_{ij}$ and let $\tilde{\mathbf{x}}_i^\perp = (-\tilde{y}_i, \tilde{x}_i)$. Define

$$S(\mathbf{x}_1, \dots, \mathbf{x}_N, \theta_1, \dots, \theta_N) = \frac{1}{2} \sum_{i=1}^N \|\dot{\mathbf{x}}_i - \omega_0 \tilde{\mathbf{x}}_i^\perp\|^2,$$

where $\omega_0 \neq 0$. S is a potential function that is minimal at 0 for circular motion of the vehicles

around their center of mass with radius $\rho_0 = |\omega_0|^{-1}$.

Define the steering control as

$$\dot{\theta}_i = \omega_0(1 + K_c \langle \tilde{\mathbf{x}}_i, \dot{\mathbf{x}}_i \rangle) - K_\theta \sum_{j=1}^N \sin(\theta_j - \theta_i),$$

where $K_c > 0$ and K_θ are scalar gains. Then, circular motion of the network is a steady solution, with the phase-locked heading arrangement a minimum of $K_\theta U$, i.e., synchronized or perfectly anti-synchronized depending on the sign of K_θ . Stability can be proved with the Lyapunov function $V_{c\theta} = K_c S + K_\theta U$. This steering control law depends only on relative position and relative heading measurements of the other vehicles.

The general form of the methodology extends the above control law to network interconnections defined by possibly time-varying graphs with limited sensing or communication links, and it provides systematic control laws to stabilize symmetric patterns of heading distributions about noncircular closed curves. It also allows for multiple graphs to handle multiple scales. For example, in the 2006 field experiment, the default motion pattern was one in which six gliders moved in coordinated pairs around three closed curves; one graph defined the smaller-scale coordination of each pair of gliders about its curve, while a second graph defined the larger-scale coordination of gliders across the three curves.

Implementation

Implementation of control of networks of underwater vehicles requires coping with the remote, hostile underwater environment. The control methodology for motion patterns and adaptive sampling, described above, was implemented in the field using a customized software infrastructure called the Glider Coordinated Control System (GCCS) (Paley et al. 2008). The GCCS combines a simple model for control planning with a detailed model of glider dynamics to accommodate the constant speed of gliders, relatively large ocean currents, waypoint tracking routines, communication only when gliders

surface (asynchronously), other latencies, and more. Other approaches consider control design in the presence of a flow field, formal methods to integrate high-resolution models of the flow field, and design tailored to propelled AUVs.

Summary and Future Directions

The multiscale, spatiotemporal dynamics of the underwater environment drive the need for well-coordinated control of networks of underwater vehicles that can manage the significant operational challenges of the opaque, uncertain, inhospitable, and dynamic oceans, lakes, and rivers. Control theory and algorithms have been developed to enable networks of vehicles to successfully operate as adaptable sensor arrays in missions that include feature tracking and adaptive sampling. Future work will improve control in the presence of strong and unpredictable flow fields and will leverage the latest in battery and underwater communication technologies. Hybrid vehicles and heterogeneous networks of vehicles will also promote advances in control. Future work will draw inspiration from the rapidly growing literature in decentralized cooperative control strategies and complex dynamic networks. Dynamics of decision-making teams of robotic vehicles and humans is yet another important direction of research that will impact the success of control of networks of underwater vehicles.

Cross-References

- [Connected and Automated Vehicles](#)
- [Motion Planning for Marine Control Systems](#)
- [Networked Systems](#)
- [Underactuated Marine Control Systems](#)

Recommended Reading

In Bellingham and Rajan (2007), it is argued that cooperative control of robotic vehicles is especially useful for exploration in remote and hostile

environments such as the deep ocean. A recent survey of robotics for environmental monitoring, including a discussion of cooperative systems, is provided in Dunbabin and Marques (2012). A survey of work on cooperative underwater vehicles is provided in Redfield (2013).

Bibliography

- Bachmayer R, Leonard NE (2002) Vehicle networks for gradient descent in a sampled environment. In: Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, pp 112–117
- Bellingham JG, Rajan K (2007) Robotics in remote and hostile environments. *Science* 318(5853):1098–1102
- Bennett A (2002) Inverse modeling of the ocean and atmosphere. Cambridge University Press, Cambridge
- Curtin TB, Bellingham JG, Catipovic J, Webb D (1993) Autonomous oceanographic sampling networks. *Oceanography* 6(3):86–94
- Dunbabin M, Marques L (2012) Robots for environmental monitoring: significant advancements and applications. *IEEE Robot Autom Mag* 19(1):24–39
- Fiorelli E, Leonard NE, Bhatta P, Paley D, Bachmayer R, Fratantoni DM (2006) Multi-AUV control and adaptive sampling in Monterey Bay. *IEEE J Ocean Eng* 31(4):935–948
- Grocholsky B (2002) Information-theoretic control of multiple sensor platforms. PhD thesis, University of Sydney
- Leonard NE, Paley DA, Lekien F, Sepulchre R, Fratantoni DM, Davis RE (2007) Collective motion, sensor networks, and ocean sampling. *Proc IEEE* 95(1):48–74
- Leonard NE, Paley DA, Davis RE, Fratantoni DM, Lekien F, Zhang F (2010) Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay. *J Field Robot* 27(6):718–740
- Ögren P, Fiorelli E, Leonard NE (2004) Cooperative control of mobile sensor networks: adaptive gradient climbing in a distributed environment. *IEEE Trans Autom Control* 49(8):1292–1302
- Paley D, Zhang F, Leonard NE (2008) Cooperative control for ocean sampling: the glider coordinated control system. *IEEE Trans Control Syst Technol* 16(4):735–744
- Redfield S (2013) Cooperation between underwater vehicles. In: Seto ML (ed) *Marine robot autonomy*. Springer, New York, pp 257–286
- Rudnick D, Davis R, Eriksen C, Fratantoni D, Perry M (2004) Underwater gliders for ocean research. *Mar Technol Soc J* 38(1):48–59
- Sepulchre R, Paley DA, Leonard NE (2008) Stabilization of planar collective motion with limited communication. *IEEE Trans Autom Control* 53(3):706–719
- Zhang F, Leonard NE (2010) Cooperative filters and control for cooperative exploration. *IEEE Trans Autom Control* 55(3):650–663

Control of Nonlinear Systems with Delays

Nikolaos Bekiaris-Liberis¹ and

Miroslav Krstic²

¹Department of Electrical and Computer Engineering, Technical University of Crete, Chania, Greece

²Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA, USA

Abstract

The reader is introduced to the predictor feedback method for the control of general nonlinear systems with input delays of arbitrary length. The delays need not necessarily be constant but can be time-varying or state-dependent. The predictor feedback methodology employs a model-based construction of the (unmeasurable) future state of the system. The analysis methodology is based on the concept of infinite-dimensional backstepping transformation – a transformation that converts the overall feedback system to a new, cascade “target system” whose stability can be studied with the construction of a Lyapunov function.

Keywords

Distributed parameter systems · Delay systems · Backstepping · Lyapunov function

Nonlinear Systems with Input Delay

Nonlinear systems of the form

$$\dot{X}(t) = f(X(t), U(t - D(t, X(t))))), \quad (1)$$

where $t \in \mathbb{R}_+$ is time, $f : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ is a vector field, $X \in \mathbb{R}^n$ is the state, $D : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}_+$ is a nonnegative function of the state of the system, and $U \in \mathbb{R}$ is the scalar input, which are ubiquitous in applications.

The starting point for designing a control law for (1), as well as for analyzing the dynamics of (1), is to consider the delay-free counterpart of (1), i.e., when $D = 0$, for which a plethora of results exists dealing with its stabilization and Lyapunov-based analysis (Krstic et al. 1995).

Systems of the form (1) constitute more realistic models for physical systems than delay-free systems. The reason is that often in engineering applications, the control that is applied to the system does not immediately affect the system. This dead time until the controller can affect the system might be due to, among other things, the long distance of the controller from the system, such as in networked control systems, or due to finite-speed transport or flow phenomena, such as in additive manufacturing and cooling systems, or due to various after-effects, such as in population dynamics.

The first step toward control design and analysis for system (1) is to consider the special case in which $D = \text{const}$. The next step is to consider the special case of system (1), in which $D = D(t)$, i.e., the delay is an a priori given function of time. Systems with time-varying delays model numerous real-world systems, such as networked control systems, traffic systems, or irrigation channels. Assuming that the input delay is an a priori defined function of time is a plausible assumption for some applications. Yet, the time variation of the delay might be the result of the variation of a physical quantity that has its own dynamics, such as in milling processes (due to speed variations), 3D printers (due to distance variations), cooling systems (due to flow rate variations), and population dynamics (due to population's size variations). Processes in this category can be modeled by systems with a delay that is a function of the state of the system, i.e., by (1) with $D = D(X)$.

In this article control designs are presented for the stabilization of nonlinear systems with input delays, with delays that are constant (Krstic 2009), time-varying (Bekiaris-Liberis and Krstic 2012), or state-dependent (Bekiaris-Liberis and Krstic 2013b), employing predictor feedback, i.e., employing a feedback law that uses the future rather than the current state of the system. Since

one employs in the feedback law the future values of the state, the predictor feedback completely cancels (compensates) the input delay, i.e., after the control signal reaches the system, the state evolves as if there were no delay at all. Since the future values of the state are not a priori known, the main control challenge is the implementation of the predictor feedback law. Having determined the predictor, the control law is then obtained by replacing the current state in a nominal state-feedback law (which stabilizes the delay-free system) by the predictor.

A methodology is presented in the article for the stability analysis of the closed-loop system under predictor feedback by constructing Lyapunov functionals. The Lyapunov functionals are constructed for a transformed (rather than the original) system. The transformed system is, in turn, constructed by transforming the original actuator state $U(\theta)$, $\theta \in [t - D, t]$ to a transformed actuator state with the aid of an infinite-

dimensional backstepping transformation. The overall transformed system is easier to analyze than the original system because it is a cascade, rather than a feedback system, consisting of a delay line with zero input, whose effect fades away in finite time, namely, after D time units, cascaded with an asymptotically stable system.

Predictor Feedback

The predictor feedback designs are based on a feedback law $U(t) = \kappa(X(t))$ that renders the closed-loop system $\dot{X} = f(X, \kappa(X))$ globally asymptotically stable. For stabilizing system (1), the following control law is employed instead:

$$U(t) = \kappa(P(t)), \quad (2)$$

where

$$P(\theta) = X(t) + \int_{t-D(t, X(t))}^{\theta} \frac{f(P(s), U(s))}{1 - D_t(\sigma(s), P(s)) - \nabla D(\sigma(s), P(s)) f(P(s), U(s))} ds \quad (3)$$

$$\sigma(\theta) = t + \int_{t-D(t, X(t))}^{\theta} \frac{1}{1 - D_t(\sigma(s), P(s)) - \nabla D(\sigma(s), P(s)) f(P(s), U(s))} ds, \quad (4)$$

for all $t - D(t, X(t)) \leq \theta \leq t$. The signal P is the predictor of X at the appropriate prediction time σ , i.e., $P(t) = X(\sigma(t))$. This fact is explained in more detail in the next paragraphs of this section. The predictor employs the future values of the state X which are not a priori available. Therefore, for actually implementing the feedback law (2), one has to employ (3). Relation (3) is a formula for the future values of the state that depends on the available measured quantities, i.e., the current state $X(t)$ and the history of the actuator state $U(\theta)$, $\theta \in [t - D(t, X(t)), t]$. To make clear the definitions of the predictor P and the prediction time σ , as well as their implementation through formulas (3) and (4), the constant delay case is discussed first.

The idea of predictor feedback is to employ in the control law the future values of the state at

the appropriate future time, such that the effect of the input delay is completely canceled (compensated). Define the quantity $\phi(t) = t - D$, which from now on is referred to as the delayed time. This is the time instant at which the control signal that currently affects the system was actually applied. To cancel the effect of this delay, the control law (2) is designed such that $U(\phi(t)) = U(t - D) = \kappa(X(t))$, i.e., such that $U(t) = \kappa(X(\phi^{-1}(t))) = \kappa(X(t + D))$. Define the prediction time σ through the relation $\phi^{-1}(t) = \sigma(t) = t + D$. This is the time instant at which an input signal that is currently applied actually affects the system. In the case of a constant delay, the prediction time is simply D time units in the future. Next an implementable formula for $X(\sigma(t)) = X(t + D)$ is derived. Performing a change of variables $t = \theta + D$, for

all $t - D \leq \theta \leq t$ in $\dot{X}(t) = f(X(t), U(t - D))$ and integrating in θ starting at $\theta = t - D$, one can conclude that P defined by (3) with $D_t = \nabla Df = 0$ and $D = \text{const}$ is the D time units ahead predictor of X , i.e., $P(t) = X(\sigma(t)) = X(t + D)$.

To better understand definition (3), the case of a linear system with a constant input delay D , i.e., a system of the form $\dot{X}(t) = AX(t) + BU(t - D)$, is considered next (see also ► [Control of Linear Systems with Delays](#) and Hale and Lunel 1993). In this case, the predictor $P(t)$ is given explicitly using the variation of constants formula, with the initial condition $P(t - D) = X(t)$, as $P(t) = e^{AD}X(t) + \int_{t-D}^t e^{A(t-\theta)}BU(\theta)d\theta$. For systems that are nonlinear, $P(t)$ cannot be written explicitly, for the same reason that a nonlinear ODE cannot be solved explicitly. So $P(t)$ is represented implicitly using the nonlinear integral equation (3). The computation of $P(t)$ from (3) is straightforward with a discretized implementation in which $P(t)$ is assigned values based on the right-hand side of (3), which involves earlier values of P and the values of the input U .

The case $D = D(t)$ is considered next. As in the case of constant delays, the main goal is to implement the predictor P . One needs first to define the appropriate time interval over which the predictor of the state is needed, which, in the constant delay case, is simply D time units in the future. The control law has to satisfy $U(\phi(t)) = \kappa(X(t))$ or $U(t) = \kappa(X(\sigma(t)))$. Hence, one needs to find an implementable formula for $P(t) = X(\sigma(t))$. In the constant delay case, the prediction horizon over which one needs to compute the predictor can be determined based on the knowledge of the delay time since the prediction horizon and the delay time are both equal to D . This is not anymore true in the time-varying case in which the delayed time is defined as $\phi(t) = t - D(t)$, whereas the prediction time as $\phi^{-1}(t) = \sigma(t) = t + D(\sigma(t))$. Employing a change of variables in $\dot{X}(t) = f(X(t), U(t - D(t)))$ as $t = \sigma(\theta)$, for all $\phi(t) \leq \theta \leq t$, and integrating in θ starting at $\theta = \phi(t)$, one obtains the formula for P given by (3) with $D_t = D'(\sigma(t))$, $\nabla Df = 0$, and $D = D(t)$.

Next the case $D = D(X(t))$ is considered. First one has to determine the predictor, i.e., the signal P such that $P(t) = X(\sigma(t))$, where $\sigma(t) = \phi^{-1}(t)$ and $\phi(t) = t - D(X(t))$. In the case of state-dependent delay, the prediction time $\sigma(t)$ depends on the predictor itself, i.e., the time when the current control reaches the system depends on the value of the state at that time, namely, the following implicit relationship holds $P(t) = X(t + D(P(t)))$ (and $X(t) = P(t - D(X(t)))$). This implicit relation can be solved by proceeding as in the time-varying case, i.e., by performing the change of variables $t = \sigma(\theta)$, for all $t - D(X(t)) \leq \theta \leq t$ in $\dot{X}(t) = f(X(t), U(t - D(X(t))))$, and integrating in θ starting at $\theta = t - D(X(t))$, to obtain the formula (3) for P with $D_t = 0$, $\nabla Df = \nabla D(P(s))f(P(s), U(s))$, and $D = D(X(t))$.

Analogously, one can derive the predictor for the case $D = D(t, X(t))$ with the difference that now the prediction time is not given explicitly in terms of P , but it is defined through an implicit relation, namely, it holds that $\sigma(t) = t + D(\sigma(t), P(t))$. Therefore, for actually computing σ , one has to proceed as in the derivation of P , i.e., to differentiate relation $\sigma(\theta) = \theta + D(\sigma(\theta), P(\theta))$ and then integrate starting at the known value $\sigma(t - D(t, X(t))) = t$. It is important to note that the integral equation (4) is needed in the computation of P only when D depends on both X and t .

Backstepping Transformation and Stability Analysis

The predictor feedback designs are based on a feedback law $\kappa(X)$ that renders the closed-loop system $\dot{X} = f(X, \kappa(X))$ globally asymptotically stable. However, in the rest of the section, it is assumed that the feedback law $\kappa(X)$ renders the closed-loop system $\dot{X} = f(X, \kappa(X) + v)$ input-to-state stable (ISS) with respect to v , i.e., there exists a smooth function $S : \mathbb{R}^n \rightarrow \mathbb{R}_+$ and class $\mathcal{K}_\infty$ functions $\alpha_1, \alpha_2, \alpha_3$, and α_4 such that

$$\begin{aligned} \alpha_3(|X(t)|) &\leq S(X(t)) \\ &\leq \alpha_4(|X(t)|) \end{aligned} \quad (5)$$

$$\begin{aligned} & \frac{\partial S(X(t))}{\partial X} f(X(t), \kappa(X(t)) + v(t)) \\ & \leq -\alpha_1(|X(t)|) + \alpha_2(|v(t)|). \end{aligned} \quad (6)$$

Imposing this stronger assumption enables one to construct a Lyapunov functional for the closed-loop systems (1)–(4) with the aid of the Lyapunov characterization of ISS defined in (5) and (6).

The stability analysis of the closed-loop systems (1)–(4) is explained next. Denote the infinite-dimensional backstepping transformation of the actuator state as

$$\begin{aligned} W(\theta) &= U(\theta) - \kappa(P(\theta)), \\ &\text{for all } t - D(t, X(t)) \leq \theta \leq t, \end{aligned} \quad (7)$$

where $P(\theta)$ is given in terms of $U(\theta)$ from (3). Using the fact that $P(t - D(t, X(t))) = X(t)$, for all $t \geq 0$, one gets from (7) that $U(t - D(t, X(t))) = W(t - D(t, X(t))) + \kappa(X(t))$. With the fact that for all $\theta \geq 0$, $U(\theta) = \kappa(P(\theta))$, one obtains from (7) that $W(\theta) = 0$, for all $\theta \geq 0$. Yet, for all $t \leq D(t, X(t))$, i.e., for all $\theta \leq 0$, $W(\theta)$ might be nonzero due to the effect of the arbitrary initial condition $U(\theta)$, $\theta \in [-D(0, X(0)), 0]$. With the above observations, one can transform system (1) with the aid of transformation (7) to the following target system:

$$\begin{aligned} \dot{X}(t) &= f(X(t), \kappa(X(t)) \\ &\quad + W(t - D(t, X(t)))) \end{aligned} \quad (8)$$

$$\begin{aligned} W(t - D(t, X(t))) &= 0, \\ &\text{for } t - D(t, X(t)) \geq 0. \end{aligned} \quad (9)$$

Using relations (5), (6), and (8), (9), one can construct the following Lyapunov functional for showing asymptotic stability of the target system (8), (9), i.e., for the overall system consisting of the vector $X(t)$ and the transformed infinite-dimensional actuator state $W(\theta)$, $t - D(t, X(t)) \leq \theta \leq t$,

$$V(t) = S(X(t)) + \frac{2}{c} \int_0^{L(t)} \frac{\alpha_2(r)}{r} dr, \quad (10)$$

where $c > 0$ is arbitrary and

$$L(t) = \sup_{t - D(t, X(t)) \leq \theta \leq t} \left| e^{c(\sigma(\theta) - t)} W(\theta) \right|. \quad (11)$$

With the invertibility of the backstepping transformation, one can then show global asymptotic stability of the closed-loop system in the original variables (X, U) . In particular, there exists a class $\mathcal{KL}$ function β such that

$$\begin{aligned} & |X(t)| + \sup_{t - D(t, X(t)) \leq \theta \leq t} |U(\theta)| \\ & \leq \beta \left(|X(0)| + \sup_{-D(0, X(0)) \leq \theta \leq 0} |U(\theta)|, t \right), \\ & \text{for all } t \geq 0. \end{aligned} \quad (12)$$

One of the main obstacles in designing globally stabilizing control laws for nonlinear systems with long input delays is the finite escape phenomenon. The input delay may be so large that the control signal cannot reach the system before its state grows unbounded. Therefore, one has to assume that the system $\dot{X} = f(X, \omega)$ is forward complete, i.e., for every initial condition and every bounded input signal, the corresponding solution is defined for all $t \geq 0$.

With the forward completeness requirement, estimate (12) holds globally for constant but arbitrary large delays. For the case of time-varying delays, estimate (12) holds globally as well but under the following four conditions on the delay:

- C1. $D(t) \geq 0$. This condition guarantees the causality of the system.
- C2. $D(t) < \infty$. This condition guarantees that all inputs applied to the system eventually reach the system.
- C3. $\dot{D}(t) < 1$. This condition guarantees that the system never feels input values that are older than the ones it has already felt, i.e., the input signal's direction never gets reversed. (This condition guarantees the existence of $\sigma = \phi^{-1}$.)

- C4. $\dot{D}(t) > -\infty$ This condition guarantees that the delay cannot disappear instantaneously but only gradually.

In the case of state-dependent delays, the delay depends on time as a result of its dependency on the state. Therefore, predictor feedback guarantees stabilization of the system when the delay satisfies the four conditions C1–C4. Yet, since the delay is a nonnegative function of the state, conditions C2–C4 are satisfied by restricting the initial state X and the initial actuator state. Therefore estimate (12) holds locally.

Cross-References

- [Control of Linear Systems with Delays](#)

Recommended Reading

The main control design tool for general systems with input delays of arbitrary length is predictor feedback. The reader is referred to Artstein (1982) for the first systematic treatment of general linear systems with constant input delays. The applicability of predictor feedback was extended in Krstic (2009) to several classes of systems, such as nonlinear systems with constant input delays and linear systems with unknown input delays. Subsequently, predictor feedback was extended to general nonlinear systems with nonconstant input and state delays (Bekiaris-Liberis and Krstic 2013a). The main stability analysis tool for systems employing predictor feedback is backstepping. Backstepping was initially introduced for adaptive control of finite-dimensional nonlinear systems (Krstic et al. 1995). The continuum version of backstepping was originally developed for the boundary control of several classes of PDEs in Krstic and Smyshlyaev (2008).

Bibliography

Artstein Z (1982) Linear systems with delayed controls: a reduction. *IEEE Trans Autom Control* 27:869–879

- Bekiaris-Liberis N, Krstic M (2012) Compensation of time-varying input and state delays for nonlinear systems. *J Dyn Syst Meas Control* 134:011009
- Bekiaris-Liberis N, Krstic M (2013a) Nonlinear control under nonconstant delays. SIAM, Philadelphia
- Bekiaris-Liberis N, Krstic M (2013b) Compensation of state-dependent input delay for nonlinear systems. *IEEE Trans Autom Control* 58:275–289
- Hale JK, Verduyn Lunel SM (1993) Introduction to functional differential equations. Springer, New York
- Krstic M (2009) Delay compensation for nonlinear, adaptive, and PDE systems. Birkhauser, Boston
- Krstic M, Kanellakopoulos I, Kokotovic PV (1995) Nonlinear and adaptive control design. Wiley, New York
- Krstic M, Smyshlyaev A (2008) Boundary control of PDEs: a course on backstepping designs. SIAM, Philadelphia

Control of Optical Tweezers

Murti V. Salapaka
Department of Electrical and Computer
Engineering, University of Minnesota
Twin-Cities, Minneapolis, MN, USA

Abstract

Optical tweezers have revolutionized how matter at the micrometer and nanometer scale can be manipulated and interrogated. In this chapter, operating principles of optical tweezers are presented, with a description of the instrumentation employed. Traditional control approaches and modes are described. Modern control methodologies are motivated, and, engineering objectives framed in the modern model based framework are illustrated via an application to motor-protein interrogation.

Keywords

Optical tweezers · Optical traps · Model based control · Single molecule

Introduction: What Are Optical Tweezers

Optical tweezers use radiation pressure of a highly convergent laser beam to trap micron- and

smaller-sized particles in three-dimensional potential wells. In 1970, Arthur Ashkin showed that momentum of laser beams can be used to accelerate and trap individual particles. In a seminal paper in 1986, Ashkin and his collaborators demonstrated that if a highly convergent laser beam formed by passing the laser through a high-numerical-aperture lens is targeted on a particle of appropriate size and optical characteristics, then the beam instead of pushing the particle away would counterintuitively trap the particle near the focus of the lens. The first optical tweezer was realized. Here a change in the laser direction altered the position of the focus and the trap location pulling the particle along. Ashkin won the Nobel Prize for Physics in 2018 largely due to the impact of optical tweezers in diverse applications including in biology for sorting cells (Zhang and Liu 2008) and in physics for exerting and measuring extremely small forces (Chu et al. 1986). Perhaps the most impact of optical tweezers is in its use in measuring and investigating behavior of single biological molecules such as molecular motors (Fazal and Block 2011). With its femto-Newton force resolution, a spatial resolution of less than a nanometer and temporal resolution with millisecond time scales, it has become possible to study the dynamics of motor proteins, such as myosin, kinesin, and dynein.

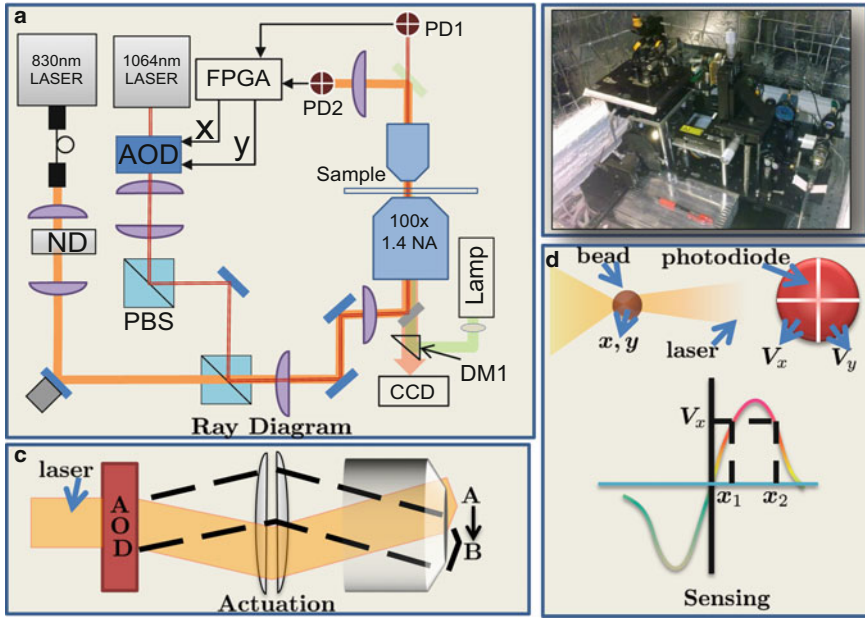
A basic modern optical tweezer setup is described by Fig. 1. Here, a trapping laser passes through an acousto-optical deflector (AOD), a polarizing beam splitter (PBS), and other optical elements that direct the laser into a high-numerical objective that focuses the laser onto the sample plane where the particle to be trapped is placed. The trapping laser exits the sample plane passing through the particle and is incident on a photodiode (PD1). Another laser, the detection laser, of considerably lower power travels through a separate PBS, passes through the particle and is incident on another photodiode (PD2). Here the location of the particle determines where the detection laser is incident on the photodiode, PD2, and thus measures the location of the bead (illustrated in Fig. 1b). The photodiode measurements, PD1 and

PD2, are processed by a field-programmable gate array which uses the measurements to determine the control signal actuated by the acousto-optical deflector.

Sensing and Actuation

Sensing: Photodiodes for measuring the position of the trapped particles were employed very early for feedback purposes (see Ashkin and Dziedzic 1977). The bandwidth and resolution afforded by photodiodes with sub-millisecond response and sub-nanometer resolution are more than adequate for most applications and are not a limiting factor in the achievable performance from the optical tweezer setups. One challenge for photodiode-based measurement is the nonlinear nature of the sensed signal. Here the position of the detection laser passing through the particle depends on whether the laser is incident near the center of the particle or near the extremities of the particle; when the particle position is such that the laser is positioned near the center, the measured signal varies linearly with the position of the particle and nonlinearly away from the center (see Fig. 1d). Learning approaches can be employed to improve the range of linear behavior (Aggarwal and Salapaka 2010).

Actuation: In modern optical tweezer setups, the direction of the trapping beam is controlled by electro-optic deflectors (EODs) or acousto-optical deflectors (AODs) with EODs providing better bandwidth with less range of motion than AODs. Both these actuation mechanisms afford bandwidths that are orders higher in relation to the time constants of the dynamics of the particle. Indeed, these devices provide high enough bandwidth whereby it is possible to trap multiple particles simultaneously, by multiplexing the laser. In AODs which are typically more prevalent in use, a diffraction grating is created by propagating waves with frequency in the acoustic range via a crystal. The specific frequency of the acoustic wave employed determines the spacing of the grating which in turn determines how much the trapping laser gets diffracted. Thus by changing the frequency in the radio-frequency range, it is



Control of Optical Tweezers, Fig. 1 Illustration of a modern optical tweezer setup

possible to steer the trapping laser's direction. Here the change in the direction of the trapping laser is completed only when the acoustic wave has traveled through the entire crystal. Other non-idealities are caused by the partial reflection of the waves at the crystal boundaries (Klein and Cook 1967). Other means of altering the trap position is to use piezo-mirrors which have the advantage of large range when compared to AODs and EODs but severely compromised bandwidth.

Models and Identification

The analysis of optical forces exerted on a particle admits two regimes: in the first regime, the particle is much smaller than the wavelength of the light, and in the second regime, the particle is larger than the wavelength. When the particle is in the Rayleigh regime with the particle small, the particle is considered as an electric dipole in an electromagnetic field. If the particle size is much larger than the laser's wavelength (the Mie Regime), ray optics-based analysis is admissible (see Ashkin 1992). In both these regimes, first

principle-based analysis can explain why the particle gets trapped in an optical tweezer. However, in a vast range of applications, the particle size is in the intermediate range where the particle size is comparable to the laser's wavelength; in such cases obtaining models from first principles is hard. For small displacements from the trap center, the laser trap behaves similar to a Hookean spring where the dynamics of the particle in the trap admits the model of an over-damped system described by

$$\beta \dot{x}_b + k_T(x_b - x_T) = \eta + f_{\text{ext}} \quad (1)$$

where x_b is the position of the particle, k_T is the stiffness of the trap, x_T denotes the position of the trapping laser, η is white noise that models thermal noise, and f_{ext} is any other external force acting on the particle. Here the trap position x_T can be controlled using an AOD.

The AOD/actuator dynamics is characterized by a transfer function $A(s)$ whereby

$$\hat{x}_T = A(s)\hat{u}(s), \quad (2)$$

where $u(t)$ is the control signal provided to the actuator. The transfer function $A(s)$ due to the physics of the AOD includes right half plane zeros that impose fundamental limits on performance.

The photodiode-based sensing is modeled as

$$y = x_b + v$$

where v models measurement noise. Taking Laplace transforms we obtain

$$\hat{x}_b = \frac{k_T}{\beta s + k_T} x_T + \underbrace{\frac{1}{\beta s + k_T}}_{G_p(s)} (\eta + f_{\text{ext}}); \quad \hat{x}_T = A(s) \hat{u}(s). \quad (3)$$

Letting $G_p(s) = \frac{1}{\beta s + k_T}$, it follows that $G_h = k_T G_p(s) A(s)$ is the transfer function mapping the control signal u to the bead displacement and x_b , and, G_p is the transfer function from thermal noise η to x_b . By conducting an open-loop experiment, it is possible to identify the transfer function $G_p(s)$ where the response due to the white thermal noise excitation, η , can be employed. Moreover, it is possible to identify G_h using the control signal u as input and x_b as the output response. Using the identified G_h and G_T , it is possible to retrieve $A(s) = \frac{G_h}{k_T G_p}$. As alluded to earlier, $A(s)$ includes right half plane zeros imposing fundamental limitations (Sehgal et al. 2009).

Optical tweezers, apart from being used for manipulating micron-sized particles, provide sensing mechanisms for forces in the femto-Newton to sub-piconewton range as well as sensing motion in the nanometer regime. These operational modes of an optical trap have revolutionized measurement at the small scale, specifically, of single molecules.

Motor proteins such as kinesin and dynein are critical components of intracellular transport. Optical tweezers provide an efficient means of probing motor proteins via bead handles where a polystyrene bead (with radii in the

500 nm – 1 μ m range) is attached to the motor protein and the polystyrene bead is trapped in an optical trap. The optically trapped bead can then be placed on a microtubule; processive motor proteins under the presence of ATP “walk” on the microtubule where the motion of the motor protein can be investigated by monitoring and assessing the forces felt by the trapped bead and its motion (see Fig. 2). Here, often if the motor-protein extension is small, a reasonable model of for the motor protein is that the force it exerts on the bead is $f_{\text{ext}} = -k_m(x_b - x_m)$ where x_m denotes the position of motor protein with $x_b - x_m$ characterizing the extension of the protein and k_m the stiffness of the motor protein. Substituting $f_{\text{ext}} = -k_m(x_b - x_m)$, we obtain

$$\hat{x}_b(s) = k_m G(s) x_m + k_T G(s) x_T + G(s) \eta \quad (4)$$

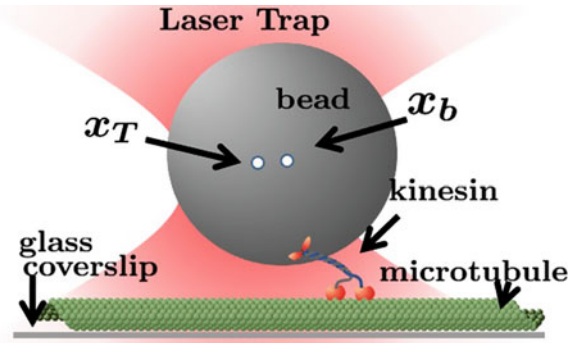
where $G(s) = \frac{1}{\beta s + k_m + k_T}$.

Traditional Control Paradigms

From the very time of its invention feedback has played a pivotal role in the operation of optical tweezers.

Optical Traps as Force Sensors: Isometric Clamps

Ashkin employed feedback to investigate optical levitation (Ashkin and Dziedzic 1977) where the position of the levitated particle was sensed using a photodiode and a reference position was maintained by altering the laser power (which formed the control signal) to counteract the force that was responsible for the displacement of the particle. The control signal, which is the change in the laser power, provided a measure of the forces on the particle. In Ashkin and Dziedzic (1977) the control signal gave a measure of the electric forces on oil drops thereby measuring single electron change accumulation. The underlying principle adopted is to reject the disturbance force (e.g., due to charge accumulation) that causes the displacement of the particle whereby the control signal rejecting the disturbance becomes



Control of Optical Tweezers, Fig. 2 The figure shows a bead in a trap with the laser focussed at x_T with the bead position, x_b . A motor protein, kinesin, attached to the bead is shown as moving the “cargo” (the bead), along a microtubule placed on a glass coverslip. A high-

resolution method to study motor-protein motion is to optically trap the cargo (the bead) carried by the motor protein while it walks on microtubule. The bead position x_b is changed in response to the change in trap position x_T and the force exerted by the kinesin as it moves

an estimate of the disturbance. The underlying principle is quite powerful; the added benefit is that the displacement is maintained near a desired operating value (the regulated value) wherein a linear analysis and behavior ensue.

In traditional designs the controller employed is $K(s)$ with $\hat{u} = K(s)(\hat{r} - \hat{y})$ where r is the reference position to be regulated. The controllers are primarily proportional or proportional-integral. Traditional practice for designing controllers has not employed models. When estimating external forces, $\hat{f}_{\text{est}} = -k_T u$ is considered as an estimate of f_{ext} . Note that the closed-loop dynamics is described by

$$\begin{aligned}\hat{x}_b &= \frac{k_T G_p A K}{1 + k_T G_p A K} \hat{r} \\ &+ \frac{G_p}{1 + k_T G_p A K} (\hat{\eta} + \hat{f}_{\text{ext}}) \\ &- \frac{k_T G_p A K}{1 + k_T G_p A K} \hat{v},\end{aligned}$$

with

$$\hat{f}_{\text{est}} = -k_T u = -k_T K(\hat{r} - \hat{y}) = k_T K \hat{x}_b,$$

where the position x_b is regulated to zero. Thus the transfer function from f_{ext} to its estimate is

$$\hat{f}_{\text{est}} = \frac{k_T K G_p}{1 + k_T G_p A K} \hat{f}_{\text{ext}}.$$

In traditional schemes for isometric clamps (Finer et al. 1994), the actuator dynamics is ignored (thus $A = 1$) and the assumption that $\frac{k_T K G_p}{1 + k_T G_p A K} \approx 1$ (which holds only at frequencies where the $|(k_T G_p A K)j\omega| \gg 1$) are assumed to hold. The underlying assumptions of the design limit the operation to low-bandwidth operation. Moreover, the behavior due to right hand plane zeros in A manifest in experiments that remained unexplained (see Aggarwal et al. 2011; Sehgal et al. 2009).

Optical Traps as Motion Sensors: Isotonic Clamps

In many applications it is important to regulate a constant force on the system being investigated. The study of motor proteins illustrates the need for regulating a constant force. In the study of motor proteins, a polystyrene bead is attached to the motor protein which traverses the microtubule (see Fig. 2). The bead position x_b increases as it is being pulled along the microtubule, whereby the bead is subjected to increasing forces and the motor protein will come under increasing forces as well. Most single molecules, motor proteins included, are characterized by the wormlike chain model where the stiffness of

the molecule changes with force. Thus the bead position, which is being measured, will provide an inaccurate estimate of the motor motion x_m . Moreover, transport properties of motor proteins are known to be altered when loading conditions change, thus making it difficult to study the effect of load forces on motor-protein behavior. Isotonic clamps regulate $x_b - x_T$ to a desired value e_d (Visscher and Block 1998; Lang et al. 2002) allowing the motor linkage to be modeled as Hookean spring. These methods are not model based and suffer from the similar drawbacks alluded to in relation to isometric clamps. Thus, traditional non-model-based designs (Visscher and Block 1998; Lang et al. 2002) are appropriate for low-bandwidth operation. A modern control approach alleviates challenges with traditional approaches outlined above.

Modern Control Approaches

Modern model-based control approaches allow for controller synthesis that address multiple objectives cast in different measures. Moreover it is possible to regulate variables that are not measured. In what follows, such a modern control approach is illustrated via a design of an isotonic clamp in the study of motor proteins. The main objectives for investigating motor-protein behavior are:

1. The force on the bead in the trap is to be regulated to a set-point F_d . This objective captures the concerns of maintaining the trap dynamics in the linear regime and be able to mirror the regulation goals in traditional schemes. Note that x_b and x_T are available as measured variables. Here, $k_T(x_T - x_b)$ which is the force acting on the bead due to the trap is to be maintained at F_d .
2. The force $k_m(x_m - x_b)$ on the motor protein is to be regulated to a desired value F_m . Note here that x_m is not a measured quantity and thus difficult to directly incorporate this objective in traditional approaches.
3. An accurate estimate x_{em} of the motor position x_m is to be determined.

The exogenous inputs are F_d , F_m , η , and x_m which are the force to be regulated on the bead, force to be regulated on the motor protein, thermal noise, and motor-protein position x_m (Fig. 3).

The regulated variables are weighted error, $z_e = W_e(F_d - k_T(x_b - x_T))$, in regulating forces on the bead, weighted error $z_{em} = W_{em}(F_m - k_m(x_m - x_b))$ in regulating forces on the motor, weighted error in motor motion estimate $z_m = W_m(x_m - x_{em})$, and weighted motor motion estimate $z_n = W_\eta x_{em}$. The weights W_e , W_{em} , W_m , and W_η allow for emphasizing different frequency ranges of relevant objectives they target.

The controller output $\tilde{u}$, apart from providing the actuation input u to the actuator, provides an estimate x_{em} of the motor motion x_m . The controller is described by $\begin{pmatrix} u \\ x_{em} \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \begin{pmatrix} F_d \\ x_b \end{pmatrix}$. Using the identified models, P_{sys} can be determined. The associated optimization problem to synthesize a controller $K(s)$ is given below :

$$\begin{aligned} \min_{K \text{ stabilizing}, \gamma} \{ & \gamma : \| [T_{z_e f_d} \ T_{z_e x_m}] \|_{H_\infty} < \gamma, \\ & \| [T_{z_{em} x_m}] \|_{H_\infty} < \gamma, \\ & \| T_{z_m x_m} \|_{H_\infty} < \gamma, \\ & \| T_{z_\eta \eta} \|_{H_2}^2 < \nu \} \end{aligned} \quad (5)$$

Here,

$$\begin{aligned} T_{z_e f_d} &= W_e(s)(1 - Gk_T^2 K_{11}(s)A(s)H(s) \\ &\quad + A(s)k_T K_{11}(s)H(s)), \\ T_{z_e x_m} &= W_e(s)k_T G(s)k_m \\ &\quad ((A(s)K_{12}(s) - 1)H(s)), \\ T_{z_{em} x_m} &= W_{em}(s)k_m G(s)k_T A(s)K_{12}(s)H(s), \\ T_{z_m x_m} &= W_m(s)(1 - G(s)k_m K_{22}(s))H(s), \text{ and,} \\ T_{z_\eta \eta} &= W_\eta G(s)K_{22}(s)H(s). \end{aligned} \quad (6)$$

- [Control Systems for Nanopositioning](#)
- [Non-raster Methods in Scanning Probe Microscopy](#)

Bibliography

- Aggarwal T, Salapaka M (2010) Real-time nonlinear correction of back-focal-plane detection in optical tweezers. *Rev Sci Instrum* 81(12):123105
- Aggarwal T, Sehgal H, Salapaka M (2011) Robust control approach to force estimation in a constant position optical tweezers. *Rev Sci Instrum* 82(11):115108
- Ashkin A (1970) Acceleration and trapping of particles by radiation pressure. *Phys Rev Lett* 24(4):156–159. <https://doi.org/10.1103/PhysRevLett.24.156>
- Ashkin A (1992) Forces of a single-beam gradient laser trap on a dielectric sphere in the ray optics regime. *Biophys J* 61(2):569–582. <http://www.biophysj.org/cgi/content/abstract/61/2/569>, <http://www.biophysj.org/cgi/reprint/61/2/569.pdf>
- Ashkin A, Dziedzic JM (1977) Feedback stabilization of optically levitated particles. *Appl Phys Lett* 30(4):202–204. <https://doi.org/10.1063/1.89335>, <http://link.aip.org/link/?APL/30/202/1>
- Ashkin A, Dziedzic JM, Bjorkholm JE, Chu S (1986) Observation of a single-beam gradient force optical trap for dielectric particles. *Opt Lett* 11(5):288. <http://ol.osa.org/abstract.cfm?URI=ol-11-5-288>
- Bhaban S, Talukdar S, Li M, Hays T, Seiler P, Salapaka M (2018) Single molecule studies enabled by model-based controller design. *IEEE/ASME Trans Mechatron* 23(4):1532–1542
- Chu S, Bjorkholm JE, Ashkin A, Cable A (1986) Experimental observation of optically trapped atoms. *Phys Rev Lett* 57(3):314–317. <https://doi.org/10.1103/PhysRevLett.57.314>
- Fazal FM, Block SM (2011) Optical tweezers study life under tension. *Nature Photonics* 5(6):318
- Finer JT, Simmons RM, Spudich JA (1994) Single myosin molecule mechanics: piconewton forces and nanometre steps. *Trends Cell Biol* 368:113–119. <https://doi.org/10.1038/368113a0>
- Klein W, Cook BD (1967) Unified approach to ultrasonic light diffraction. *IEEE Trans Sonics Ultrason* 14(3):123–134
- Lang MJ, Asbury CL, Shaevitz JW, Block SM (2002) An automated two-dimensional optical force clamp for single molecule studies. *Biophys J* 83(1):491–501. <http://www.biophysj.org/cgi/content/abstract/83/1/491>, <http://www.biophysj.org/cgi/reprint/83/1/491.pdf>
- Scherer C, Gahinet P, Chilali M (1997) Multiobjective output-feedback control via LMI optimization. *IEEE Trans Autom Control* 42(7):896–911
- Sehgal H, Aggarwal T, Salapaka M (2009) High bandwidth force estimation for optical tweezers. *Appl Phys Lett* 94(15):153114

- Visscher K, Block S (1998) Versatile optical traps with feedback control. *Methods Enzymol* 298:460–489
- Zhang H, Liu KK (2008) Optical tweezers for single cells. *J R Soc Interface* 5(24):671–690

Control of Quantum Systems

Ian R. Petersen

Research School of Electrical, Energy and Materials Engineering, Australian National University, Canberra, ACT, Australia

Abstract

Quantum control theory is concerned with the control of systems whose dynamics are governed by the laws of quantum mechanics. Quantum control may take the form of open-loop quantum control or quantum feedback control. Also, quantum feedback control may consist of measurement-based feedback control, in which the controller is a classical system governed by the laws of classical physics. Alternatively, quantum feedback control may take the form of coherent feedback control in which the controller is a quantum system governed by the laws of quantum mechanics. In the area of open-loop quantum control, questions of controllability along with optimal control and Lyapunov control methods are discussed. In the case of quantum feedback control, LQG and H^∞ control methods are discussed.

Keywords

Quantum control · Quantum controllability · Measurement-based quantum feedback · Coherent quantum feedback

Introduction

Quantum control is the control of systems whose dynamics are described by the laws of quantum

This work was supported by the Australian Research Council under grants FL11010002 and DP180101805.

physics rather than classical physics. The dynamics of quantum systems must be described using quantum mechanics which allows for uniquely quantum behavior such as entanglement and coherence. There are two main approaches to quantum mechanics which are referred to as the Schrödinger picture and the Heisenberg picture. In the Schrödinger picture, quantum systems are modelled using the Schrödinger equation or a master equation which describe the evolution of the system state or density operator. In the Heisenberg picture, quantum systems are modelled using quantum stochastic differential equations which describe the evolution of system observables. These different approaches to quantum mechanics lead to different approaches to quantum control. Important areas in which quantum control problems arise include physical chemistry, atomic and molecular physics, and optics. Detailed overviews of the field of quantum control can be found in the survey papers Dong and Petersen (2010); Brif et al. (2010) and the monographs Wiseman and Milburn (2010), D'Alessandro (2007), Jacobs (2014), and Nurdin and Yamamoto (2017).

A fundamental problem in a number of approaches to quantum control is the controllability problem. Quantum controllability problems are concerned with finite dimensional quantum systems modelled using the Schrödinger picture of quantum mechanics and involve the structure of corresponding Lie groups or Lie algebras; e.g., see D'Alessandro (2007). These problems are typically concerned with closed quantum systems which are quantum systems isolated from their environment. For a controllable quantum system, an open loop control strategy can be constructed in order to manipulate the quantum state of the system in a general way. Such open loop control strategies are referred to as coherent control strategies. Time-optimal control is one method of constructing these control strategies which has been applied in applications including physical chemistry and in nuclear magnetic resonance systems; e.g., see Khaneja et al. (2001). An alternative approach to open loop quantum control is the Lyapunov approach; e.g., see Wang and Schirmer (2010). This approach extends the

classical Lyapunov control approach in which a control Lyapunov function is used to construct a stabilizing state feedback control law. However in quantum control, state feedback control is not allowed since classical measurements change the quantum state of a system and the Heisenberg uncertainty principle forbids the simultaneous exact classical measurement of non-commuting quantum variables. Also, in many quantum control applications, the timescales are such that real-time classical measurements are not technically feasible. Thus, in order to obtain an open loop control strategy, the deterministic closed loop system is simulated as if the state feedback control were available, and this enables an open loop control strategy to be constructed. As an alternative to coherent open loop control strategies, some classical measurements may be introduced leading to incoherent control strategies; e.g., see Dong et al. (2009).

In addition to open loop quantum control approaches, a number of approaches to quantum control involve the use of feedback; e.g., see Wiseman and Milburn (2010) and Nurdin and Yamamoto (2017). This quantum feedback may either involve the use of classical measurements, in which case the controller is a classical (non-quantum) system, or involve the case where no classical measurements are used since the controller itself is a quantum system. The case in which the controller itself is a quantum system is referred to as coherent quantum feedback control; e.g., see Lloyd (2000) and James et al. (2008). Quantum feedback control may be considered using the Schrödinger picture in which case, the quantum systems under consideration are modelled using stochastic master equations. Alternatively using the Heisenberg picture, the quantum systems under consideration are modelled using quantum stochastic differential equations. Applications in which quantum feedback control can be applied include quantum optics and atomic physics. In addition, quantum control can potentially be applied to problems in quantum information (e.g., see Nielsen and Chuang 2000) such as quantum error correction (e.g., see Kerckhoff et al. 2010) or the preparation of quantum states. Quantum information and

quantum computing in turn have great potential in solving intractable computing problems such as factoring large integers using Shor's algorithm; see Shor (1994).

Schrödinger Picture Models of Quantum Systems

The state of a closed quantum system can be represented by a unit vector $|\psi\rangle$ in a complex Hilbert space $\mathcal{H}$. Such a quantum state is also referred to as a wave function. In the Schrödinger picture, the time evolution of the quantum state is defined by the Schrödinger equation which is in general a partial differential equation. An important class of quantum systems are finite-level systems in which the Hilbert space is finite dimensional. In this case, the Schrödinger equation is a linear ordinary differential equation of the form

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = H_0 |\psi(t)\rangle$$

where H_0 is the free Hamiltonian of the system, which is a self-adjoint operator on $\mathcal{H}$; e.g., see Merzbacher (1970). Also, $\hbar$ is the reduced Planck's constant, which can be assumed to be one with a suitable choice of units. In the case of a controlled closed quantum system, this differential equation is extended to a bilinear ordinary differential equation of the form

$$i \frac{\partial}{\partial t} |\psi(t)\rangle = \left[H_0 + \sum_{k=1}^m u_k(t) H_k \right] |\psi(t)\rangle \quad (1)$$

where the functions $u_k(t)$ are the control variables and the H_k are corresponding control Hamiltonians, which are also assumed to be self-adjoint operators on the underlying Hilbert space. These models are used in the open loop control of closed quantum systems.

To represent open quantum systems, it is necessary to extend the notion of quantum state to density operators ρ which are positive operators with trace one on the underlying Hilbert space $\mathcal{H}$. In this case, the Schrödinger picture model of a quantum system is given in terms of a master

equation which describes the time evolution of the density operator. In the case of an open quantum system with Markovian dynamics defined on a finite dimensional Hilbert space of dimension N , the master equation is a matrix differential equation of the form

$$\begin{aligned} \dot{\rho}(t) = & -i \left[\left(H_0 + \sum_{k=1}^m u_k(t) H_k \right), \rho(t) \right] \\ & + \frac{1}{2} \sum_{j,k=0}^{N^2-1} \alpha_{j,k} \left([F_j \rho(t), F_k^\dagger] \right. \\ & \left. + [F_j, \rho(t) F_k^\dagger] \right); \end{aligned} \quad (2)$$

e.g., see Breuer and Petruccione (2002). Here the notation $[X, \rho] = X\rho - \rho X$ refers to the commutation operator, and the notation † denotes the adjoint of an operator. Also, $\{F_j\}_{j=0}^{N^2-1}$ is a basis set for the space of bounded linear operators on $\mathcal{H}$ with $F_0 = I$. Also, the matrix $A = (\alpha_{j,k})$ is assumed to be positive definite. These models, which include the Lindblad master equation for dissipative quantum systems as a special case (e.g., see Wiseman and Milburn 2010), are used in the open loop control of finite-level Markovian open quantum systems.

In quantum mechanics, classical measurements are described in terms of self-adjoint operators on the underlying Hilbert space referred to as observables; e.g., see Breuer and Petruccione (2002). An important case of measurements is projective measurements in which an observable M is decomposed as $M = \sum_{k=1}^m k P_k$ where the P_k are orthogonal projection operators on $\mathcal{H}$; e.g., see Nielsen and Chuang (2000). Then, for a closed quantum system with quantum state $|\psi\rangle$, the probability of an outcome k from the measurement is given by $\langle \psi | P_k | \psi \rangle$ which denotes the inner product between the vector $|\psi\rangle$ and the vector $P_k |\psi\rangle$. This notation is referred to as Dirac notation and is commonly used in quantum mechanics. If the outcome of the quantum measurement is k , the state of the quantum system collapses to the new value of $\frac{P_k |\psi\rangle}{\sqrt{\langle \psi | P_k | \psi \rangle}}$. This change in the quantum

state as a result of a measurement is an important characteristic of quantum mechanics. For an open quantum system which is in a quantum state defined by a density operator ρ , the probability of a measurement outcome k is given by $\text{tr}(P_k \rho)$. In this case, the quantum state collapses to $\frac{P_k \rho P_k}{\text{tr}(P_k \rho)}$.

In the case of an open quantum system with continuous measurements of an observable X , we can consider a stochastic master equation as follows:

$$\begin{aligned} d\rho(t) = & -i \left[\left(H_0 + \sum_{k=1}^m u_k(t) H_k \right), \rho(t) \right] dt \\ & -\kappa [X, [X, \rho(t)]] dt \\ & + \sqrt{2\kappa} (X\rho(t) + \rho(t)X) \\ & - 2\text{tr}(X\rho(t)) \rho(t) dW \end{aligned} \quad (3)$$

where κ is a constant parameter related to the measurement strength and dW is a standard Wiener increment which is related to the continuous measurement outcome $y(t)$ by

$$dW = dy - 2\sqrt{\kappa} \text{tr}(X\rho(t)) dt; \quad (4)$$

e.g., see Wiseman and Milburn (2010). These models are used in the measurement feedback control of Markovian open quantum systems. Also, equations (3), (4) can be regarded as a quantum filter in which $\rho(t)$ is the conditional density of the quantum system obtained by filtering the measurement signal $y(t)$; e.g., see Bouten et al. (2007) and Gough et al. (2012).

Heisenberg Picture Models of Quantum Systems

In the Heisenberg picture of quantum mechanics, the observables of a system evolve with time, and the quantum state remains fixed. This picture may also be extended slightly by considering the time evolution of general operators on the underlying Hilbert space rather than just observables which are required to be self-adjoint operators. An important class of open quantum systems which are considered in the Heisenberg

picture arises when the underlying Hilbert space is infinite dimensional, and the system represents a collection of independent quantum harmonic oscillators interacting with a number of external quantum fields. Such linear quantum systems are described in the Heisenberg picture by linear quantum stochastic differential equations (QSDEs) of the form

$$\begin{aligned} dx(t) &= Ax(t)dt + BdW(t); \\ dy(t) &= Cx(t)dt + Dw(t) \end{aligned} \quad (5)$$

where A , B , C , and D are real or complex matrices and $x(t)$ is a vector of possibly non-commuting operators on the underlying Hilbert space $\mathcal{H}$; e.g., see James et al. (2008) and Nurdin and Yamamoto (2017). Also, the quantity $dW(t)$ is decomposed as

$$dW(t) = \beta_w(t)dt + d\tilde{w}(t)$$

where $\beta_w(t)$ is an adapted process and $\tilde{w}(t)$ is a quantum Wiener process with Itô table:

$$d\tilde{w}(t)d\tilde{w}(t)^\dagger = F_w dt.$$

Here, $F_w \geq 0$ is a real or complex matrix. The quantity $w(t)$ represents the components of the input quantum fields acting on the system. Also, the quantity $y(t)$ represents the components of interest of the corresponding output fields that result from the interaction of the harmonic oscillators with the incoming fields.

In order to represent physical quantum systems, the components of vector $x(t)$ are required to satisfy certain commutation relations of the form

$$[x_j(t), x_k(t)] = 2i\Theta_{jk}, \quad j, k = 1, 2, \dots, n, \quad \forall t$$

where the matrix $\Theta = (\Theta_{jk})$ is skew symmetric. The requirement to represent a physical quantum system places restrictions on the matrices A , B , C , and D , which are referred to as physical realizability conditions; e.g., see James et al. (2008) and Shaiju and Petersen (2012). QSDE models of the form (5) arise frequently in the area of

quantum optics. They can also be generalized to allow for nonlinear quantum systems such as arise in the areas of nonlinear quantum optics and superconducting quantum circuits; e.g., see Bertet et al. (2012). These models are used in the feedback control of quantum systems in both the case of classical measurement feedback and in the case of coherent feedback in which the quantum controller is also a quantum system and is represented by such a QSDE model. Some of the properties of linear quantum systems can be described using the quantum Kalman decomposition which was derived in Zhang et al. (2018).

(S, L, H) Quantum System Models

An alternative method of modelling an open quantum system as opposed to the stochastic master equation (SME) approach or the quantum stochastic differential equation (QSDE) approach, which were considered above, is to simply model the quantum system in terms of the physical quantities which underlie the SME and QSDE models. For a general open quantum system, these quantities are the *scattering matrix* S which is a matrix of operators on the underlying Hilbert space, the coupling operator L which is a vector of operators on the underlying Hilbert space, and the system Hamiltonian which is a self-adjoint operator on the underlying Hilbert space; e.g., see Gough and James (2009). For a given (S, L, H) model, the corresponding SME model or QSDE model can be calculated using standard formulas; e.g., see Bouten et al. (2007) and James et al. (2008). Also, in certain circumstances, an (S, L, H) model can be calculated from an SME model or a QSDE model. For example, if the linear QSDE model (5) is physically realizable, then a corresponding (S, L, H) model can be found. In fact, this amounts to the definition of physical realizability.

Open Loop Control of Quantum Systems

A fundamental question in the open loop control of quantum systems is the question of controllability. For the case of a closed quantum system of the form (1), the question of controllability

can be defined as follows (e.g., see Albertini and Alessandro 2003):

Definition 1 (Pure State Controllability) The quantum system (1) is said to be *pure state controllable* if for every pair of initial and final states $|\psi_0\rangle$ and $|\psi_f\rangle$ there exist control functions $\{u_k(t)\}$ and a time $T > 0$ such that the corresponding solution of (1) with initial condition $|\psi_0\rangle$ satisfies $|\psi(T)\rangle = |\psi_f\rangle$.

Alternative definitions have also been considered for the controllability of the quantum system (1); e.g., see Albertini and Alessandro (2003) and Grigoriu et al. (2013) in the case of open quantum systems. The following theorem provides a necessary and sufficient condition for pure state controllability in terms of the Lie algebra $\mathcal{L}_0$ generated by the matrices $\{-iH_0, -iH_1, \dots, -iH_m\}$, $\mathfrak{u}(N)$ the Lie algebra corresponding to the unitary group of dimension N , $\mathfrak{su}(N)$ the Lie algebra corresponding to the special unitary group of dimension N , $\mathfrak{sp}(\frac{N}{2})$ the $\frac{N}{2}$ dimensional symplectic group, and $\tilde{\mathcal{L}}$ the Lie algebra conjugate to $\mathfrak{sp}(\frac{N}{2})$.

Theorem 1 (See D'Alessandro 2007) *The quantum system (1) is pure state controllable if and only if the Lie algebra $\mathcal{L}_0$ satisfies one of the following conditions:*

1. $\mathcal{L}_0 = \mathfrak{su}(N)$;
2. $\mathcal{L}_0$ is conjugate to $\mathfrak{sp}(\frac{N}{2})$;
3. $\mathcal{L}_0 = \mathfrak{u}(N)$;
4. $\mathcal{L}_0 = \text{span}\{iI_{N \times N}\} \oplus \tilde{\mathcal{L}}$.

Similar conditions have been obtained when alternative definitions of controllability are used.

Once it has been determined that a quantum system is controllable, the next task in open loop quantum control is to determine the control functions $\{u_k(t)\}$ which drive a given initial state to a given final state. An important approach to this problem is the optimal control approach in which a time optimal control problem is solved using Pontryagin's maximum principle to construct the control functions $\{u_k(t)\}$ which drives the given initial state to the given final state in minimum time; e.g., see Khaneja et al. (2001). This approach works well for low-dimensional quantum systems but is

computationally intractable for high-dimensional quantum systems.

An alternative approach for high-dimensional quantum systems is the Lyapunov control approach. In this approach, a Lyapunov function is selected which provides a measure of the distance between the current quantum state and the desired terminal quantum state. An example of such a Lyapunov function is

$$V = \langle \psi(t) - \psi_f | \psi(t) - \psi_f \rangle \geq 0;$$

e.g., see Mirrahimi et al. (2005). A state feedback control law is then chosen to ensure that the time derivative of this Lyapunov function is negative. This state feedback control law is then simulated with the quantum system dynamics (1) to give the required open loop control functions $\{u_k(t)\}$.

Classical Measurement-Based Quantum Feedback Control

A Schrödinger Picture Approach to Classical Measurement-Based Quantum Feedback Control

In the Schrödinger picture approach to classical measurement-based quantum feedback control with weak continuous measurements, we begin with the stochastic master equations (3), (4) which are considered as both a model for the system being controlled and as a filter which will form part of the final controller. These filter equations are then combined with a control law of the form

$$u(t) = f(\rho(t))$$

where the function $f(\cdot)$ is designed to achieve a particular objective such as stabilization of the quantum system. Here $u(t)$ represents the vector of control inputs $u_k(t)$. An example of such a quantum control scheme is given in the paper Mirrahimi and van Handel (2007) in which a Lyapunov method is used to design the control law $f(\cdot)$ so that a quantum system consisting of an atomic ensemble interacting with an electromagnetic field is stabilized about a specified state $\rho_f = |\psi_m\rangle\langle\psi_m|$. Here, the notation $|\psi_m\rangle\langle\psi_m|$

refers to the outer product between the vector $|\psi_m\rangle$ and itself.

A Heisenberg Picture Approach to Classical Measurement-Based Quantum Feedback Control

In this Heisenberg picture approach to classical measurement-based quantum feedback control, we begin with a quantum system which is described by linear quantum stochastic equations of the form (5). In these equations, it is assumed that the components of the output vector all commute with each other and so can be regarded as classical quantities. This can be achieved if each of the components is obtained via a process of homodyne detection from the corresponding electromagnetic field; e.g., see Bachor and Ralph (2004). Also, it is assumed that the input electromagnetic field $w(t)$ can be decomposed as

$$dw(t) = \begin{bmatrix} \beta_u(t)dt + d\tilde{w}_1(t) \\ dw_2(t) \end{bmatrix} \quad (6)$$

where $\beta_u(t)$ represents the classical control input signal, and $\tilde{w}_1(t)$, $w_2(t)$ are quantum Wiener processes. The control signal displaces components of the incoming electromagnetic field acting on the system via the use of an electrooptic modulator; e.g., see Bachor and Ralph (2004).

The classical measurement feedback-based controllers to be considered are classical systems described by stochastic differential equations of the form

$$\begin{aligned} dx_K(t) &= A_K x_K(t)dt + B_K dy(t); \\ \beta_u(t)dt &= C_K x_K(t)dt. \end{aligned} \quad (7)$$

For a given quantum system model (5), the matrices in the controller (7) can be designed using standard classical control theory techniques such as LQG control (see Doherty and Jacobs 1999) or H^∞ control (see James et al. 2008).

Coherent Quantum Feedback Control

Coherent feedback control of a quantum system corresponds to the case in which the controller

itself is a quantum system which is coupled in a feedback interconnection to the quantum system being controlled; e.g., see Lloyd (2000). This type of control by interconnection is closely related to the behavioral interpretation of feedback control; e.g., see Polderman and Willems (1998).

An important approach to coherent quantum feedback control occurs in the case when the quantum system to be controlled is a linear quantum system described by the QSDEs (5). Also, it is assumed that the input field is decomposed as in (6). However in this case, the quantity $\beta_u(t)$ represents a vector of non-commuting operators on the Hilbert space underlying the controller system. These operators are described by the following linear QSDEs, which represent the quantum controller:

$$\begin{aligned} dx_K(t) &= A_K x_K(t)dt + B_K dy(t) + \bar{B}_K d\bar{w}_K(t); \\ dy_K(t) &= C_K x_K(t)dt + \bar{D}_K d\bar{w}_K(t). \end{aligned} \quad (8)$$

Then, the input $\beta_u(t)$ is identified as

$$\beta_u(t) = C_K x_K(t).$$

Here the quantity

$$dw_K(t) = \begin{bmatrix} dy(t) \\ d\bar{w}_K(t) \end{bmatrix} \quad (9)$$

represents the quantum fields acting on the controller quantum system and where $w_K(t)$ corresponds to a quantum Wiener process with a given Itô table. Also, $y(t)$ represents the output quantum fields from the quantum system being controlled. Note that in the case of coherent quantum feedback control, there is no requirement that the components of $y(t)$ commute with each other, and this in fact represents one of the main advantages of coherent quantum feedback control as opposed to classical measurement-based quantum feedback control.

An important requirement in coherent feedback control is that the QSDEs (8) should satisfy the conditions for physical realizability; e.g., see James et al. (2008). Subject to these constraints,

the controller (8) can then be designed according to an H^∞ or LQG criterion; e.g., see James et al. (2008) and Nurdin et al. (2009). In the case of coherent quantum H^∞ control, it is shown in James et al. (2008) that for any controller matrices (A_K, B_K, C_K) , the matrices $(\bar{B}_K, \bar{D}_K)$ can be chosen so that the controller QSDEs (8) are physically realizable. Furthermore, the choice of the matrices $(\bar{B}_K, \bar{D}_K)$ does not affect the H^∞ performance criterion considered in James et al. (2008). This means that the coherent controller can be designed using the same approach as designing a classical H^∞ controller.

In the case of coherent LQG control such as considered in Nurdin et al. (2009), the choice of the matrices $(\bar{B}_K, \bar{D}_K)$ significantly affects the closed loop LQG performance of the quantum control system. This means that the approach used in solving the coherent quantum H^∞ problem given in James et al. (2008) cannot be applied to the coherent quantum LQG problem. To date there exist only some non-convex optimization methods which have been applied to the coherent quantum LQG problem (e.g., see Nurdin et al. 2009; Khodaparastsichani et al. 2017), and the general solution to the coherent quantum LQG control problem remains an open question.

Cross-References

- [Conditioning of Quantum Open Systems](#)
- [Learning Control of Quantum Systems](#)
- [Pattern Formation in Spin Ensembles](#)
- [Quantum Stochastic Processes and the Modelling of Quantum Noise](#)
- [Robustness Issues in Quantum Control](#)
- [Single-Photon Coherent Feedback Control and Filtering](#)

Bibliography

- Albertini F, D Alessandro D (2003) Notions of controllability for bilinear multilevel quantum systems. *IEEE Trans Autom Control* 48:1399–1403
- Bachor H, Ralph T (2004) A guide to experiments in quantum optics, 2nd edn. Wiley-VCH, Weinheim

- Bertet P, Ong FR, Boissonneault M, Bolduc A, Mallet F, Doherty AC, Blais A, Vion D, Esteve D (2012) Circuit quantum electrodynamics with a nonlinear resonator. In: Dykman M (ed) *Fluctuating nonlinear oscillators: from nanomechanics to quantum superconducting circuits*. Oxford University Press, Oxford
- Bouten L, van Handel R, James M (2007) An introduction to quantum filtering. *SIAM J Control Optim* 46(6):2199–2241
- Breuer H, Petruccione F (2002) *The theory of open quantum systems*. Oxford University Press, Oxford
- Brif C, Chakrabarti R, Rabitz H (2010) Control of quantum phenomena: past, present and future. *New J Phys* 12:075008
- D'Alessandro D (2007) *Introduction to quantum control and dynamics*. Chapman & Hall/CRC, Boca Raton
- Doherty A, Jacobs K (1999) Feedback-control of quantum systems using continuous state-estimation. *Phys Rev A* 60:2700–2711
- Dong D, Petersen IR (2010) Quantum control theory and applications: a survey. *IET Control Theory Appl* 4(12):2651–2671. arXiv:0910.2350
- Dong D, Lam J, Tarn T (2009) Rapid incoherent control of quantum systems based on continuous measurements and reference model. *IET Control Theory Appl* 3: 161–169
- Gough J, James MR (2009) The series product and its application to quantum feedforward and feedback networks. *IEEE Trans Autom Control* 54(11):2530–2544
- Gough JE, James MR, Nurdin HI, Combes J (2012) Quantum filtering for systems driven by fields in single-photon states or superposition of coherent states. *Phys Rev A* 86:043819
- Grigoriu A, Rabitz H, Turinici G (2013) Controllability analysis of quantum systems immersed within an engineered environment. *J Math Chem* 51(6):1548–1560
- Jacobs K (2014) *Quantum measurement theory and its applications*. Cambridge University Press, Cambridge/New York
- James MR, Nurdin HI, Petersen IR (2008) H^∞ control of linear quantum stochastic systems. *IEEE Trans Autom Control* 53(8):1787–1803
- Kerckhoff J, Nurdin HI, Pavlichin DS, Mabuchi H (2010) Designing quantum memories with embedded control: photonic circuits for autonomous quantum error correction. *Phys Rev Lett* 105:040502
- Khaneja N, Brockett R, Glaser S (2001) Time optimal control in spin systems. *Phys Rev A* 63(032308)
- Khodaparastichani A, Vladimirov IG, Petersen IR (2017) A numerical approach to optimal coherent quantum LQG controller design using gradient descent. *Automatica* 85:314–326
- Lloyd S (2000) Coherent quantum feedback. *Phys Rev A* 62(022108)
- Merzbacher E (1970) *Quantum mechanics*, 2nd edn. Wiley, New York
- Mirrahimi M, van Handel R (2007) Stabilizing feedback controls for quantum systems. *SIAM J Control Optim* 46(2):445–467
- Mirrahimi M, Rouchon P, Turinici G (2005) Lyapunov control of bilinear Schrödinger equations. *Automatica* 41:1987–1994
- Nielsen M, Chuang I (2000) *Quantum computation and quantum information*. Cambridge University Press, Cambridge, UK
- Nurdin HI, Yamamoto N (2017) *Linear Dynamical quantum systems: analysis, synthesis, and control*. Springer, Berlin
- Nurdin HI, James MR, Petersen IR (2009) Coherent quantum LQG control. *Automatica* 45(8):1837–1846
- Polderman JW, Willems JC (1998) *Introduction to mathematical systems theory: a behavioral approach*. Springer, New York
- Shaiju AJ, Petersen IR (2012) A frequency domain condition for the physical realizability of linear quantum systems. *IEEE Trans Autom Control* 57(8):2033–2044
- Shor P (1994) Algorithms for quantum computation: discrete logarithms and factoring. In: Goldwasser S (ed) *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science*. IEEE Computer Society, Los Alamitos, pp 124–134
- Wang W, Schirmer SG (2010) Analysis of Lyapunov method for control of quantum states. *IEEE Trans Autom Control* 55(10):2259–2270
- Wiseman HM, Milburn GJ (2010) *Quantum measurement and control*. Cambridge University Press, Cambridge
- Zhang G, Grivopoulos S, Petersen IR, Gough JE (2018) The Kalman decomposition for linear quantum systems. *IEEE Trans Autom Control* 63(2):331–346

Control of Ship Roll Motion

Tristan Perez¹ and Mogens Blanke^{2,3}

¹Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, QLD, Australia

²Department of Electrical Engineering, Automation and Control Group, Technical University of Denmark (DTU), Lyngby, Denmark

³Centre for Autonomous Marine Operations and Systems (AMOS), Norwegian University of Science and Technology, Trondheim, Norway

Abstract

The undesirable effects of roll motion of ships (rocking about the longitudinal axis) became noticeable in the mid-nineteenth century when significant changes were introduced

to the design of ships as a result of sails being replaced by steam engines and the arrangement being changed from broad to narrow hulls. The combination of these changes led to lower transverse stability (lower restoring moment for a given angle of roll) with the consequence of larger roll motion. The increase in roll motion and its effect on cargo and human performance lead to the development several control devices that aimed at reducing and controlling roll motion. The control devices most commonly used today are fin stabilizers, rudder, anti-roll tanks, and gyro stabilizers. The use of different types of actuators for control of ship roll motion has been amply demonstrated for over 100 years. Performance, however, can still fall short of expectations because of difficulties associated with control system design, which have proven to be far from trivial due to fundamental performance limitations and large variations of the spectral characteristics of wave-induced roll motion. This short article provides an overview of the fundamentals of control design for ship roll motion reduction. The overview is limited to the most common control devices. Most of the material is based on Perez (Ship motion control. Advances in industrial control. Springer, London, 2005) and Perez and Blanke (Ann Rev Control 36(1):1367–5788, 2012).

Keywords

Roll damping · Ship motion control

Ship Roll Motion Control Techniques

One of the most commonly used devices to attenuate ship motion are the fin stabilisers. These are small controllable fins located on the bilge of the hull usually amid ships. These devices attain a performance in the range of 60–90 % of roll reduction (root mean square) (Sellars and Martin 1992). They require control systems that sense the vessel's roll motion and act by changing the angle of the fins. These devices are expensive and introduce underwater noise that can affect

sonar performance, they add to propulsion losses, and they can be damaged. Despite this, they are among the most commonly used ship roll motion control device. From a control perspective, highly nonlinear effects (dynamic stall) may appear when operating in severe sea states and heavy rolling conditions (Gaillardie 2002).

During studies of ship damage stability conducted in the late 1800s, it was observed that under certain conditions the water inside the vessel moved out of phase with respect to the wave profile, and thus, the weight of the water on the vessel counteracted the increase of pressure on the hull, hence reducing the net roll excitation moment. This led to the development of fluid anti-roll tank stabilizers. The most common type of anti-roll tank is the U-tank, which comprises two reservoirs, located one on port and one on starboard, connected at the bottom by a duct. Anti-roll tanks can be either passive or active. In passive tanks, the fluid flows freely from side to side. According to the density and viscosity of the fluid used, the tank is dimensioned so that the time required for most of the fluid to flow from side to side equals the natural roll period of the ship. Active tanks operate in a similar manner, but they incorporate a control system that modifies the natural period of the tank to match the actual ship roll period. This is normally achieved by controlling the flow of air from the top of one reservoir to the other. Anti-roll tanks attain a medium to high performance in the range of 20–70 % of roll angle reduction (RMS) (Marzouk and Nayfeh 2009). Anti-roll tanks increase the ship displacement. They can also be used to correct list (steady-state roll angle), and they are the preferred stabilizer for icebreakers.

Rudder-roll stabilization (RRS) is a technique based on the fact that the rudder is located not only aft, but also below the center of gravity of the vessel, and thus the rudder imparts not only yaw but also roll moment. The idea of using the rudder for simultaneous course keeping and roll reduction was conceived in the late 1960s by observations of anomalous behavior of autopilots that did not have appropriate wave filtering – a feature of the autopilot that prevents the rudder from reacting to every single wave; see, for example, Fossen and Perez (2009) for a discussion on

wave filtering. Rudder-roll stabilization has been demonstrated to attain medium to high performance in the range of 50–75 % of roll reduction (RMS) (Baitis et al. 1983; Källström et al. 1988; Blanke et al. 1989; van Amerongen et al. 1990; Oda et al. 1992). The upgrade of the rudder machinery is required to be able to attain slew rates in the range 10–20 deg/s for RRS to have sufficient control authority.

A gyrostabilizer uses the gyroscopic effects of large rotating wheels to generate a roll reducing torque. The use of gyroscopic effects was proposed in the early 1900s as a method to eliminate roll, rather than to reduce it. Although the performance of these systems was remarkable, up to 95 % roll reduction, their high cost, the increase in weight, and the large stress produced on the hull masked their benefits and prevented further developments. However, a recent increase in development of gyrostabilizers has been seen in the yacht industry (Perez and Steinmann 2009).

Fins and rudder give rise to lift forces in proportion to the square of flow velocity past the fin. Hence, roll stabilization by fin or rudder is not possible at low or zero speed. Only U-tanks and gyro devices are able to provide stabilization in these conditions. For further details about the performance of different devices, see Sellars and Martin (1992), and for a comprehensive description of the early development of devices, see Chalmers (1931).

Modeling of Ship Roll Motion for Control Design

The study of roll motion dynamics for control system design is normally done in terms of either one- or four-degrees-of-freedom (DOF) models. The choice between models of different complexity depends on the type of motion control system considered.

For a one-degree-of-freedom (1DOF) case, the following model is used:

$$\dot{\phi} = p, \quad (1)$$

$$I_{xx} \dot{p} = K_h + K_w + K_c, \quad (2)$$

where ϕ is roll angle, p is roll rate, and I_{xx} is rigid-body moment of inertia about the x -axis of a body-fixed coordinate system, where K_h is hydrostatic and hydrodynamic torques, K_w torque generated by wave forces acting on the hull, and K_c the control torques. The hydrodynamic torque can be approximated by the following parametric model: $K_h \approx K_{\dot{p}} \dot{p} + K_p p + K_{p|p|} p|p| + K(\phi)$. The first term represents a hydrodynamic torque in roll due to pressure change that is proportional to the roll accelerations, and the coefficient $K_{\dot{p}}$ is called roll added mass (inertia). The second term is a damping term, which captures forces due to wave making and linear skin friction, and the coefficient K_p is a linear damping coefficient. The third term is a nonlinear damping term, which captures forces due to viscous effects. The last term is the restoring torque due to gravity and buoyancy.

For a 4DOF model (surge, sway, roll, and yaw), motion variables considered are $\eta = [\phi \ \psi]^T$, $v = [u \ v \ p \ r]^T$, $\tau_i = [X \ Y \ K \ N]^T$, where ψ is the yaw angle, the body-fixed velocities are u -surge and v -sway, and r is the yaw rate. The forces and torques are X -surge, Y -sway, K -roll, and N -yaw. With these variables, the following mathematical model is usually considered:

$$\dot{\eta} = \mathbf{J}(\eta) v, \quad (3)$$

$$\mathbf{M}_{RB} \dot{v} + \mathbf{C}_{RB}(v)v = \tau_h + \tau_c + \tau_d, \quad (4)$$

where $\mathbf{J}(\eta)$ is a kinematic transformation, $\mathbf{M}_{RB}$ is the rigid-body inertia matrix that corresponds to expressing the inertia tensor in body-fixed coordinates, $\mathbf{C}_{RB}(v)$ is the rigid-body Coriolis and centripetal matrix, and τ_h , τ_c , and τ_d represent the hydrodynamic, control, and disturbance vector of force components and torques, respectively.

The hydrostatic and hydrodynamic forces are $\tau_h \approx -\mathbf{M}_A \dot{v} - \mathbf{C}_A(v)v - \mathbf{D}(v)v - \mathbf{K}(\phi)$. The first two terms have origin in the motion of a vessel in an irrotational flow in a nonviscous fluid. The third term corresponds to damping forces due to potential (wave making), skin friction, vortex shedding, and circulation (lift and drag). The hydrodynamic effects involved are quite complex, and different approaches based on superposition of either odd-term Taylor expansions or square modulus ($x|x|$) series expan-

sions are usually considered Abkowitz (1964) and Fedyaevsky and Sobolev (1964). The $\mathbf{K}(\phi)$ term represents the restoring forces in roll due to buoyancy and gravity. The 4DOF model captures parameter dependency on ship speed as well as the couplings between steering and roll, and it is useful for controller design. For additional details about mathematical model of marine vehicles, see Fossen (2011).

Wave-Disturbance Models

The action of the waves creates changes in pressure on the hull of the ship, which translate into forces and moments. It is common to model the ship motion response due to waves within a linear framework and to obtain two frequency-response functions (FRF), wave to excitation $F_i(j\omega, U, \chi)$ and wave to motion $H_i(j\omega, U, \chi)$ response functions, where i indicates the degree of freedom. These FRF depend on the wave frequency, the ship speed, and the angle χ at which the waves encounter the ship – this is called the encounter angle.

The wave elevation in deep water is approximately a stochastic process that is zero mean, stationary for short periods of time, and Gaussian (Haverre and Moan 1985). Under these assumptions, the wave elevation ζ is fully described by a power spectral density $\Phi_{\zeta\zeta}(\omega)$. With a linear response assumption, the power spectral density of wave to excitation force and wave to motion can be expressed as

$$\Phi_{FF,i}(j\omega) = |F_i(j\omega, U, \chi)|^2 \Phi_{\zeta\zeta}(j\omega),$$

$$\Phi_{\eta\eta,i}(j\omega) = |H_i(j\omega, U, \chi)|^2 \Phi_{\zeta\zeta}(j\omega).$$

These spectra are models of the wave-induced forces and motions, respectively, from which it is common to generate either time series of wave excitation forces in terms of the encounter frequency to be used as input disturbances in simulation models or time series of wave-induced motion to be used as output disturbance; see, for example, Perez (2005) and references herein.

Roll Motion Control and Performance Limitations

The analysis of performance of ship roll motion control by means of force actuators is usually

conducted within a linear framework by linearizing the models. For a SISO loop where the wave-induced roll motion is considered an output disturbance, the Bode integral constraint applies. This imposes restrictions on one's freedom to shape the closed-loop transfer function to attenuate the motion due to the wave-induced forces in different frequency ranges. These results have important consequences on the design of a roll motion control system since the frequency of the waves seen from the vessel changes significantly with the sea state, the speed of the vessel, and the wave encounter angle. The changing characteristics on open-loop roll motion in conjunction with the Bode integral constraint make the control design challenging since roll amplification may occur if the control design is not done properly. For some roll motion control problems, like using the rudder for simultaneous roll attenuation and heading control, the system presents non-minimum phase dynamics. In this case, the trade-off of reduced sensitivity vs. amplification of roll motion is dominating at frequencies close to the non-minimum phase zero – a constraint with origin in the Poisson integral (Hearns and Blanke 1998); see also Perez (2005).

It should be noted that non-minimum phase dynamics also occurs with fin stabilizers, when the stabilizers are located aft of the center of gravity. With the fins at this location, they behave like a rudder and introduce non-minimum phase dynamics and heading interference at low wave-excitation frequencies. These aspects of fin location were discussed by Lloyd (1989).

The above discussion highlights general design constraints that apply to roll motion control systems in terms of the dynamics of the vessel and actuator. In addition to these constraints, one needs also to account for limitations in actuator slew rate and angle.

Controls Techniques Used in Different Roll Control Systems

Fin Stabilizers

In regard to fin stabilizers, the control design is commonly addressed using the 1DOF model (1) and (2). The main issues associated with control

design are the parametric uncertainty in model and the Bode integral constraint. This integral constraint can lead to roll amplification due to changes in the spectrum of the wave-induced roll moment with sea state and sailing conditions (speed and encounter angle). Fin machinery is designed so that the rate of the fin motion is fast enough, and actuator rate saturation is not an issue in moderate sea states. The fins could be used to correct heeling angles (steady-state roll) when the ship makes speed, but this is avoided due to added resistance. If it is used, integral action needs to include anti-windup. In terms of control strategies, PID, $\mathcal{H}_\infty$, and LQR techniques have been successfully applied in practice. Highly nonlinear effects (dynamic stall) may appear when operating in severe sea states and heavy rolling conditions, and proposals for applications of model predictive control have been put forward to constraint the effective angle of attack of the fins. In addition, if the fins are located too far aft along the ship, the dynamic response from fin angle to roll can exhibit non-minimum phase dynamics, which can limit the performance at low encounter frequencies. A thorough review of the control literature can be found in Perez and Blanke (2012).

Rudder-Roll Stabilization

The problem of rudder-roll stabilization requires the 4DOF model (3) and (4), which captures the interaction between roll, sway, and yaw together with the changes in the hydrodynamic forces due to the forward speed. The response from rudder to roll is non-minimum phase (NMP), and the system is characterized by further constraints due to the single-input-two-output nature of the control problem – attenuate roll without too much interference with the heading. Studies of fundamental limitations due to NMP dynamics have been approached using standard frequency-domain tools by Hearn and Blanke (1998) and Perez (2005). A characterization of the trade-off between roll reduction vs. increase of interference was part of the controller design in Stoustrup et al. (1994). Perez (2005) determined the limits obtainable using optimal control with full disturbance information. The latter also incorporated constraints due to the limiting authority of the

control action in rate and magnitude of rudder machinery and stall conditions of the rudder. The control design for rudder-roll stabilization has been addressed in practice using PID, LQG, and $\mathcal{H}_\infty$ and standard frequency-domain linear control designs. The characteristics of limited control authority were solved by van Amerongen et al. (1990) using automatic gain control. In the literature, there have been proposals put forward for the use of model predictive control, QFT, sliding-mode nonlinear control, and autoregressive stochastic control. Combined use of fin and rudder has also been investigated. Grimbé et al. (1993) and later Roberts et al. (1997) used $\mathcal{H}_\infty$ control techniques. Thorough comparison of controller performances for warships was published in Crossland (2003). A thorough review of the control literature can be found in Perez and Blanke (2012).

Gyrostabilizers

Using a single gimbal suspension gyrostabilizer for roll damping control, the coupled vessel-roll-gyro model can be modeled as follows:

$$\dot{\phi} = p, \quad (5)$$

$$K_{\dot{p}} \dot{p} + K_p p + K_{\phi} \phi = K_w - K_g \dot{\alpha} \cos \alpha \quad (6)$$

$$I_p \ddot{\alpha} + B_p \dot{\alpha} + C_p \sin \alpha = K_g p \cos \alpha + T_p, \quad (7)$$

where (6) represents the 1DOF roll dynamics and (7) represents the dynamics of the gyrostabilizer about the axis of the gimbal suspension, where α is the gimbal angle, equivalent to the precession angle for a single gimbal suspension, I_p is gimbal and wheel inertia about the gimbal axis, B_p is the damping, and C_p is a restoring term of the gyro about the precession axis due to location of the gyro center of mass relative to the precession axis (Arnold and Maunder 1961). T_p is the control torque applied to the gimbal. The use of twin counter-spinning wheels prevents gyroscopic coupling with other degrees of freedom. Hence, the control design for gyrostabilizers can be based on a linear single-degree-of-freedom model for roll.

The wave-induced roll moment K_w excites the roll motion. As the roll motion develops, the roll rate p induces a torque along the precession axis

of the gyrostabilizer. As the precession angle α develops, there is reaction torque done on the vessel that opposes the wave-induced moment. The latter is the roll stabilizing torque, $X_g \triangleq -K_g \dot{\alpha} \cos \alpha \approx -K_g \dot{\alpha}$. This roll torque can only be controlled indirectly through the precession dynamics in (7) via T_p . In the model above, the spin angular velocity ω_{spin} is controlled to be constant; hence the wheels' angular momentum $K_g = I_{spin} \omega_{spin}$ is constant.

The precession control torque T_p is used to control the gyro. As observed by Sperry (Chalmers 1931), the intrinsic behavior of the gyrostabilizer is to use roll rate to generate a roll torque. Hence, one could design a precession torque controller such that from the point of view of the vessel, the gyro behaves as damper. Depending on how precession torque is delivered, it may be necessary to constraint precession angle and rate. This problem has been recently considered in Donaire and Perez (2013) using passivity-based control.

U-tanks

U-tanks can be passive or active. Roll reduction is achieved by attempting to transfer energy from the roll motion to motion of liquid within the tank and using the weight of the liquid to counteract the wave excitation moment. A key aspect of the design is the dimension and geometry of the tank to ensure that there is enough weight due to the displaced liquid in the tank and that the oscillation of the fluid in the tank matches the vessel natural frequency in roll; see Holden and Fossen (2012) and references herein. The design of the U-tank can ensure a single-frequency matching, at which the performance is optimized, and for this frequency the roll natural frequency is used. As the frequency of roll motion departs from this, a degradation of roll reduction occurs. Active U-tanks use valves to control the flow of air from the top of the reservoirs to extend the frequency matching in sailing conditions in which the roll dominant frequency is lower than the roll natural frequency – the flow of air is used to delay the motion of the liquid from one reservoir to the other. This control is achieved by detecting the dominant roll frequency and using this infor-

mation to control the air flow from one reservoir to the other. If the roll dominant frequency is higher than the roll natural frequency, the U-tank is used in passive mode, and the standard roll reduction degradation occurs.

Summary and Future Directions

This article provides a brief summary of control aspects for the most common ship roll motion control devices. These aspects include the type of mathematical models used to design and analyze the control problem, the inherent fundamental limitations and the constraints that some of the designs are subjected to, and the performance that can be expected from the different devices. As an outlook, one of the key issues in roll motion control is the model uncertainty and the adaptation to the changes in the environmental conditions. As the vessel changes speed and heading, or as the seas build up or abate, the dominant frequency range of the wave-induced forces changes significantly. Due to the fundamental limitations discussed, a nonadaptive controller may produce roll amplification rather than roll reduction. This topic has received some attention in the literature via multi-mode control switching, but further work in this area could be beneficial. In the recent years, new devices have appeared for stabilization at zero speed, like flapping fins and rotating cylinders. Also the industry's interest in roll gyro-stabilizers has been re-ignited. The investigation of control designs for these devices has not yet received much attention within the control community. Hence, it is expected that this will create a potential for research activity in the future.

Cross-References

- [Fundamental Limitation of Feedback Control](#)
- [H-Infinity Control](#)
- [H₂ Optimal Control](#)
- [Linear Quadratic Optimal Control](#)
- [Mathematical Models of Ships and Underwater Vehicles](#)

Bibliography

- Abkowitz M (1964) Lecture notes on ship hydrodynamics—steering and manoeuvrability. Technical report Hy-5, Hydro and Aerodynamics Laboratory, Lyngby
- Arnold R, Maunder L (1961) Gyrodynamics and its engineering applications. Academic, New York/London
- Baitis E, Woollaver D, Beck T (1983) Rudder roll stabilization of coast guard cutters and frigates. *Nav Eng J* 95(3):267–282
- Blanke M, Haals P, Andreassen KK (1989) Rudder roll damping experience in Denmark. In: Proceedings of IFAC workshop CAMS'89, Lyngby
- Chalmers T (1931) The automatic stabilisation of ships. Chapman and Hall, London
- Crossland P (2003) The effect of roll stabilization controllers on warship operational performance. *Control Eng Pract* 11:423–431
- Donaire A, Perez T (2013) Energy-based nonlinear control of ship roll gyro-stabiliser with precession angle constraints. In: 9th IFAC conference on control applications in marine systems, Osaka
- Fedyayevsky K, Sobolev G (1964) Control and stability in ship design. State Union Shipbuilding, Leningrad
- Fossen TI (2011) Handbook of marine craft hydrodynamics and motion control. Wiley, Chichester
- Fossen T, Perez T (2009) Kalman filtering for positioning and heading control of ships and offshore rigs. *IEEE Control Syst Mag* 29(6):32–46
- Gaillard G (2002) Dynamic behavior and operation limits of stabilizer fins. In: IMAM international maritime association of the Mediterranean, Creta
- Grimble M, Katebi M, Zang Y (1993) $\mathcal{H}_\infty$ -based ship fin-rudder roll stabilisation. In: 10th ship control system symposium SCSS, Ottawa, vol 5, pp 251–265
- Haverre S, Moan T (1985) On some uncertainties related to short term stochastic modelling of ocean waves. In: Probabilistic offshore mechanics. Progress in engineering science. CML
- Hearn G, Blanke M (1998) Quantitative analysis and design of rudder roll damping controllers. In: Proceedings of CAMS'98, Fukuoka, pp 115–120
- Holden C, Fossen TI (2012) A nonlinear 7-DOF model for U-tanks of arbitrary shape. *Ocean Eng* 45: 22–37
- Källström C, Wessel P, Sjölander S (1988) Roll reduction by rudder control. In: Spring meeting-STAR symposium, 3rd IMSDC, Pittsburgh
- Lloyd A (1989) Seakeeping: ship behaviour in rough weather. Ellis Horwood
- Marzouk OA, Nayfeh AH (2009) Control of ship roll using passive and active anti-roll tanks. *Ocean Eng* 36:661–671
- Oda H, Sasaki M, Seki Y, Hotta T (1992) Rudder roll stabilisation control system through multivariable autoregressive model. In: Proceedings of IFAC conference on control applications of marine systems—CAMS
- Perez T (2005) Ship motion control. Advances in industrial control. Springer, London
- Perez T, Blanke M (2012) Ship roll damping control. *Ann Rev Control* 36(1):1367–5788
- Perez T, Steinmann P (2009) Analysis of ship roll gyro-stabiliser control. In: 8th IFAC international conference on manoeuvring and control of marine craft, Guaruja
- Roberts G, Sharif M, Sutton R, Agarwal A (1997) Robust control methodology applied to the design of a combined steering/stabiliser system for warships. *IEE Proc Control Theory Appl* 144(2):128–136
- Sellars F, Martin J (1992) Selection and evaluation of ship roll stabilization systems. *Mar Technol SNAME* 29(2):84–101
- Stoustrup J, Niemann HH, Blanke M (1994) Rudder-roll damping for ships- a new $\mathcal{H}_\infty$ approach. In: Proceedings of 3rd IEEE conference on control applications, Glasgow, pp 839–844
- van Amerongen J, van der Klugt P, van Nauta Lemke H (1990) Rudder roll stabilization for ships. *Automatica* 26:679–690

Control Structure Selection

Sigurd Skogestad

Department of Chemical Engineering,
Norwegian University of Science and
Technology (NTNU), Trondheim, Norway

Abstract

Control structure selection deals with selecting what to control (outputs), what to measure and what to manipulate (inputs), and also how to split the controller in a hierarchical and decentralized manner. The most important issue is probably the selection of the controlled variables (outputs), $CV = Hy$, where y are the available measurements and H is a degree of freedom that is seldom treated in a systematic manner by control engineers. This entry discusses how to find H for both for the upper (slower) economic layer and the lower (faster) regulatory layer in the control hierarchy. Each layer may be split in a decentralized fashion. Systematic approaches for input/output (IO) selection are presented.

Keywords

Control configuration · Control hierarchy · Control structure design · Decentralized control · Economic control · Input-output controllability · Input/output selection · Plantwide control · Regulatory control · Supervisory control

Introduction

Consider the generalized controller design problem in Fig. 1 where P denotes the generalized plant model. Here, the objective is to design the controller K , which, based on the sensed outputs v , computes the inputs (MVs) u such that the variables z are kept small, in spite of variations in the variables w , which include disturbances (d), varying setpoints/references (CV_s) and measurement noise (n),

$$w = [d, CV_s, n]$$

The variables z , which should be kept small, typically include the control error for the selected controlled variables (CV) plus the plant inputs (u),

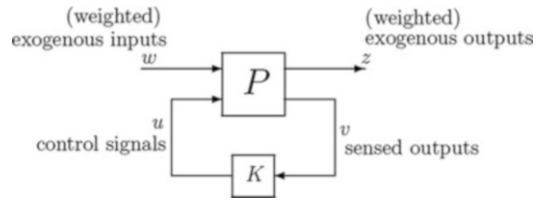
$$z = [CV - CV_s; u]$$

The variables v , which are the inputs to the controller, include all known variables, including measured outputs (y_m), measured disturbances (d_m) and setpoints,

$$v = [y_m; d_m; CV_s].$$

The cost function for designing the optimal controller K is usually the weighted control error, $J' = ||W'z||$. The reason for using a prime on J (J'), is to distinguish it from the economic cost J which we later use for selecting the controlled variables (CV).

Notice that it is assumed in Fig. 1 that we know what to measure (v), manipulate (u), and, most importantly, which variables in z we would like to keep at setpoints (CV), that is, we have assumed a given control structure. The term “control structure selection” (CSS) and its synonym “control



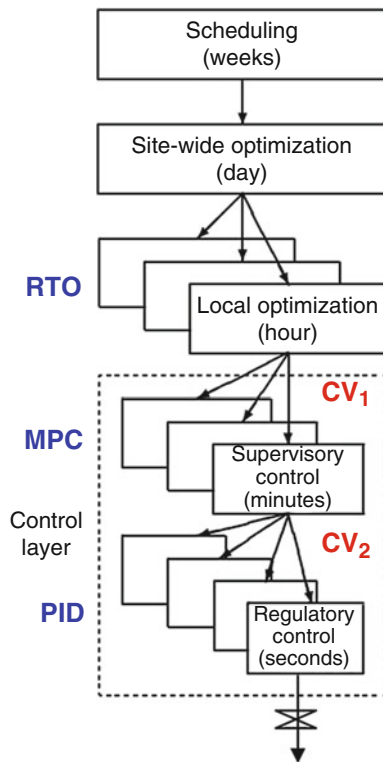
Control Structure Selection, Fig. 1 General formulation for designing the controller K . The plant P is controlled by manipulating u , and is disturbed by the signals w . The controller uses the measurements v , and the control objective is to keep the outputs (weighted control error) z as small as possible

structure design” (CSD) is associated with the overall *control philosophy* for the system with emphasis on the *structural decisions* which are a prerequisite for the controller design problem in Fig. 1:

1. *Selection of controlled variables* (CVs, “outputs,” included in z in Fig. 1)
2. *Selection of manipulated variables* (MVs, “inputs,” u in Fig. 1)
3. *Selection of measurements* y (included in v in Fig. 1)
4. *Selection of control configuration* (structure of overall controller K that interconnects the controlled, manipulated and measured variables; structure of K in Fig. 1)
5. *Selection of type of controller* K (PID, MPC, LQG, H-infinity, etc.) and objective function (norm) used to design and analyze it.

Decisions 2 and 3 (selection of u and y) are sometimes referred to as the input/output (IO) selection problem. In practice, the controller (K) is usually divided into several layers, operating on different time scales (see Fig. 2), which implies that we in addition to selecting the (primary) controlled variables ($CV_1 \equiv CV$) must also select the (secondary) variables that interconnect the layers (CV_2).

Control structure selection includes all the *structural* decisions that the engineer needs to make when designing a control system, but it does not involve the actual design of each individual controller block. Thus, it involves the deci-



Control Structure Selection, Fig. 2 Typical control hierarchy, as illustrated for a process plant

sions necessary to make a block diagram (Fig. 1; used by control engineers) or process & instrumentation diagram (used by process engineers) for the entire plant, and provides the starting point for a detailed controller design.

The term “plantwide control,” which is a synonym for “control structure selection,” is used in the field of process control. Control structure selection is particularly important for process control because of the complexity of large processing plants, but it applies to all control applications, including vehicle control, aircraft control, robotics, power systems, biological systems, social systems, and so on.

It may be argued that control structure selection is more important than the controller design itself. Yet, control structure selection is hardly covered in most control courses. This is probably related to the complexity of the problem, which requires the knowledge from several engineering fields. In the mathematical sense, the control

structure selection problem is a formidable combinatorial problem which involves a large number of discrete decision variables.

Overall Objectives for Control and Structure of the Control Layer

The starting point for control system design is to define clearly the operational objectives. There are usually two main objectives for control:

1. Longer-term economic operation (minimize economic cost J subject to satisfying operational constraints)
2. Stability and short-term regulatory control

The first objective is related to “making the system operate as intended,” where economics are an important issue. Traditionally, control engineers have not been much involved in this step. The second objective is related to “making sure the system stays operational,” where stability and robustness are important issues, and this has traditionally been the main domain of control engineers. In terms of designing the control system, the second objective (stabilization) is usually considered first. An example is bicycle riding; we first need to learn how to stabilize the bicycle (regulation), before trying to use it for something useful (optimal operation), like riding to work and selecting the shortest path.

We use the term “economic cost,” because usually the cost function J can be given a monetary value, but more generally, the cost J could be any scalar cost. For example, the cost J could be the “environmental impact” and the economics could then be given as constraints.

In theory, the optimal strategy is to combine the control tasks of optimal economic operation and stabilization/regulation in a single centralized controller K , which at each time step collects all the information and computes the optimal input changes. In practice, simpler controllers are used. The main reason for this is that in most cases one can obtain acceptable control performance with simple structures, where each controller block involves only a few variables. Such control sys-

tems can be designed and tuned with much less effort, especially when it comes to the modeling and tuning effort.

So how are large-scale systems controlled in practise? Usually, the controller K is decomposed into several subcontrollers, using two main principles

- *Decentralized (local) control*. This “horizontal decomposition” of the control layer is usually based on separation in space, for example, by using local control of individual units.
- *Hierarchical (cascade) control*. This “vertical decomposition” is usually based on time scale separation, as illustrated for a process plant in Fig. 2. The upper three layers in Fig. 2 deal explicitly with economic optimization and are not considered here. We are concerned with the two lower *control layers*, where the main objective is to track the setpoints specified by the layer above.

In accordance with the two main objectives for control, the control layer is in most cases divided hierarchically in two layers (Fig. 2):

1. A “slow” supervisory (economic) layer
2. A “fast” regulatory (stabilization) layer

Another reason for the separation in two control layers, is that the tasks of economic operation and regulation are fundamentally different. Combining the two objectives in a single cost function, which is required for

designing a single centralized controller K , is like trying to compare apples and oranges. For example, how much is an increased stability margin worth in monetary units [\$]? Only if there is a reasonable benefit in combining the two layers, for example, because there is limited time scale separation between the tasks of regulation and optimal economics, should one consider combining them into a single controller.

Notation and Matrices H_1 and H_2 for Controlled Variable Selection

The most important notation is summarized in Table 1 and Fig. 3. To distinguish between the two control layers, we use “1” for the upper supervisory (economic) layer and “2” for the regulatory layer, which is “secondary” in terms of its place in the control hierarchy.

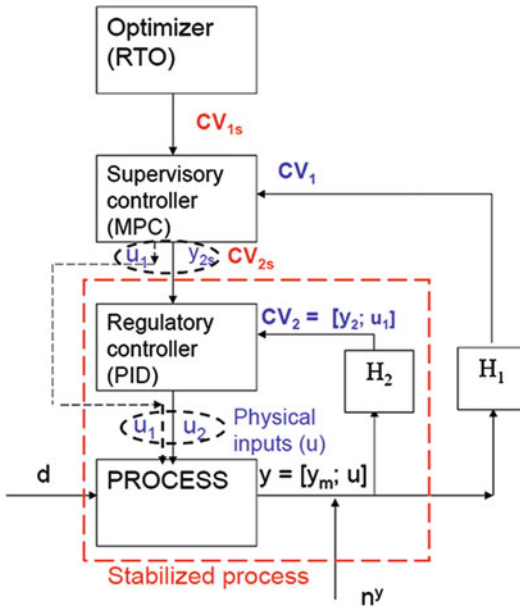
There is often limited possibility to select the input set (u) as it is usually constrained by the plant design. However, there may be a possibility to add inputs or to move some to another location, for example, to avoid saturation or to reduce the time delay and thus improve the input-output controllability.

There is much more flexibility in terms of output selection, and the most important structural decision is related to the selection of controlled variables in the two control layers, as given by the decision matrices H_1 and H_2 (see Fig. 3).

$$CV_1 = H_1 y$$

Control Structure Selection, Table 1 Important notation

$u = [u_1; u_2]$ = set of all available physical plant inputs
u_1 = inputs used directly by supervisory control layer
u_2 = inputs used by regulatory layer
y_m = set of all measured outputs
$y = [y_m; u]$ = combined set of measurements and inputs
y_2 = controlled outputs in regulatory layer (subset or combination of y); $\dim(y_2) = \dim(u_2)$
$CV_1 = H_1 y$ = controlled variables in supervisory layer; $\dim(CV_1) = \dim(u)$
$CV_2 = [y_2; u_1] = H_2 y$ = controlled variables in regulatory layer; $\dim(CV_2) = \dim(u)$
$MV_1 = CV_{2s} = [y_{2s}; u_1]$ = manipulated variables in supervisory layer; $\dim(MV_1) = \dim(u)$
$MV_2 = u_2$ = manipulated variables in regulatory layer; $\dim(MV_2) = \dim(u_2) \leq \dim(u)$



Control Structure Selection, Fig. 3 Block diagram of a typical control hierarchy, emphasizing the selection of controlled variables for supervisory (economic) control ($CV_1 = H_1 y$) and regulatory control ($CV_2 = H_2 y$)

$$CV_2 = H_2 y$$

Note from the definition in Table 1 that $y = [y_m; u]$. Thus, y includes, in addition to the candidate measured outputs (y_m), also the physical inputs u . This allows for the possibility of selecting an input u as a “controlled” variable, which means that this input is kept constant (or, more precisely, the input is left “unused” for control in this layer).

In general, H_1 and H_2 are “full” matrices, allowing for measurement combinations as controlled variables. However, for simplicity, especially in the regulatory layer, we often prefer to control individual measurements, that is, H_2 is usually a “selection matrix,” where each row in H_2 contains one 1-element (to identify the selected variable) with the remaining elements set to 0. In this case, we can write $CV_2 = H_2 y = [y_2; u_1]$, where y_2 denotes the actual controlled variables in the regulatory layer, whereas u_1 denotes the “unused” inputs (u_1), which are left as degrees of freedom for the supervisory layer. Note that this indirectly determines the

inputs u_2 used in the regulatory layer to control y_2 , because u_2 is what remains in the set u after selecting u_1 . To have a simple control structure, with as few regulatory loops as possible, it is desirable that H_2 is selected such that there are many inputs (u_1) left “unused” in the regulatory layer.

Example. Assume there are three candidate output measurements (temperatures T) and two inputs (flowrates q),

$$y_m = [T_a T_b T_c], \quad u = [q_a q_b]$$

and we have by definition $y = [y_m; u]$. Then the choice

$$H_2 = [0 \ 1 \ 0 \ 0 \ 0; \ 0 \ 0 \ 0 \ 0 \ 1]$$

means that we have selected $CV_2 = H_2 y = [T_b; q_b]$. Thus, $u_1 = q_b$ is an unused input for regulatory control, and in the regulatory layer we close one loop, using $u_2 = q_a$ to control $y_2 = T_b$. If we instead select

$$H_2 = [1 \ 0 \ 0 \ 0 \ 0; \ 0 \ 0 \ 1 \ 0 \ 0]$$

then we have $CV_2 = [T_a; T_c]$. None of these are inputs, so u_1 is an empty set in this case. This means that we need to close two regulatory loops, using $u_2 = [q_a; q_b]$ to control $y_2 = [T_a; T_c]$.

Supervisory Control Layer and Selection of Economic Controlled Variables (CV_1)

Some objectives for the supervisory control layer are given in Table 2. The main structural issue for the supervisory control layer, and probably the most important decision in the design of any control system, is the selection of the primary (economic) controlled variable CV_1 . In many cases, a good engineer can make a reasonable choice based on process insight and experience. However, the control engineer must realize that this is a critical decision. The main rules and issues for selecting CV_1 are

Control Structure Selection, Table 2 Objectives of supervisory control layer

O1. Control primary “economic” variables CV_1 at setpoint using as degrees of freedom MV_1 , which includes the setpoints to the regulatory layer ($y_{2s} = CV_{2s}$) as well as any “unused” degrees of freedom (u_1)
O2. Switch controlled variables (CV_1) depending on operating region, for example, because of change in active constraints
O3. Supervise the regulatory layer, for example, to avoid input saturation (u_2), which may destabilize the system
O4. Coordinate control loops (multivariable control) and reduce effect of interactions (decoupling)
O5. Provide feedforward action from measured disturbances
O6. Make use of additional inputs, for example, to improve the dynamic performance (usually combined with input readjusting control) or to extend the steady-state operating range (split range control)
O7. Make use of extra measurements, for example, to estimate the primary variables CV_1

 CV_1 Rule 1. Control active constraints (almost always)

- Active constraints may often be identified by engineering insight, but more generally requires optimization based on a detailed model.

For example, consider the problem of minimizing the driving time between two cities (cost $J = T$). There is a single input ($u = \text{fuel flow } f[\text{l/s}]$) and the optimal solution is often constrained. When driving a fast car, the active constraint may be the speed limit ($CV_1 = v[\text{km/h}]$ with setpoint $v_{\max}$, e.g., $v_{\max} = 100 \text{ km/h}$). When driving an old car, the active constraint may be the maximum fuel flow ($CV_1 = f[\text{l/s}]$ with setpoint $f_{\max}$). The latter corresponds to an input constraint ($u_{\max} = f_{\max}$) which is trivial to implement (“full gas”); the former corresponds to an output constraint ($y_{\max} = v_{\max}$) which requires a controller (“cruise control”).

- For “hard” output constraints, which cannot be violated at any time, we need to introduce a *backoff* (safety margin) to guarantee feasibility. The backoff is defined as the difference between the optimal value and the actual setpoint, for example, we need to back off from the speed limit because of the possibility for measurement error and imperfect control

$$CV_{1,s} = CV_{1,\max} - \text{backoff}$$

For example, to avoid exceeding the speed limit of 100 km/h, we may set $\text{backoff} = 5 \text{ km/h}$, and use a setpoint $v_s = 95 \text{ km/h}$ rather than 100 km/h.

 CV_1 Rule 2. For the remaining unconstrained degrees of freedom, look for “self-optimizing” variables which when held constant, indirectly lead to close-to-optimal operation, in spite of disturbances.

- Self-optimizing variables ($CV_1 = H_1 y$) are variables which when kept constant, indirectly (through the action of the feedback control system) lead to close-to optimal adjustment of the inputs (u) when there are disturbances (d).
- An ideal self-optimizing variable is the gradient of the cost function with respect to the unconstrained input. $CV_1 = dJ/du = J_u$
- More generally, since we rarely can measure the gradient J_u , we select $CV_1 = H_1 y$. The selection of a good H_1 is a nontrivial task, but some quantitative approaches are given below.

For example, consider again the problem of driving between two cities, but assume that the objective is to minimize the total fuel, $J = V[\text{liters}]$. Here, driving at maximum speed will consume too much fuel, and driving too slow is also nonoptimal. This is an unconstrained optimization problem, and identifying a good CV_1 is not obvious. One option is to maintain

a constant speed ($CV_1 = v$), but the optimal value of v may vary depending on the slope of the road. A more “self-optimizing” option, could be to keep a constant fuel rate ($CV_1 = f[l/s]$), which will imply that we drive slower uphill and faster downhill. More generally, one can control combinations, $CV_1 = H_1 y$ where H_1 is a “full” matrix.

CV₁ Rule 3. For the unconstrained degrees of freedom, one should *never* control a variable that reaches its maximum or minimum value at the optimum, for example, never try to control directly the cost J . Violation of this rule gives either infeasibility (if attempting to control J at a lower value than $J_{\min}$) or nonuniqueness (if attempting to control J at higher value than $J_{\min}$).

Assume again that we want to minimize the total fuel needed to drive between two cities, $J = V[l]$. Then one should avoid fixing the total fuel, $CV_1 = V[l]$, or, alternatively, avoid fixing the fuel consumption (“gas mileage”) in liters pr. km ($CV_1 = f[l/km]$). Attempting to control the fuel consumption $[l/km]$ below the car’s minimum value is obviously not possible (infeasible). Alternatively, attempting to control the fuel consumption above its minimum value has two possible solutions; driving slower or faster than the optimum. Note that the policy of controlling the fuel rate $f[l/s]$ at a fixed value will never become infeasible.

For CV₁-Rule 2, it is always possible to find good variable combinations (i.e., H_1 is a “full” matrix), at least locally, but whether or not it is possible to find good individual variables (H_1 is a selection matrix), is not obvious. To help identify potential “self-optimizing” variables ($CV_1 = c$), the following requirements may be used:

Requirement 1. The optimal value of c is insensitive to disturbances, that is, $dc_{\text{opt}}/dd = H_1 F$ is small. Here $F = dy_{\text{opt}}/dd$ is the optimal sensitivity matrix (see below).

Requirement 2. The variable c is easy to measure and control accurately

Requirement 3. The value of c is sensitive to changes in the manipulated variable, u ; that is,

the gain, $G = HG^y$, from u to c is large (so that even a large error in controlled variable, c , results in only a small variation in u .) Equivalently, the optimum should be “flat” with respect to the variable, c . Here $G^y = dy/du$ is the measurement gain matrix (see below).

Requirement 4. For cases with two or more controlled variables c , the selected variables should not be closely correlated.

All four requirements should be satisfied. For example, for the operation of a marathon runner, the heart rate may be a good “self-optimizing” controlled variable c (to keep at constant set-point). Let us check this against the four requirements. The optimal heart rate is weakly dependent on the disturbances (requirement 1) and the heart rate is easy to measure (requirement 2). The heart rate is quite sensitive to changes in power input (requirement 3). Requirement 4 does not apply since this is a problem with only one unconstrained input (the power). In summary, the heart rate is a good candidate.

Regions and switching. If the optimal active constraints vary depending on the disturbances, new controlled variables (CV_1) must be identified (offline) for each active constraint region, and on-line switching is required to maintain optimality. In practise, it is easy to identify when to switch when one reaches a constraint. It is less obvious when to switch out of a constraint, but actually one simply has to monitor the value of the unconstrained CVs from the neighbouring regions and switch out of the constraint region when the unconstrained CV reaches its setpoint.

In general, one would like to simplify the control structure and reduce need for switching. This may require using a suboptimal CV_1 in some regions of active constraints. In this case, the setpoint for CV_1 may not be its nominally optimal value (which is the normal choice), but rather a “robust setpoint” (with backoff) which reduces the loss when we are outside the nominal constraint region.

Structure of supervisory layer. The supervisory layer may either be centralized, e.g., using

model predictive control (MPC), or decomposed into simpler subcontrollers using standard elements, like decentralized control (PID), cascade control, selectors, decouplers, feedforward elements, ratio control, split range control, and input midrange control (also known as input resetting, valve position control or habituating control). In theory, the performance is better with the centralized approach (e.g., MPC), but the difference can be small when designed by a good engineer. The main reasons for using simpler elements is that (1) the system can be implemented in the existing “basic” control system, (2) it can be implemented with little model information, and (3) it can be build up gradually. However, such systems can quickly become complicated and difficult to understand for other than the engineer who designed it. Therefore, model-based centralized solutions (MPC) are often preferred because the design is more systematic and easier to modify.

Quantitative Approach for Selecting Economic Controlled Variables, CV_1

A quantitative approach for selecting economic controlled variables is to consider the effect of the choice $CV_1 = H_1 y$ on the economic cost J when disturbances d occur. One should also include noise/errors (n^y) related to the measurements and inputs.

Step S1. *Define operational objectives (economic cost function J and constraints)*

We first quantify the operational objectives in terms of a scalar cost function J [\$/s] that should be minimized (or equivalently, a scalar profit function, $P = -J$, that should be maximized). For process control applications, this is usually easy, and typically we have

$$J = \text{cost feed} + \text{cost utilities (energy)} \\ - \text{value products [$/s]}$$

Note that the economic cost function J is used to *select* the controlled variables (CV_1), and another cost function (J'), typically involving the deviation in CV_1 from their optimal

setpoints CV_{1s} , is used for the actual controller design (e.g., using MPC).

Step S2. *Find optimal operation for expected disturbances*

Mathematically, the optimization problem can be formulated as

$$\text{minimize } J(u, x, d)$$

subject to:

$$\text{Model equations: } dx/dt = f(u, x, d)$$

$$\text{Operational constraints: } g(u, x, d) \leq 0$$

In many cases, the economics are determined by the steady-state behavior, so we can set $dx/dt = 0$. The optimization problem should be resolved for the expected disturbances (d) to find the truly optimal operation policy, $u_{\text{opt}}(d)$. The nominal solution (d_{nom}) may be used to obtain the setpoints (CV_{1s}) for the selected controlled variables. In practise, the optimum input $u_{\text{opt}}(d)$ cannot be realized, because of model error and unknown disturbances d , so we use a feedback implementation where u is adjusted to keep the selected variables CV_1 at their nominally optimal setpoints.

Together with obtaining the model, the optimization step S2 is often the most time consuming step in the entire plantwide control procedure.

Step S3. *Select supervisory (economic) controlled variables, CV_1*

CV_1 -Rule 1: Control Active Constraints

A primary goal for solving the optimization problem is to find the expected regions of active constraints, and a constraint is said to be “active” if $g = 0$ at the optimum. The optimally active constraints will vary depending on disturbances (d) and market conditions (prices).

CV_1 -Rule 2: Control Self-Optimizing Variables

After having identified (and controlled) the active constraints, one should consider the remaining lower-dimension unconstrained optimization

problem, and for the remaining unconstrained degrees of freedom one should search for *control “self-optimizing” variables* c .

1. **“Brute force” approach.** Given a set of controlled variables $CV_1 = c = H_1 y$, one computes the cost $J(c, d)$ when we keep c constant ($c = c_s + H_1 n^y$) for various disturbances (d) and measurement errors (n^y). In practise, this is done by running a large number of steady-state simulations to try to cover the expected future operation.
2. **“Local” approaches** based on a quadratic approximation of the cost J . Linear models are used for the effect of u and d on y .

$$y = G^y u + G_d^y d$$

This is discussed in more detail in Alstad et al. (2009) and references therein. The main local approaches are:

- 2A. **Maximum gain rule: maximize the minimum singular value of $G = H_1 G^y$.** In other words, the maximum gain rule, which essentially is a quantitative version of Requirements 1, 3 and 4 given above, says that one should control “sensitive” variables, with a large scaled gain G from the inputs (u) to $c = H_1 y$. This rule is good for pre-screening and also yields good insight.
- 2B. **Nullspace method.** This method yields optimal measurement combinations for the case with no noise, $n^y = 0$. One must first obtain the optimal measurement sensitivity matrix F , defined as

$$F = dy^{\text{opt}}/dd.$$

Each column in F expresses the optimal change in the y 's when the independent variable (u) is adjusted so that the system remains optimal with respect to the disturbance d . Usually, it is simplest to obtain F numerically by optimizing the model. Alternatively, we can obtain F from a quadratic approximation of the cost function

$$F = G_d^y - G^y J_{uu}^{-1} J_{ud}$$

Then, assuming that we have at least as many (independent) measurements y as the sum of the number of (independent) inputs (u) and disturbances (d), the optimal is to select $c = H_1 y$ such that

$$H_1 F = 0$$

Note that H_1 is a nonsquare matrix, so $H_1 F = 0$ does not require that $H_1 = 0$ (which is a trivial uninteresting solution), but rather that H_1 is in the nullspace of F^T .

- 2C. **Exact local method (loss method).** This extends the nullspace method to include noise (n^y) and allows for any number of measurements. The noise and disturbances are normalized by introducing weighting matrices W_{ny} and W_d (which have the expected magnitudes along the diagonal) and then the expected loss, $L = J - J_{\text{opt}}(d)$, is minimized by selecting H_1 to solve the following problem

$$\min_{H_1} \|M(H_1)\|_2$$

where $\| \cdot \|_2$ denotes the Frobenius norm and

$$\begin{aligned} M(H_1) &= J_{uu}^{1/2} (H_1 G^y)^{-1} H_1 Y, Y \\ &= [F W_d W_{ny}]. \end{aligned}$$

Note here that the optimal choice with $W_{ny} = 0$ (no noise) is to choose H_1 such that $H_1 F = 0$, which is the nullspace method. For the general case, when H_1 is a “full” matrix, this is a convex problem and the optimal solution is $H_1^T = (Y Y')^{-1} G^y Q$ where Q is any nonsingular matrix.

Regulatory Control Layer

The main purpose of the regulatory layer is to “stabilize” the plant, preferably using a *simple* control structure (e.g., single-loop PID controllers) which does not require changes

during operation. “Stabilize” is here used in a more extended sense to mean that the process does not “drift” too far away from acceptable operation when there are disturbances. The regulatory layer should make it possible to use a “slow” supervisory control layer that does not require a detailed model of the high-frequency dynamics. Therefore, in addition to track the setpoints given by the supervisory layer (e.g., MPC), the regulatory layer may directly control primary variables (CV_1) that require fast and tight control, like economically important active constraints.

In general, the design of the regulatory layer involves the following structural decisions:

1. Selection of controlled outputs y_2 (among all candidate measurements y_m).
2. Selection of inputs $MV_2 = u_2$ (a subset of all available inputs u) to control the outputs y_2 .
3. Pairing of inputs u_2 and outputs y_2 (since decentralized control is normally used).

Decisions 1 and 2 combined (IO selection) is equivalent to selecting H_2 (Fig. 3). Note that we do not “use up” any degrees of freedom in the regulatory layer because the set points (y_{2s}) become manipulated variables (MV_1) for the supervisory layer (see Fig. 3). Furthermore, since the set points are set by the supervisory layer in a cascade manner, the system eventually approaches the same steady-state (as defined by the choice of economic variables CV_1) regardless of the choice of controlled variables in the regulatory layer.

The inputs for the regulatory layer (u_2) are selected as a subset of all the available inputs (u). For stability reasons, one should avoid input saturation in the regulatory layer. In particular, one should avoid using inputs (in the set u_2) that are optimally constrained in some disturbance region. Otherwise, in order to avoid input saturation, one needs to include a backoff for the input when entering this operational region, and doing so will have an economic penalty.

In the regulatory layer, the outputs (y_2) are usually selected as individual measurements and they are often not important variables in them-

selves. Rather, they are “extra outputs” that are controlled in order to “stabilize” the system, and their setpoints (y_{2s}) are changed by the layer above, to obtain economical optimal operation. For example, in a distillation column one may control a temperature somewhere in the middle of the column ($y_2 = T$) in order to “stabilize” the column profile. Its setpoint ($y_{2s} = T_s$) is adjusted by the supervisory layer to obtain the desired product composition ($y_1 = c$).

Input-Output (IO) Selection for Regulatory Control (u_2, y_2)

Finding the truly optimal control structure, including selecting inputs and outputs for regulatory control, requires finding also the optimal controller parameters. This is an extremely difficult mathematical problem, at least if the controller K is decomposed into smaller controllers. In this section, we consider some approaches which does not require that the controller parameters be found. This is done by making assumptions related to achievable control performance (controllability) or perfect control.

Before we look at the approaches, note again that the IO-selection for regulatory control may be combined into a single decision, by considering the selection of

$$CV_2 = [y_2; u_1] = H_2 y$$

Here u_1 denotes the inputs that are not used by the regulatory control layer. This follows because we want to use all inputs u for control, so assuming that the set u is given, “selection of inputs u_2 ” (decision 2) is by elimination equivalent to “selection of inputs u_1 .” Note that CV_2 include all variables that we keep at desired (constant) values within the fast time horizon of the regulatory control layer, including the “unused” inputs u_1

Survey by Van de Wal and Jager

Van de Wal and Jager provide an overview of methods for input-output selection, some of which include:

1. “Accessibility” based on guaranteeing a cause–effect relationship between the selected inputs (u_2) and outputs (y_2). Use of such measures may eliminate unworkable control structures.
2. “State controllability and state observability” to ensure that any unstable modes can be stabilized using the selected inputs and outputs.
3. “Input-output controllability” analysis to ensure that y_2 can be acceptably controlled using u_2 . This is based on scaling the system, and then analysing the transfer matrices $G_2(s)$ (from u_2 to y_2) and G_{d2} (from expected disturbances d to y_2). Some important controllability measures are right half plane zeros (unstable dynamics of the inverse), condition number, singular values, relative gain array, etc. One problem here is that there are many different measures, and it is not clear which should be given most emphasis.
4. “Achievable robust performance.” This may be viewed as a more detailed version of input-output controllability, where several relevant issues are combined into a single measure. However, this requires that the control problem can actually be formulated clearly, which may be very difficult, as already mentioned. In addition, it requires finding the optimal robust controller for the given problem, which may be very difficult.

Most of these methods are useful for analyzing a given structure (u_2, y_2) but less suitable for selection. Also, the list of methods is also incomplete, as disturbance rejection, which is probably the most important issue for the regulatory layer, is hardly considered.

A Systematic Approach for IO-Selection Based on Minimizing State Drift Caused by Disturbances

The objectives of the regulatory control layer are many, and Yelchuru and Skogestad (2013) list 13 partly conflicting objectives. To have a truly systematic approach to regulatory control design, including IO-selection, we would need to quantify all these partially conflicting objectives in terms of a scalar cost function J_2 . We here

consider a fairly general cost function,

$$J_2 = ||Wx||$$

which may be interpreted as the weighted state drift. One justification for considering the state drift, is that the regulatory layer should ensure that the system, as measured by the weighted states Wx , does not drift too far away from the desired state, and thus stays in the “linear region” when there are disturbances. Note that the cost J_2 is used to *select* controlled variables (CV_2) and not to design the controller (for which the cost may be the control error, $J_2' = ||CV_2 - CV_{2s}||$).

Within this framework, the IO-selection problem for the regulatory layer is then to select the nonsquare matrix H_2 ,

$$CV_2 = H_2 y$$

where $y = [y_m; u]$, such that the cost J_2 is minimized. The cause for changes in J_2 are disturbances d , and we consider the linear model (in deviation variables)

$$\begin{aligned} y &= G^y u + G_d^y d \\ x &= G^x u + G_d^x d \end{aligned}$$

where the G -matrices are transfer matrices. Here, G_d^x gives the effect of the disturbances on the states with no control, and the idea is to reduce the disturbance effect by closing the regulatory control loops. Within the “slow” time scale of the supervisory layer, we can assume that CV_2 is perfectly controlled and thus constant, or $CV_2 = 0$ in terms of deviation variables. This gives

$$CV_2 = H_2 G^y u + H_2 G_d^y d = 0$$

and solving with respect to u gives

$$u = - (H_2 G^y)^{-1} (H_2 G_d^y) d$$

and we have

$$x = P_d^x (H_2) d$$

where

$$P_d^x(H_2) = G_d^x - G^x (H_2 G^y)^{-1} H_2 G_d^y$$

is the disturbance effect for the “partially” controlled system with only the regulatory loops closed. Note that it is not generally possible to make $P_d^x = 0$ because we have more states than we have available inputs. To have a small “state drift,” we want $J_2 = \|W P_d d\|$ to be small, and to have a simple regulatory control system we want to close as few regulatory loops as possible. Assume that we have normalized the disturbances so that the norm of d is 1, then we can solve the following problem

For $0, 1, 2, \dots$ etc. loops closed solve:
 $\min_{H_2} \|M_2(H_2)\|$

where $M_2 = W P_d^x$ and $\dim(u_2) = \dim(y_2) =$
 no. of loops closed.

By comparing the value of $\|M_2(H_2)\|$ with different number of loops closed (i.e., with different H_2), we can then decide on an appropriate regulatory layer structure. For example, assume that we find that the value of J_2 is 110 (0 loops closed), 0.2 (1 loop), and 0.02 (2 loops), and assume we have scaled the disturbances and states such that a J_2 -value less than about 1 is acceptable, then closing 1 regulatory loop is probably the best choice.

In principle, this is straightforward, but there are three remaining *issues*: (1) We need to choose an appropriate norm, (2) we should include measurement noise to avoid selecting insensitive measurements and (3) the problem must be solvable numerically.

Issue 1. The norm of M_2 should be evaluated in the frequency range between the “slow” bandwidth of the supervisory control layer (ω_{B1}) and the “fast” bandwidth of the regulatory control layer (ω_{B2}). However, since it is likely that the system sometimes operates without the supervisory layer, it is reasonable to evaluate the norm of P_d^x in the frequency range from 0 (steady state) to ω_{B2} . Since we want H_2 to be a constant (not frequency-dependent) matrix, it is reasonable to choose H_2 to minimize the norm of M_2 at the frequency where $\|M_2\|$ is expected to have its peak. For some mechanical systems, this may

be at some resonance frequency, but for process control applications it is usually at steady state ($\omega = 0$), that is, we can use the steady-state gain matrices when computing P_d^x . In terms of the norm, we use the 2-norm (Frobenius norm), mainly because it has good numerical properties, and also because it has the interpretation of giving the expected variance in x for normally distributed disturbances.

Issues 2 and 3. If we include also measurement noise n^y , which we should, then the expected value of J_2 is minimized by solving the problem $\min_{H_2} \|M_2(H_2)\|_2$ where (Yelchuru and Skogestad 2013)

$$M_2(H_2) = J_{2uu}^{1/2} (H_2 G^y)^{-1} H_2 Y_2$$

$$\begin{aligned} Y_2 &= [F_2 W_d \ W_n]; \quad F_2 = \frac{\partial y_{\text{opt}}}{\partial d} \\ &= G^y J_{2uu}^{-1} J_{2ud} - G_{dy} \end{aligned}$$

where $J_{2uu} \triangleq \frac{\partial^2 J_2}{\partial u^2} = 2G^{xT} W^T W G^x$, $J_{2ud} \triangleq \frac{\partial^2 J_2}{\partial u \partial d} = 2G^{xT} W^T W G_{dx}$,

Note that this is the same mathematical problem as the “exact local method” presented for selecting $CV_1 = H_1 y$ for minimizing the economic cost J , but because of the specific simple form for the cost J_2 , it is possible to obtain analytical formulas for the optimal sensitivity, F_2 . Again, W_d and W_{ny} are diagonal matrices, expressing the expected magnitude of the disturbances (d) and noise (for y).

For the case when H_2 is a “full” matrix, this can be reformulated as a convex optimization problem and an explicit solution is

$$H_2^T = (Y_2 Y_2^T)^{-1} G^y (G^y)^T (Y_2 Y_2^T)^{-1} G^y)^{-1} J_{2uu}^{1/2}$$

and from this we can find the optimal value of J_2 . It may seem restrictive to assume that H_2 is a “full” matrix, because we usually want to control individual measurements, and then H_2 should be a selection matrix, with 1’s and 0’s. Fortunately, since we in this case want to control as many measurements (y_2) as inputs (u_2), we have that H_2 is square in the selected set, and the opti-

mal value of J_2 when H_2 is a selection matrix is the same as when H_2 is a full matrix. The reason for this is that specifying (controlling) any linear combination of y_2 , uniquely determines the individual y_2 's, since $\dim(u_2) = \dim(y_2)$. Thus, we can find the optimal selection matrix H_2 , by searching through all the candidate square sets of y . This can be effectively solved using the branch and bound approach of Kariwala and Cao, or alternatively it can be solved as a mixed-integer problem with a quadratic program (QP) at each node (Yelchuru and Skogestad 2012). The approach of Yelchuru and Skogestad can also be applied to the case where we allow for disjunct sets of measurement combinations, which may give a lower J_2 in some cases.

Comments on the state drift approach.

1. We have assumed that we perfectly control y_2 using u_2 , at least within the bandwidth of the regulatory control system. Once one has found a candidate control structure (H_2), one should check that it is possible to achieve acceptable control. This may be done by analyzing the input-output controllability of the system $y_2 = G_2 u_2 + G_{2d} d$, based on the transfer matrices $G_2 = H_2 G^y$ and $G_{2d} = H_2 G_d^y$. If the controllability of this system is not acceptable, then one should consider the second-best matrix H_2 (with the second-best value of the state drift J_2) and so on.
2. The state drift cost drift $J_2 = ||Wx||$ is in principle independent of the economic cost (J). This is an advantage because we know that the economically optimal operation (e.g., active constraints) may change, whereas we would like the regulatory layer to remain unchanged. However, it is also a disadvantage, because the regulatory layer determines the initial response to disturbances, and we would like this initial response to be in the right direction economically, so that the required correction from the slower supervisory layer is as small as possible. Actually, this issue can be included by extending the state vector x to include also the economic controlled variables, CV_1 , which is selected based on the economic cost J . The weight matrix W may then be used to adjust the relative weights of avoiding drift in the internal states x and economic controlled variables CV_1 .
3. The above steady-state approach does not consider input-output pairing, for which dynamics are usually the main issue. The main pairing rule is to “pair close” in order to minimize the effective time delay between the selected input and output. For a more detailed approach, decentralized input-output controllability must be considered.

Summary and Future Directions

Control structure design involves the structural decisions that must be made before designing the actual controller, and it is in most cases a much more important step than the controller design. In spite of this, the theoretical tools for making the structural decisions are much less developed than for controller design. This chapter summarizes some approaches, and it is expected, or at least hoped, that this important area will further develop in the years to come.

The most important structural decision is usually related to selecting the economic controlled variables, $CV_1 = H_1 y$, and the stabilizing controlled variables, $CV_2 = H_2 y$. However, control engineers have traditionally not used the degrees of freedom in the matrices H_1 and H_2 , and this chapter has summarized some approaches.

There has been a belief that the use of “advanced control,” e.g., MPC, makes control structure design less important. However, this is not correct because also for MPC must one choose inputs ($MV_1 = CV_{2s}$) and outputs (CV_1). The selection of CV_1 may to some extent be avoided by use of “Dynamic Real-Time Optimization (DRTO)” or “Economic MPC,” but these optimizing controllers usually operate on a slower time scale by sending setpoints to the basic control layer ($MV_1 = CV_{2s}$), which means that selecting the variables CV_2 is critical for achieving (close to) optimality on the fast time scale.

Cross-References

- [Control Hierarchy of Large Processing Plants: An Overview](#)
- [Industrial MPC of Continuous Processes](#)
- [PID Control](#)

Bibliography

- Alstad V, Skogestad S (2007) Null space method for selecting optimal measurement combinations as controlled variables. *Ind Eng Chem Res* 46(3): 846–853
- Alstad V, Skogestad S, Hori ES (2009) Optimal measurement combinations as controlled variables. *J Process Control* 19:138–148
- Downs JJ, Skogestad S (2011) An industrial and academic perspective on plantwide control. *Ann Rev Control* 17:99–110
- Engell S (2007) Feedback control for optimal process operation. *J Proc Control* 17:203–219
- Foss AS (1973) Critique of chemical process control theory. *AIChE J* 19(2):209–214
- Kariwala V, Cao Y (2010) Bidirectional branch and bound for controlled variable selection. Part III. Local average loss minimization. *IEEE Trans Ind Inform* 6: 54–61
- Kookos IK, Perkins JD (2002) An Algorithmic method for the selection of multivariable process control structures. *J Proc Control* 12:85–99
- Morari M, Arkun Y, Stephanopoulos G (1973) Studies in the synthesis of control structures for chemical processes. Part I. *AIChE J* 26:209–214
- Narraway LT, Perkins JD (1993) Selection of control structure based on economics. *Comput Chem Eng* 18:S511–S515
- Skogestad S (2000) Plantwide control: the search for the self-optimizing control structure. *J Proc Control* 10:487–507
- Skogestad S (2004) Control structure design for complete chemical plants. *Comput Chem Eng* 28(1–2):219–234
- Skogestad S (2012) Economic plantwide control, chapter 11. In: Rangaiah GP, Kariwala V (eds) *Plantwide control. Recent developments and applications*. Wiley, Chichester, pp 229–251. ISBN:978-0-470-98014-9
- Skogestad S, Postlethwaite I (2005) *Multivariable feedback control*, 2nd edn. Wiley, Chichester
- van de Wal M, de Jager B (2001) Review of methods for input/output selection. *Automatica* 37:487–510
- Yelchuru R, Skogestad S (2012) Convex formulations for optimal selection of controlled variables and measurements using Mixed Integer Quadratic Programming. *J Process Control* 22:995–1007
- Yelchuru R, Skogestad S (2013) Quantitative methods for regulatory layer selection. *J Process Control* 23: 58–69

Control System Optimization Methods in the Frequency Domain

Christina M. Ivler

Shiley School of Engineering, University of Portland, Portland, OR, USA

Abstract

When designing a compensator, there are typically a range of solutions that provide a closed-loop response meeting the design requirements. An optimal solution is one that meets the design specifications while minimizing or maximizing an objective function. The process for formulating control compensator design as an optimization problem is described in this article. An example case is given to illustrate control system optimization methods in the frequency domain.

Keywords

Multi-objective optimization · Specifications · Design parameters · Summed objectives · Pareto-optimal · Crossover frequency · Actuator activity

Introduction

As discussed in ► [“Classical Frequency-Domain Design Methods”](#), classical frequency domain methods can be used for selecting a dynamic compensator $D(s)$ that meets the design requirements for a closed-loop system, illustrated by Fig. 1. The method presented herein is a more sophisticated extension of that, where the goal is to determine an optimum compensator $D(s)$. Minimizing actuator activity of the control system is a common optimization objective. Additionally, minimizing the gain at higher frequencies reduces sensitivity to noise and eliminates unnecessary actuator fatigue. The design specifications are formulated in the fre-

quency domain for ease and speed of evaluation. The control designer selects a compensation structure, (e.g., lead-lag, proportional-integral-derivative), and the associated design parameters to be optimized within the compensator (e.g., lead frequency, proportional gain, integral gain). Then a multi-objective optimization engine, such as CONDUIT[®] in Tischler et al. (2017), can be used to find the optimal settings for those compensator design parameters.

Design Specifications

Of key importance to successful control system design, including design using optimization, is the careful selection of the requirements. When developing the compensator, a set of comprehensive specifications is selected to guide the design. This set prescribes the behavior of the desired system, typically covering desired stability robustness, command tracking, steady-state error, damping ratio, and disturbance rejection capabilities of the system. Frequency-response-based specifications are recommended because of the ease and speed of evaluation and generally linear variation of frequency-response metrics with design parameters. In contrast, time domain specifications require time simulations, which are more computationally expensive to run and evaluate than frequency-response-based specifications. This is especially important for direct optimization methods because many iterations and subsequent evaluations of the specifications are required.

To ensure stability, stable eigenvalue (λ_j) would be selected as a design requirement, $Re(\lambda_j) \leq 0$. Then minimum gain margin (GM) and phase margin (PM), defined in ►“Frequency-Response and Frequency-Domain Models”, are typically specified to ensure sufficient stability robustness such that magnitude and phase variation can be tolerated prior to onset of instability. A key command tracking requirement is bandwidth, ω_{BW} also defined in ►“Frequency-Response and Frequency-Domain

Models”. Bandwidth is a measure of the maximum input frequency for which the system has acceptable tracking and characterizes the speed of the response. The resonant peak, M_r , which is the maximum amplitude of the closed-loop frequency response as in ►“Frequency-Response and Frequency-Domain Models”, is another possible design specification which can be used to characterize allowable overshoot. The allowable steady-state error (e_{ss}) is another common command tracking requirement. A minimum damping ratio ζ for all eigenvalues may also be prescribed to ensure that the system has sufficient damping. If the control designer is concerned with disturbance rejection capabilities, minimum acceptable of disturbance rejection bandwidth ω_{DRB} and maximum allowable disturbance rejection peak M_d , both of which are defined in ►“Frequency-Response and Frequency-Domain Models” may be specified as additional design requirements. Disturbance rejection bandwidth defines the speed of recovery from disturbances, and the disturbance rejection peak describes the overshoot of the disturbance response.

The exact design specifications will vary based on industry best practices and standards, depending on the intended application.

Objective Functions

In the case where multiple compensator solutions meet the set of design specifications, it seems natural to choose a design that is optimal in some way. Actuator usage is often selected as an objective function to minimize, which can provide many benefits:

- Improves stability robustness,
- More efficient energy consumption of the actuators,
- Relieves actuator saturation, which can cause undesirable behavior, such as limit cycles.

Moreover, minimizing high-frequency actuator activity has several additional benefits:

- Reduces likelihood of exciting higher frequency unmodeled dynamics, such as structural modes of the physical system,
- Lessens sensitivity to sensor noise, which typically occurs at higher frequency than the plant dynamics (if the sensor is chosen appropriately).

For the system in Fig. 1, actuator response to a reference input is given by $A(s)$:

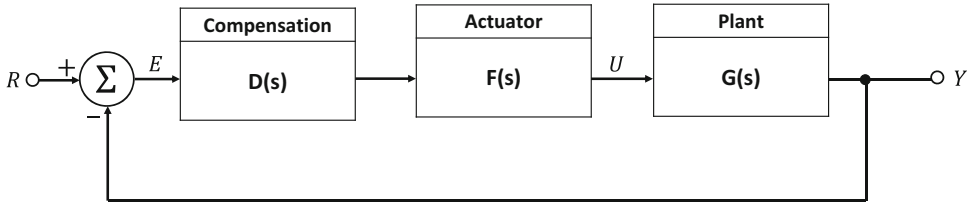
$$A(s) = \frac{U(s)}{R(s)} = \frac{D(s)F(s)}{1 + D(s)F(s)G(s)} \quad (1)$$

A metric for the actuator usage is the variance of the actuator response σ_a^2 . In the frequency domain, σ_a^2 due to a white noise input, which excites the system at all frequencies, is proportional to the area under of the square of the magnitude response of $A(s)$ as described in (Tischler et al. 2017):

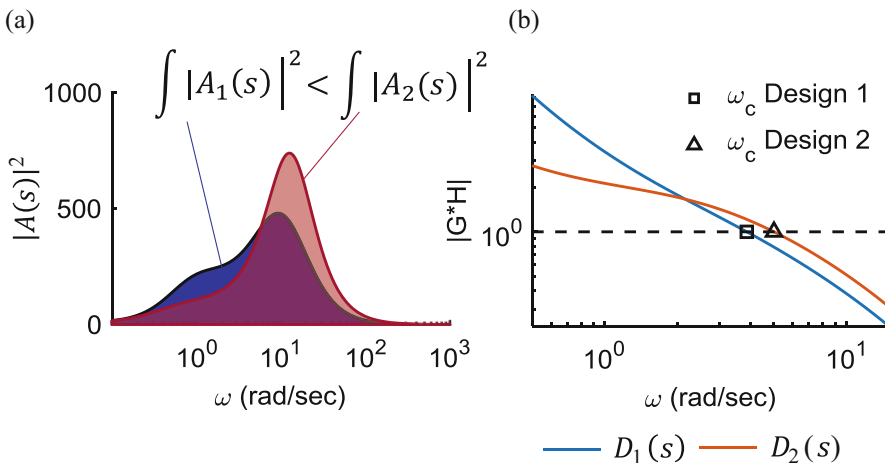
$$f_{\text{obj}_1} = \int_0^\infty |A(j\omega)|^2 d\omega. \quad (2)$$

Thus by minimizing the objective f_{obj_1} , actuator usage is also minimized. The objective function f_{obj_1} can be visualized as the shaded area under the plot of $|A(s)|^2$, illustrated in Fig. 2a for two different compensators $D_1(s)$ and $D_2(s)$, applied to the same plant $G(s)$. For practicality, the integral is only evaluated over a finite region, as shown in Fig. 2.

By itself f_{obj_1} provides information about the magnitude of the overall actuator usage but not the frequency content of the signal. A second objective is needed to ensure that high-frequency actuator activity is minimized. A recommended f_{obj_2} is the open-loop magnitude 1 (0db) crossover frequency ω_c which characterizes the frequency where the combination of the plant and compensator start to attenuate:



Control System Optimization Methods in the Frequency Domain, Fig. 1 Feedback system showing compensation



Control System Optimization Methods in the Frequency Domain, Fig. 2 (a) Actuator activity in the frequency domain (b) Open-loop crossover frequency

$$D(j\omega_c)F(j\omega_c)G(j\omega_c) = 1 \quad f_{\text{obj}_2} = \omega_c \quad (3) \quad \text{requirement:}$$

In Fig. 2a, it is clear that the total area under the curve cannot capture the differences in the frequency distribution of the actuator activity, which is concentrated at higher frequency in $D_2(s)$. However, the crossover frequency ω_c , which is higher for $D_2(s)$, better captures this effect.

Optimizing for a combination f_{obj_1} and f_{obj_2} ensures that both the overall actuator usage and the frequency of the activity are minimal. A recommended cost function to be minimized is given by:

$$J = \bar{f}_{\text{obj}_1} + \bar{f}_{\text{obj}_2} \quad (4)$$

where $\bar{f}_{\text{obj}_1}$ and $\bar{f}_{\text{obj}_2}$ are normalized (or weighted) versions of f_{obj_1} and f_{obj_2} .

Multi-objective Optimization Formulation

Modern control state-space methods such as LQR, described in Franklin et al. (2019), provide optimal solutions that have stability robustness while minimizing actuator activity. However, there is no direct way to tune weighting matrices that are used in modern control methods (such as Q and R) to simultaneously meet the design requirements for key frequency domain design specifications like bandwidth, peak resonance, disturbance rejection bandwidth, etc. In the case that one or multiple frequency domain design requirements are in place, a direct multi-objective optimization algorithm can be used to find the solution that meets all requirements with minimal actuator activity.

In multi-objective optimization, there are many solvers available and associated methods to frame the problem. A commonly accepted method of formulating an optimization-based control design problem, used by Tischler et al. (2017), is as a weighted sum optimization of the objective functions $\bar{f}_{\text{obj}_i}$ and where the design criteria $\bar{f}_i$ are reformulated so that a value of less than zero indicates compliance with the

$$\begin{aligned} &\text{minimize } J(x) = \sum \bar{f}_{\text{obj}_i}, \\ &\text{subject to : } \bar{f}_i(x) \leq 0, x_{\min} \leq x \leq x_{\max} \end{aligned} \quad (5)$$

The summed objective $J(x)$ to be minimized is a sum of the normalized objective functions. The unknown quantities to determine, x , are the compensator elements to be optimized, for example, the gain and pole/zero locations of a lead compensator. The objective functions are normalized so that they are expressed in equivalent units and to ensure that each individual objective is appropriately weighted in the optimization. A recommended method is to normalize each objective in units of relative “goodness”:

$$\bar{f}_{\text{obj}_i} = \frac{f_{\text{obj}_i} - f_{\text{obj}_{i\text{good}}}}{f_{\text{obj}_{i\text{bad}}} - f_{\text{obj}_{i\text{good}}}}. \quad (6)$$

A comprehensive set of system characteristics f_i are selected with associated design requirements $f_{i\text{good}}$ and $f_{i\text{bad}}$. After these are determined, then the normalized design specifications can be formulated similarly to the design objectives:

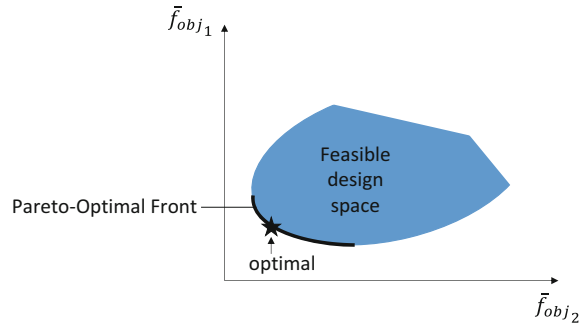
$$\bar{f}_i = \frac{f_i - f_{i\text{good}}}{f_{i\text{bad}} - f_{i\text{good}}}, \quad (7)$$

such that $\bar{f}_i = 0$ when the design characteristic is exactly equal to the design requirement $f_{i\text{good}}$. Any values $\bar{f}_i(x) \leq 0$ indicate that design requirement has been achieved. This methodology is used in the CONDUIT[®] software as described by Tischler et al. (2017). It is an important step that ensures all design specifications are equally weighted by their good and bad values so that all the design constraints are in the same normalized unit system. This provides a more balanced design space and better posed optimization.

The optimization concept can be visualized by Fig. 3, which illustrates the concepts of feasible design space, Pareto-optimal front and optimal solution. The feasible design space indicates where there are solutions for x that meet all the design requirements. The optimal point is

Control System Optimization Methods in the Frequency Domain, Fig. 3

Illustration of optimization design space and Pareto-optimal front



where the sum of both objective functions is smallest. The black line represents the Pareto-optimal front, which shows optimal solutions if there was a different weighting of the objective functions. For example, if it was desirable to further reduce (improve) f_{obj1} , the Pareto-optimal trade-off is achieved at the cost of larger (worse) f_{obj2} .

Optimization Process

In the approach presented by Tischler et al. (2017), and integrated into the CONDUIT[®] software, a three-tiered optimization approach is employed. All design specifications are broken into three categories:

- Hard requirements: all design specifications associated with stability,
- Soft requirements: all design specifications that are associated with performance,
- Objective requirements: all design specifications included in the objective function.

The optimization is also split into three Phases:

- Phase 1: The optimization engine first works toward meeting all the hard requirements, while ignoring the soft and objective requirements. This moves the solution into a stable, reasonable design space,
- Phase 2: All soft requirements are optimized to meet the design specification, while still enforcing the hard requirements,

- Phase 3: When a feasible design that meets all hard and soft requirements is achieved, the optimization will then minimize the objective functions within the feasible design space.

As described by Tischler et al. (2017), a robust methodology for control system optimization is achieved with this phased approach. This method guides the optimization toward the feasible design space prior to optimizing the objective cost function. It is not useful to minimize the objectives if the design requirements are not met, as they represent the necessary performance needed for the system to behave satisfactorily. In the case that the design is not feasible or the minimum cost function is still too high after optimization, the design requirements can be relaxed.

Example Case

To illustrate the concept and advantages of direct multi-objective optimization for design of a dynamic compensator $D(s)$, a simple design example is presented for the block diagram in Fig. 1. For this example case, the dynamics of the system to be controlled $G(s)$ can be represented by the following transfer function model:

$$G(s) = \frac{0.2(s + 0.5)}{(s + 3)(s + 0.1)}, \quad (8)$$

That has associated actuator dynamics:

$$F(s) = \frac{20}{s + 20}, \quad (9)$$

And assuming the control designer has selected a compensator of the following form, which is effectively a proportional-integral type control system:

$$D(s) = \frac{k(s+a)}{s}, \quad (10)$$

The resulting closed-loop response $T(s)$ and actuator response $A(s)$ are:

$$T(s) = \frac{G(s)F(s)D(s)}{1 + G(s)F(s)D(s)}, \quad (11)$$

$$A(s) = \frac{G(s)F(s)}{1 + G(s)F(s)D(s)}. \quad (12)$$

The design parameters x of the compensator to be selected via optimization are:

- The gain k ,
- The zero location $-a$.

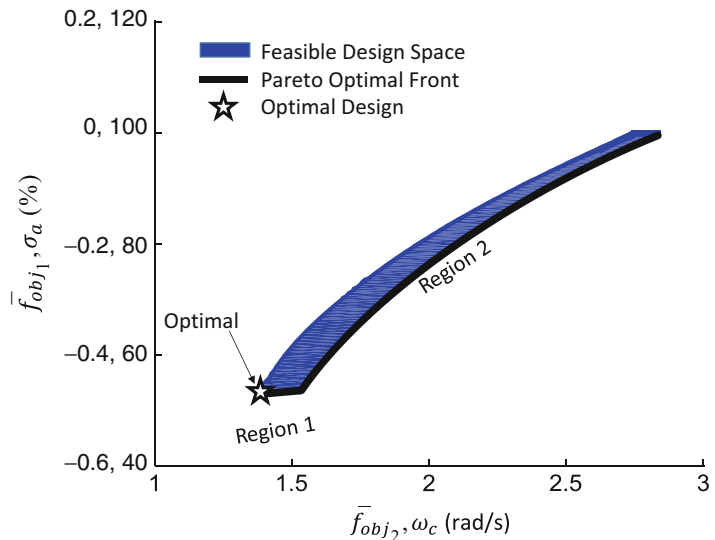
The design specifications and classifications (hard requirement, soft requirement, and objective function) are:

- Stability specifications (Hard requirements): $\text{real}(\lambda_i) \leq 0$, $GM \geq 6$ db, $PM \geq 45$ deg
- Command tracking specifications (Soft requirements): $\omega_{BW} \geq 1.5$ rad/s, $M_r \leq 1$ db, $e_{ss} \leq 5\%$ (step input)

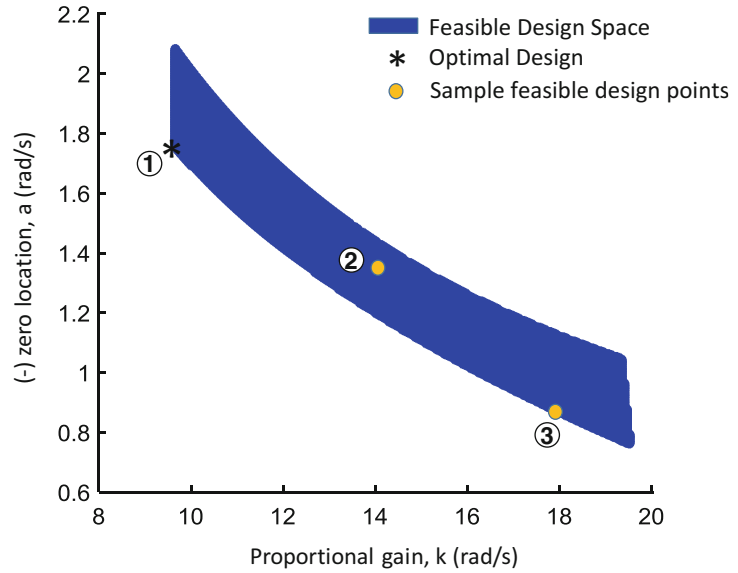
- Disturbance specifications (Soft requirements): $\omega_{DRB} \geq 0.7$ rad/s, $M_d \leq 5$ db
- Summed objective functions: $f_{obj1} = \int_{0.01}^{1000} |A(s)|^2 d\omega$, $f_{obj2} = \omega_c$.

The feasible design space for this problem can be visualized by the chart shown in Fig. 4, which was identified via discrete perturbation of the design parameters and associated mapping of the design space. Due to the low number of design parameters, this is possible for demonstration purposes, but is not as efficient as an optimization engine such as CONDUIT®. This plot shows the optimal solution with an asterisk, and also the Pareto-optimal front which is highlighted in black. Along this Pareto-optimal front, trade-off of increasing crossover frequency, associated with increased performance (higher bandwidth, rise time, etc.), can be achieved at the cost of increased actuator activity. In Region 1 of the Pareto-optimal front, as labeled in Fig. 4, it can be seen that increased performance is achieved with relatively little change in actuator activity. In contrast, further increases in crossover frequency into Region 2 come with a much larger associated cost in actuator activity. The control designer might choose to take advantage of a higher performing design from Region 1 of the Pareto front, so that some design margin is present above the minimum required performance at the cost of slight increase in actuator activity.

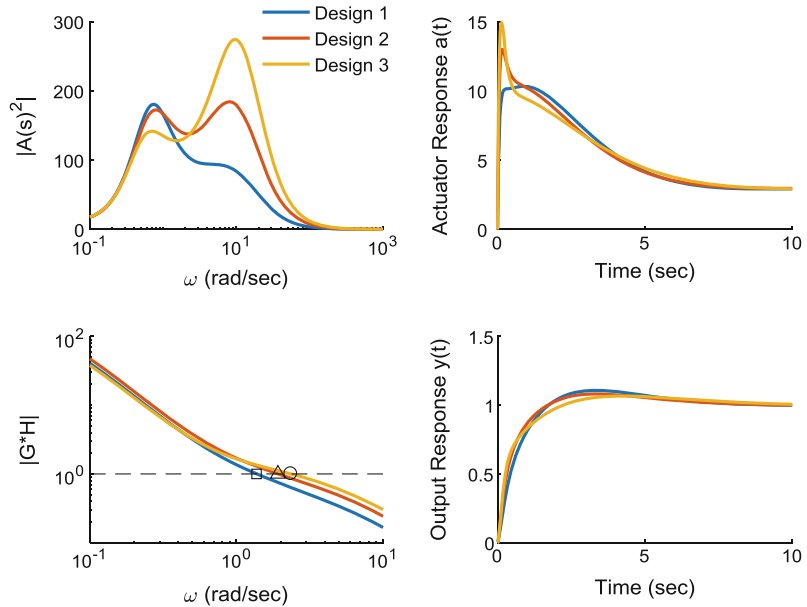
Control System Optimization Methods in the Frequency Domain, Fig. 4 Feasible design space, and optimal solution for example case
 $G(s) = \frac{0.2(s+0.5)}{(s+3)(s+0.1)}$



Control System Optimization Methods in the Frequency Domain, Fig. 5 Feasible design parameter combinations and optimal design point



Control System Optimization Methods in the Frequency Domain, Fig. 6 Optimal and feasible design performance for example case



The optimal design point is shown relative to all feasible values of the design parameters (a , k) in Fig. 5. A comparison of the optimal point with two other feasible, but non-optimal, designs from Fig. 5 was performed. The two non-optimal design points selected in Fig. 5 are shown with circle markers, and the optimal design point with an asterisk. Figure 6 shows key design features for each of these three designs from Fig. 5. While all three designs meet the design specifications, it is clear from Fig. 6a that the optimal design

has reduced actuator activity and has less high-frequency content than the other designs. The time-domain performance of the optimal closed-loop response is slightly slower (associated with lower bandwidth) than the non-optimal solutions because the optimal solution provides the minimum performance needed to just meet the design requirement, as this solution provides minimum actuator usage. The time-history response of the actuator confirms that its usage is reduced for the optimal design, as expected.

Summary and Future Directions

Control system optimization is a method for determining a dynamic compensator design that meets all the specifications with minimum actuator activity. The selection of the design specifications, classification of the specifications, and objective functions significantly affect the optimal solution. These should be carefully selected according to industry standards for the application at hand.

Cross-References

- ▶ [Classical Frequency-Domain Design Methods](#)
- ▶ [Frequency-Response and Frequency-Domain Models](#)

Bibliography

- Franklin GF, Powell JD, Emami-Naeini A (2019) Feedback control of dynamic systems, 8th edn. Pearson, Upper Saddle River
- Tischler MB, Berger T, Ivler CM, Mansur, HM, Soong, J (2017) Practical methods for aircraft and rotorcraft flight control design using multi-objective parameter optimization. AIAA education series. Reston, VA

Control Systems for Nanopositioning

Giovanni Cherubini, Angeliki Pantazi,
Abu Sebastian, Mark Lantz, and
Evangelos Eleftheriou
IBM Research – Zurich, Rüschlikon,
Switzerland

Abstract

An essential role in nanotechnology applications is played by nanopositioning control systems, which allow nanometer-scale precision and accuracy at dimensions of tens of nanometers or less. Nanopositioners are precision mechatronics systems aiming at moving or identifying objects over a certain

distance with a resolution of the order of a nanometer. In nanopositioners, actuation, position sensing, and feedback control are the key components that determine how successfully the stringent requirements on resolution, accuracy, stability, and bandwidth are achieved. Historically, nanopositioning found its earlier applications in scanning probe microscopy. Today it turns out to be increasingly important for lithography tools, semiconductor inspection systems, molecular biology, metrology, nanofabrication, and nanomanufacturing. Moreover, nanopositioning control represents a fundamental requirement in storage systems ranging from probe-based storage devices to mechatronic tape drive systems to achieve ultrahigh areal density and storage capacity.

Keywords

Scanning probe microscopy · Feedback control systems · Robust control · Data storage systems · Nanolithography

Introduction

The past three decades have witnessed the emergence and explosive growth of nanoscience and nanotechnology, which significantly advanced our understanding and control of physical processes on the nanometer and subnanometer scale. Nanopositioning control systems are at the core of this revolution. Nanopositioners may be viewed as precision mechatronics systems that are capable of positioning of samples with a resolution that could be as high as a fraction of a nanometer to enable interrogation and modification of the sample surface. The early applications were found in scanning probe microscopy (Salapaka et al. 2002; Devasia et al. 2007). Moving or identifying objects with a scanning probe microscope (SPM) requires positioning systems with atomic-scale resolution. SPM is one of the most important tools in nanoscale science and engineering. For example, SPM has enabled the visualization of individual atoms and even of chemical bonds between

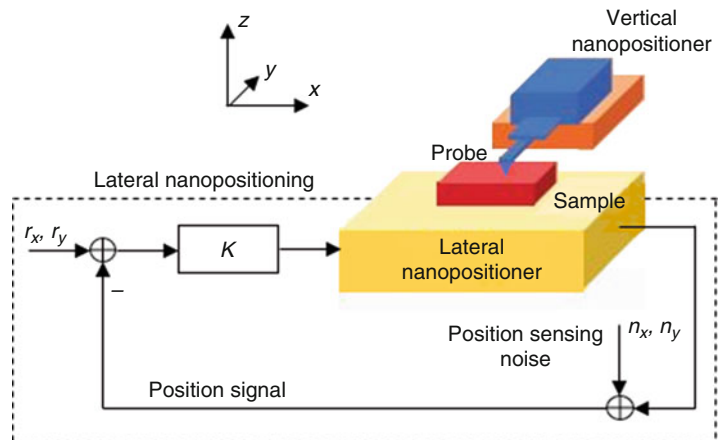
atoms in a single molecule. More recently, nanopositioning systems have become key elements of novel lithography tools that represent an alternative to optical lithography systems (Oomen et al. 2014). During the production process of integrated circuits, a wafer scanner system must track a predefined reference trajectory with extreme accuracy in six motion degrees of freedom (DOFs) for mask alignment and semiconductor inspection. Furthermore, nanopositioning systems are essential for imaging, alignment, and nanomanipulation as required, for example, in molecular biology for cell tracking and DNA analysis (Ando et al. 2007) and in metrology, nanoassembly, and nanomanufacturing applications. Finally, they are also essential for data storage devices where accurate positioning of write/read elements is required to achieve ultrahigh areal density and near-optimum capacity, such as hard disk drives (HDDs), tape drives, and optical drives (Cherubini et al. 2012). For example, nanometer-scale precision was demonstrated in probe storage devices with areal densities of multi-terabit per square inch (Tbit/in²) (Sebastian et al. 2008). The extensive range of applications with operation under vastly diverse conditions poses significant challenges for the control of nanopositioning devices, as they necessitate high resolution, high bandwidth, and robust control designs.

Nanopositioning for Scanning Probe Microscopy

Figure 1 depicts a typical SPM device, where the sample under investigation is positioned relative to an atomically sharp probe tip mounted on a micromechanical cantilever spring that interacts with the sample (Tuma et al. 2013). A feedback loop provides the required accuracy and precision for lateral nanopositioning of an electromechanical device, known as nanopositioner or scanner. Vertical nanopositioning, not depicted in the figure, regulates the force between the tip and the sample (Sebastian et al. 2003). Actuation and sensing objectives are significantly different for the lateral and vertical positioners. Therefore, lateral and vertical nanopositioning are usually decoupled. An example of vertical nanopositioning system is the self-tuning proportional-integral (PI) control system (Tajaddodianfar et al. 2018) introduced to improve the robustness of scanning tunneling microscopes (STMs) against tip crashing into the sample surface, while patterning the surface with a resolution down to a single atom through lithography or subsequent imaging. It turns out that the stability of the STM vertical control system is affected by variations in the local tunnel barrier height, a local quantum mechanical property that depends on the physical properties of the tip apex and of those of the sample surface atoms

Control Systems for Nanopositioning,

Fig. 1 Lateral nanopositioning in SPM



into which the current tunnels. In the remaining part of this section, however, we will focus on lateral nanopositioning. The primary objective in lateral nanopositioning is to make the scanner follow, or track, a given reference trajectory by an appropriate control method, which may rely on feedforward control, feedback control, or combinations thereof. Over the wide bandwidths of the order of 10 kHz or more required by applications at high scanner speeds, the model uncertainties, mechanical cross-coupling, and actuation noise cannot be considered negligible. Hence feedforward control alone is not adequate, and feedback control is essential to achieve nanoscale accuracy and precision at high speeds. A feedback controller, denoted by K in Fig. 1, is used to control the scanner position in the two x and y dimensions, with the objective to follow a predetermined (r_x, r_y) reference trajectory, relying on the positional information provided by a position sensor. The tracking error includes both deterministic and stochastic components, which mainly affect the positioning accuracy and the precision, respectively. Deterministic tracking errors are mostly originated by an insufficient closed-loop bandwidth, mechanical resonant scanner modes, and actuator nonlinearities, e.g., hysteresis. Stochastic tracking errors are typically originated by external vibrations and noise, actuator noise, and position-sensing noise. To minimize both the deterministic and the stochastic tracking error components, it is therefore essential to simultaneously improve the position sensor, the scanning device, the feedback controller, and the reference trajectory.

Low-Noise Sensing

The amount of noise in the position measurement signal determines the sensor resolution over a given bandwidth, which in turn determines the resolution achievable in the nanopositioning feedback system. The resolution is usually characterized in terms of the standard deviation of the sensing noise with the scanner at rest. The resolution therefore deteriorates as the sensing bandwidth increases. The actuator noise can be minimized by carefully designing the drive circuitry. With increasing feedback bandwidth,

however, the sensing noise may enter the feedback loop introducing random positioning errors and adversely affect the positioning precision. The amount of scanner motion induced by the measurement noise may be comparable to the dimensions of the nanoscale structures that are to be resolved. For example, the sensing noise-induced scanner motion may amount to several nanometers over a bandwidth of 10 kHz. The measurement noise can be decreased by using advanced low-noise sensing concepts. Some widely used position sensors that attain nanometer- or subnanometer-scale resolution include capacitive, piezoresistive, optical, and magnetoresistive sensors.

Scanners

Scanner properties eventually determine the achievable performance of a nanopositioning system, the control architectures, the sensing scheme, and the scan trajectories that can be used. An SPM scanner usually must satisfy three requirements: (1) a linear relationship between the actuation signal and the scan table position, with well-defined dynamic behavior; (2) a large range of motion of at least tens to hundreds of micrometers for a variety of applications; and (3) a flat dynamic response over a wide bandwidth to ensure low sensitivity of closed-loop performance to variations in the scanner dynamics and to enable efficient control designs that provide low sensitivity to sensing noise. Most SPM scanners are based on piezoelectric actuation, because of the achievable large bandwidth, robustness, and large actuation force. Tube-based piezoelectric scanners are not expensive and are easy to design and use. However, they are not well suited for high-speed applications due to the relatively high amount of cross-coupling between the positioning axes and the low resonant frequencies. An alternative approach is provided by flexure-guided scanners actuated by piezoelectric stack actuators, where the sample is displaced by elastic deformation of a flexible structure (flexure) that guides the scan table. The main disadvantage of piezoelectric actuators is that they exhibit nonlinear characteristics, to be ascribed to hysteresis, creep, and drift. To

compensate for these nonlinearities, broadband feedback control is typically required, which may negatively affect the closed-loop positioning resolution. Compared to piezoelectric actuators, the dynamics of electromagnetic and electrostatic actuators are much more linear. A disadvantage of electromagnetic and electrostatic actuation, however, is the low actuation force, which limits the achievable range of motion in stiff scanners such as high bandwidth flexure-guided scanners. An elegant approach to decouple the requirements of having a high resonant frequency and a large range is provided by dual-stage positioning, as applied, for example, in optical and magnetic recording. A dual-stage nanopositioner usually comprises a slow, large-range scanner linked in series with a short-range scanner optimized for high speed. In this system, more challenging requirements are imposed on the high-speed scanner. In SPM, only one axis typically requires high-speed actuation, in which case the short-range scanner can be designed solely for uniaxial actuation to allow a clean dynamic response and low interaxial crosstalk. For scanning in other scan directions, as well as scanning over a large area, the large-range scanner is used.

Feedback Control

A suitable feedback control architecture is characterized by good tracking and disturbance rejection performance, while being insensitive to the position-sensing noise. A further requirement is the robustness of the controller to variations in the system mechanical properties. Figure 2 shows three feedback control techniques commonly applied in nanopositioning. Figure 2a depicts the conventional proportional-integral-derivative (PID) feedback control technique for the x positioning axis, where the controller, K_x , includes the proportional, integral, and derivative terms. The controller input is given by the position error signal, e_x , obtained by subtracting the noisy position measurement, $x + n_x$, where the position measurement signal and additive measurement noise are denoted by x and n_x , respectively, from the reference signal, r_x . The performance of PID control is limited in terms

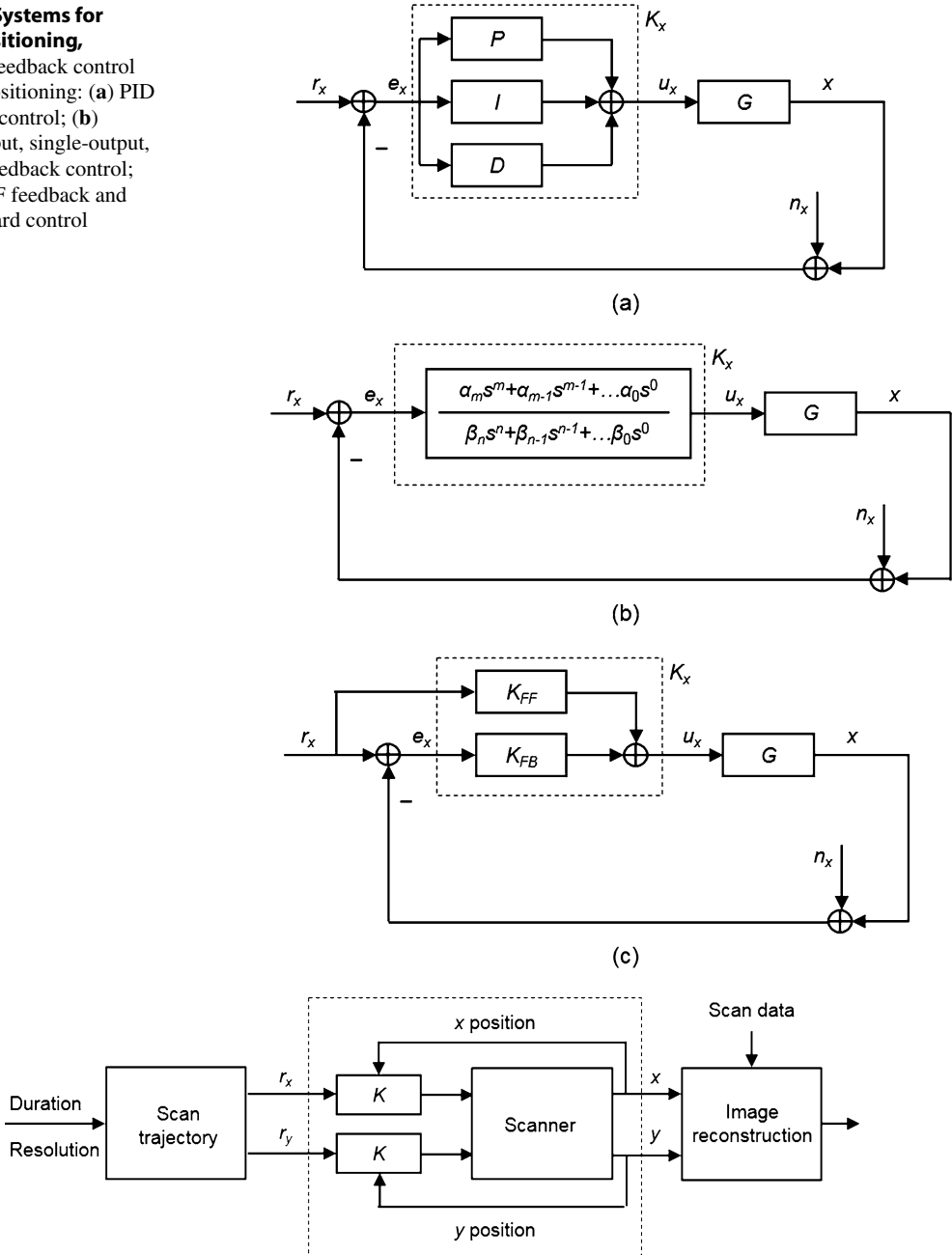
of achieving a high closed-loop bandwidth and robustness to plant uncertainties, which make PID controllers less suitable for high-speed nanopositioning systems in spite of the popularity of the method. Figure 2b shows a 1-DOF feedback control technique, where the feedback controller, K_x , is assumed to be a single-input, single-output dynamical system specifically designed for a given closed-loop performance. In nanopositioning, robust control design methods, such as the H_∞ approach, in which the controller is obtained by mathematical optimization, have been shown to extend the closed-loop bandwidth significantly while maintaining robust performance. However, with 1-DOF controllers, achieving a good tracking performance in the presence of significant sensing noise remains a challenging problem because of inherent limitations such as the waterbed effect. This issue can be overcome by a 2-DOF controller, as depicted in Fig. 2c. In 2-DOF control, the feedback component and the feedforward component of the controller are decoupled. Using this approach, the reference tracking transfer function, from r_x to x , can be designed independently of the noise sensitivity transfer function, from n_x to x . A very high positioning precision can be achieved by designing the controller to carefully shape the closed-loop noise sensitivity transfer function. In applications where the reference signal is of repetitive nature, specialized nonlinear control methods such as adaptive control, repetitive control, and iterative learning control can be used. Advanced control architectures for noise-resilient nanopositioning have been proposed that combine some of the above-mentioned methods, e.g., combining robust control design and shaping of the noise sensitivity transfer function.

Scan Trajectories

The scan trajectory determines the reference signals that must be tracked by the nanopositioner. Therefore, it has a significant impact on the choice of the control technique, the achievable tracking performance, and the sensitivity of the closed-loop system to sensing noise. Figure 3 shows the scan trajectory and image

Control Systems for Nanopositioning,

Fig. 2 Feedback control in nanopositioning: (a) PID feedback control; (b) single-input, single-output, 1-DOF feedback control; (c) 2-DOF feedback and feedforward control



Control Systems for Nanopositioning, Fig. 3 Scan trajectory and image reconstruction in SPM nanopositioning

reconstruction concepts for a nanopositioning system. The scan trajectory reference signals $r_x(t)$ and $r_y(t)$ to be tracked by the nanopositioning system are determined for the required scan duration and spatial resolution. Tracking is usually achieved by a feedback loop relying

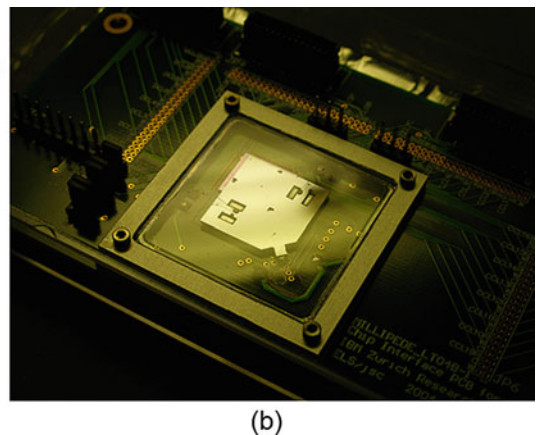
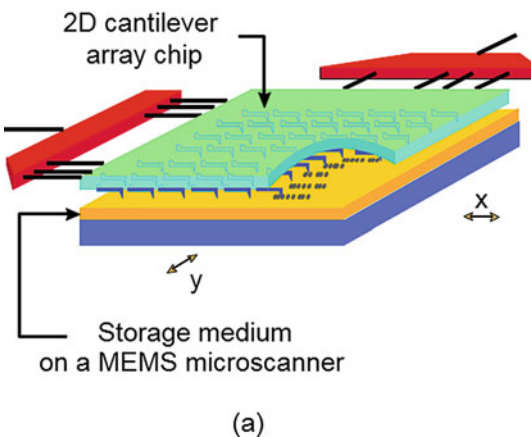
on measurements of the x and the y scanner position. At each position $x(t)$, $y(t)$, a measure of a local surface property referred to as scan data, e.g., the topography, is obtained. For almost all nanopositioning applications, a scan trajectory is provided to systematically traverse

a two-dimensional sample surface to enable imaging and/or manipulation of the sample. A raster scan trajectory is conventionally adopted, which scans the sample in consecutive lines by actuating the scanner with a triangular waveform in one axis and by shifting the sample position in steps or continuously in the orthogonal axis. The raster scan trajectory continues to be the most widely used scan trajectory for both industrial and scientific instruments, mostly because of the simplicity of the image reconstruction. However, with increasing scan speeds, the high-frequency components of the trajectory reference signals excite the mechanical resonant modes of the nanopositioner and introduce unwanted residual vibrations and tracking errors. These high-frequency components come from the sharp turnaround points of the raster scan trajectory and result in a wide spectrum of frequencies of the reference signal, requiring broadband feedback compensation and hence deteriorating the noise resilience of the system. For this reason, the frequency spectrum is often shaped by smoothening the turnaround points. More recently, alternative scan patterns have been proposed that cover the sample area with a desired resolution but at the same time require only reference signals with a narrowband frequency spectrum, e.g., spiral scanning, where

the reference signals are sinusoidal waveforms of varying frequency and/or amplitude.

Nanopositioning for Probe-Based Data Storage

Subnanometer precision at fast rates is required in a wide range of applications, including probe-based data storage (Eleftheriou et al. 2003). Probe-based data storage is an alternative to magnetic storage that has been proposed for both mobile and archival applications because of its ultrahigh areal density, which can achieve several Tbit/in². Among the various probe-based data storage concepts, thermomechanical recording and retrieval of data encoded as nanometer-scale indentations in thin polymer films was the first to be demonstrated in a fully functional probe-based storage device. A schematic illustrating a probe-based storage device and a photograph of a miniaturized device prototype are shown in Fig. 4 (Sebastian et al. 2008). An array of thermomechanical probes is used to read and write information. The data are recorded as indentations on a thin polymer film, which acts as the storage medium. In data storage applications, high bandwidth and high-resolution nanopositioning are essential. This dual objective requires the



Control Systems for Nanopositioning, Fig. 4 Scanning probe-based storage device: (a) diagram of the storage device comprising the two-dimensional microcantilever

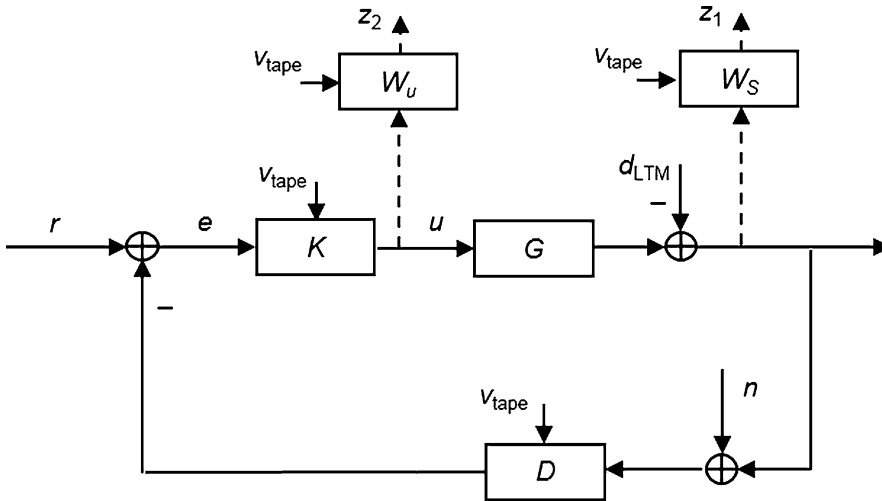
array chip and the storage medium on a MEMS microscanner; (b) photograph of the probe-based storage device prototype

design and development of adequate sensors and actuators as well as suitable control strategies. The positioning system for the scanning probe-based data storage device prototype consists primarily of an electromechanical microscanner that positions the storage medium relative to the cantilever array, thermal position sensors that provide global position information, and servo fields that generate medium-derived position information. Unlike typical SPM nanopositioning applications, which require relative positioning, data storage devices require absolute positioning. Moreover, this absolute positioning needs to be provided across the entire area of the storage field with nanometer-scale precision. Hence, position information must be inscribed on the medium. In the probe storage device prototype, certain storage fields, known as servo fields, and the associated microcantilevers are therefore reserved for the generation of position information. Prewritten servo patterns in the servo fields provide the needed alternative position signal. This signal measures the deviation of the microcantilever tip from the track center during the read/write operation. The medium-derived position error signal (PES) is based on sequences of indentations, which are mutually displaced along the orthogonal direction to the scan direction, arranged in such a way as to produce two signals in quadrature, which can be combined to provide a PES. During regular operation of the device, the nanopositioning system, referred to as the servo system, has two main functions. First, from an arbitrary initial position of the scan table, the target track for read/write operations must be located, which is achieved by the seek-and-settle procedure. Typical worst-case seek times are of the order of 1 ms. The second function of the servo system is to maintain the position of the read/write probes on the center of the target track during the scanning process along the length of this track, which is achieved by the track-follow procedure. During track-follow, each track is scanned in the horizontal direction with constant velocity, while maintaining the fine positioning in the cross-track direction in the presence of disturbances and noise. For probe-based data storage, where the data rate per probe

is relatively low, a linear scan velocity of a few millimeters per second is typically considered. Hence, the requirement on tracking bandwidth is rather modest. In contrast, disturbance rejection is required, but its extent depends on the application. In mobile applications, for example, it is necessary to have good disturbance rejection capability up to about 500 Hz. The higher the requirement on disturbance rejection, the higher the sensitivity of the system to measurement noise. This fundamental trade-off needs to be addressed when designing controllers for probe-based storage devices.

Nanoscale Track-Following in Tape Drives

The volume of digital data produced each year is growing at an ever-increasing pace. Moreover, new regulatory requirements imply that a larger fraction of this data will have to be preserved. All of this translates into a growing need for cost-effective digital archives. State-of-the-art linear tape products found on the market achieve an areal storage density of about 10 Gbit/in² and a cartridge capacity of about 20 terabytes. Future scaling will require a dramatic increase in track density and hence the improvement of the positioning accuracy of the read/write heads at the nanoscale. A single-channel tape areal density of 201 Gbit/in² on a prototype sputtered magnetic tape medium has been recently demonstrated, which would correspond to a cartridge capacity of about 330 terabytes (Furrer et al. 2018a). The lateral position of the head is derived from the relative timing of bursts of pulses in the servo readback waveforms, which are obtained by reading servo patterns that are written on the tape with opposite or identical azimuth angles. A closed-loop control system has the task of following with high accuracy the lateral tape motion (LTM) while attenuating effects from noise in the position measurement. These conflicting requirements can be systematically addressed using the H_∞ control framework. Figure 5 depicts a schematic diagram of the H_∞ control formulation for the track-following



Control Systems for Nanopositioning, Fig. 5 Block diagram of the H_∞ control formulation for the track-following controller design

controller design (Pantazi et al. 2015; Furrer et al. 2018b). The lateral dynamics of the head actuator are described by the transfer function G . The transfer function is dominated by a main actuator resonance at a frequency of about 100 Hz and is captured well by a second-order model. The input to the controller, K , is the PES, e , that measures the difference between the target track location on the tape and the estimated head position in the lateral direction. In addition to measurement noise effects, the PES is affected by the estimation delay, captured by D , which is a function of the tape speed v_{tape} . As the tape speed is increased, the servo pattern delay is reduced while at the same time the LTM disturbances, denoted by d_{LTM} , shift to higher frequencies. Because of the speed dependence of both the delay and the frequency characteristics of the lateral tape motion, a set of track-following controllers are designed, with each controller optimized for a given tape speed. Specifically, for each tape speed, the weighting transfer functions W_s and W_u are adapted to capture the control objectives for the LTM disturbance rejection and for limiting the control effort, respectively. For the tape areal density demonstration of 201 Gbit/in², a standard deviation of the PES of less than 6.5 nm was achieved over the speed range of 1.2–4.1 m/s, corresponding to

the full speed range of a typical commercial tape drive.

Summary and Future Directions

The demand for video-rate SPM will increase, particularly in fields that involve the investigation of biological specimens, high-throughput nanomachining, and nanofabrication. Therefore, approaches to improve the scan speed of the SPM, for example, by employing non-raster scan methods, are likely to gain increased use. In this context, internal model controllers, which include the dynamic modes of the reference and disturbance signals in the feedback controller while preserving system stability, can provide adequate tracking and robust performance without imposing a high control bandwidth. The study of the non-linear dynamics of the vertical positioning systems will motivate the development of advanced control systems to detect the distance from the SPM cantilever tip to the sample substrate in a more robust manner. High-density storage technologies based on scanning probe devices will continue to be investigated. Besides the thermomechanical data storage approach discussed earlier, ferroelectric and phase change materials are candidates for next-generation high-density SPM

data storage technologies. Advanced positioning control methods are also envisaged to improve the density of recording in data storage systems, for example, to achieve densities of the order of a few Tbit/in² in HDDs and capacities of the order of a few hundreds of terabytes in tape systems.

Cross-References

- [Non-raster Methods in Scanning Probe Microscopy](#)
- [Robust Control in Gap Metric](#)

Bibliography

- Ando T, Uchihashi T, Kodera N, Yamamoto D, Taniguchi M, Miyagi A, Yamashita H (2007) Highspeed atomic force microscopy for observing dynamic biomolecular processes. *J Mol Recognit Interdiscip J* 20(6):448–458
- Cherubini G, Chung CC, Messner WC, Moheimani SR (2012) Control methods in data-storage systems. *IEEE Trans Control Syst Technol* 20(2):296–322
- Devasia S, Eleftheriou E, Moheimani SR (2007) A survey of control issues in nanopositioning. *IEEE Trans Control Syst Technol* 15(5):802–823
- Eleftheriou E, Antonakopoulos T, Binnig GK, Cherubini G, Despont M, Dholakia A, Duerig U, Lantz MA, Pozidis H, Rothuizen HE, Vettiger P (2003) Millipede – a MEMS-based scanning-probe data-storage system. *IEEE Trans Magn* 39(2):938–945
- Furrer S, Lantz MA, Reininger P, Pantazi A, Rothuizen HE, Cideciyan RD, Cherubini G, Haerberle W, Eleftheriou E, Tachibana J, Sekiguchi N (2018a) 201 Gb/in² recording areal density on sputtered magnetic tape. *IEEE Trans Magn* 54(2):1–8
- Furrer S, Pantazi A, Cherubini G, Lantz MA (2018b) Compressional wave disturbance suppression for nanoscale track-following on flexible tape media. In: *Proceedings of the American Control Conference, Milwaukee*, pp 6678–6683
- Oomen T, van Herpen R, Quist S, van de Wal M, Bosgra O, Steinbuch M (2014) Connecting system identification and robust control for next-generation motion control of a wafer stage. *IEEE Trans Control Syst Technol* 22(1):102–118
- Pantazi A, Furrer S, Rothuizen HE, Cherubini G, Jelitto J, Lantz MA (2015) Nanoscale track-following for tape storage. In: *Proceedings of the American Control Conference, Chicago*, pp 2837–2843
- Salapaka SM, Sebastian A, Cleveland JP, Salapaka MV (2002) High bandwidth nano-positioner: a robust control approach. *Rev Sci Instrum* 73(9):3232–3241
- Sebastian A, Salapaka SM, Cleveland JP (2003) Robust control approach to atomic force microscopy. In: *Proceedings of the IEEE International Conference on Decision and Control, Hawaii*, pp 3443–3444
- Sebastian A, Pantazi A, Pozidis H, Eleftheriou E (2008) Nanopositioning for probe-based data storage. *IEEE Control Syst Mag* 28(4):26–35
- Tajaddodianfar F, Moheimani SOR, Randall JN (2018) Scanning tunneling microscope control: a self-tuning PI controller based on online local barrier height estimation. *IEEE Trans Control Syst Technol* 27(5):2004–2015
- Tuma T, Sebastian A, Lygeros J, Pantazi A (2013) The four pillars of nanopositioning for scanning probe microscopy: the position sensor, the scanning device, the feedback controller, and the reference trajectory. *IEEE Control Syst Mag* 33(6):68–85

Control-Based Segmentation

Liangjia Zhu, Ivan Kolesov, Peter Karasev, and Allen Tannenbaum

Department of Computer Science, Stony Brook University, Stony Brook, NY, USA

Abstract

Image segmentation is a fundamental problem in image processing, computer vision, and medical imaging applications. The design of generic, automated methods that work for various objects and imaging modalities is a daunting task. Instead of proposing new specific segmentation algorithms, in this entry, we review a general design principle on how to integrate user interactions from the perspective of feedback control theory. In particular, impulsive control and Lyapunov stability analysis are employed to design and analyze an interactive segmentation system. Stabilization conditions are derived to guide algorithm design. The effectiveness and robustness of the proposed methodologies are demonstrated.

Keywords

Segmentation · Active contours · Feedback · Observers

Introduction

Given the variability of real-world imagery, one many times needs to supplement the given automatic algorithm with some human interaction, which brings in the entire subject of how this interaction should take place. We propose to use techniques from feedback control for this purpose making use of the ideas elucidated in Karasev et al. (2013) and Zhu et al. (2018). Thus, we study steerable segmentation from a closed-loop perspective. We desire an interactive system to enable a user to create excellent segmentation results with a minimal amount of time and effort. With the rich resources from control theory, we can quantitatively design and evaluate the performance of a control-based segmentation system. Figure 1 sketches the proposed interactive segmentation framework.

The literature for segmentation is much too huge to survey here. The bottom line is that automated segmentation methods have never been widely adopted in real-world settings, such as in the clinic. As a result, users directly participate in the segmentation loop in various image segmentation systems. This body of methods is called *interactive segmentation* (Boykov and Funka-Lea 2006; Vezhnevets and Konouchine 2005). Typically, the user begins initializing the given algorithm and then iteratively modifies the intermediate results until one obtains a satisfactory result. This process may be viewed as a feedback control process, whereby the user's editing action is a controller, the visualization/monitoring is an observer, and the changes of the intermediate results drive the system dynamics. In this sense, the user automatically fills in the "information" gap between an imperfect model to that of a desired segmentation result. However, most of the work along these lines does not explicitly model the user's role from a systems and control point of view. More precisely, user inputs are utilized *passively* without consideration to their contribution to the stability or convergence of the overall system. To the best of our knowledge, there are very few attempts to model segmentation from the perspective of feedback control. The work of Karasev et al. (2013) and Zhu et al. (2018)

precisely considers this task and is summarized in the present entry.

Open-Loop Automatic Segmentation

A large class of segmentation algorithms may be considered as dynamic processes. Starting from a given region, these approaches evolve the region based on certain segmentation criteria. Examples include region growing/competition, classical active contour models, and distance-based segmentation. Typically, the dynamic process can be described by a differential/difference equation, driven by the optimization of certain energy function(al)s. Active contours implemented via level sets are a key example (Osher and Fedkiw 2003).

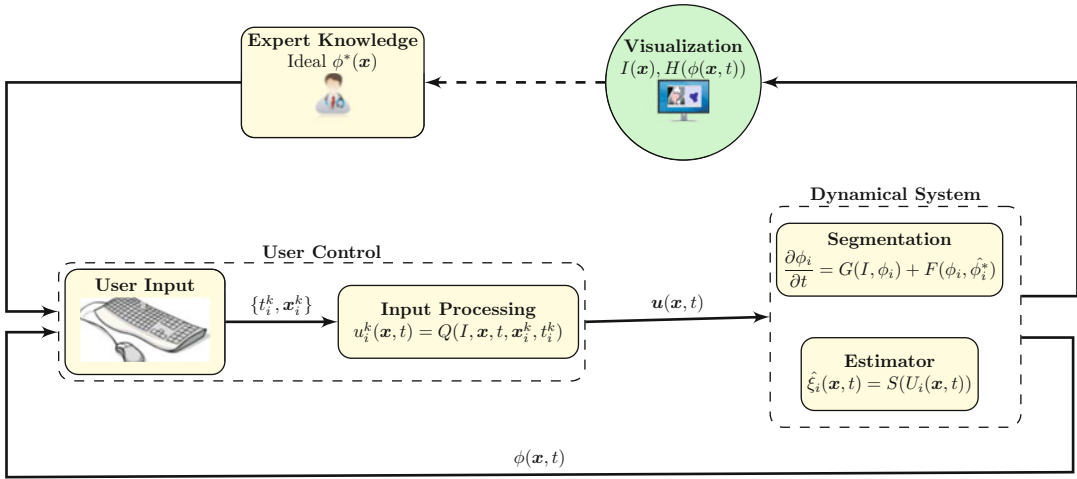
Let $I : \Lambda \rightarrow \mathbb{R}^m$ be an image defined on $\Lambda \subset \mathbb{R}^n$, where $m \geq 1$ and $n \geq 2$. Suppose an image to be segmented consists of N regions. Each region $\Lambda_i(\mathbf{x}, t)$ is associated with a level set function $\psi_i(\mathbf{x}, t)$ and is evolving from an initial state $\psi_i(\mathbf{x}, 0) = \psi_i^0(\mathbf{x})$ with $\{\mathbf{x} | \psi_i^0(\mathbf{x}) = 0\} = \partial \Lambda_i(\mathbf{x}, 0)$. The Heaviside function, denoted by $H(\psi(\mathbf{x}))$, is used to indicate the exterior and interior regions, and its derivative is denoted by $\delta(\psi(\mathbf{x}))$. The dynamical system describing the evolution is defined as follows:

$$\begin{aligned} \frac{\partial \psi_i(\mathbf{x}, t)}{\partial t} &= G(\psi_i(\mathbf{x}, t), I) \delta(\psi_i(\mathbf{x}, t)) \\ \psi_i(\mathbf{x}, 0) &= \psi_i^0(\mathbf{x}) \end{aligned} \quad (1)$$

where $G(\psi_i(\mathbf{x}, t), I)$ is a generic term representing an image force and the interactions between ψ_i and ψ_j , where $j \neq i$. We decompose $G(\psi_i(\mathbf{x}, t), I)$ into two *competing* components as

$$\begin{aligned} G(\psi_i(\mathbf{x}, t), I) &= -(g(\psi_i(\mathbf{x}, t), I) \\ &\quad - g^c(\psi_i(\mathbf{x}, t), I)), \end{aligned} \quad (2)$$

where $g(\cdot) \geq 0$ represents an internal image energy of $\psi_i(\mathbf{x}, t)$ and $g^c(\psi_i(\mathbf{x}, t), I) \geq 0$ is the image energy from all other $\psi_j(\mathbf{x}, t)$, $j \neq i$. A negative sign is used in front of equation (2) due to the definition of H for interior region. This



Control-Based Segmentation, Fig. 1 Diagram of the control-based segmentation framework. The feedback compensates for deficiencies in automatic segmentation by utilizing the expert's knowledge

decomposition technique has been used for modeling multiple active contours in different region-based algorithms (Vázquez et al. 2006). On the other hand, since distance information from given points of an image may be implemented using the level set formulation, segmentation algorithms that are based on clustering pixels according to the minimal distance to given seeds naturally fit into this formulation. In particular, the proposed framework is applicable to (1) *region-based active contour models* and (2) *distance-based clustering*.

Region-Based Active Contours

Let $g_r(\psi_i(\mathbf{x}, t), I)$ be a function of statistics for a region ψ_i . The image force $G(\psi_i(\mathbf{x}, t), I)$ can be rewritten as

$$G(\psi_i(\mathbf{x}, t), I) = -(g_r(\psi_i(\mathbf{x}, t), I) - g_r^c(\psi_i(\mathbf{x}, t), I)), \quad (3)$$

where $g_r^c(\psi_i(\mathbf{x}, t), I) = \min_{j \neq i} \{g_r(\psi_j(\mathbf{x}, t), I)\}$. A simple example is the first order statistics defined as

$$g_r(\psi_i(\mathbf{x}, t), I) = \int_{\Lambda} \|I(\mathbf{x}) - \mu_i(t)\|_2^2 H(\psi_i(\mathbf{x}, t)) d\mathbf{x}, \quad (4)$$

where $\mu_i(t)$ is the mean intensity of region Λ_i at time t (Chan and Vese 2001). Other region-based energies (Lankton and Tannenbaum 2008; Gao et al. 2012; Michailovich et al. 2007) may be employed similarly.

Distance-Based Clustering

Distance-based clustering methods assign labels to pixels based on the minimal distance to given seed points. Letting Λ_{x_i} denote the seed region for the i th label, the distance between any point $\mathbf{x}$ and the label is defined as

$$\mathbf{d}(\mathbf{x}, \Lambda_{x_i}) := \min_{y \in \Lambda_{x_i}} \min_{C \in \pi(\mathbf{x}, y)} \int_0^1 g_\gamma(C(p)) \|C'(p)\| dp, \quad (5)$$

where $\pi(\mathbf{x}, y)$ is any path connecting points $\mathbf{x}$ and y and p is the parameterization of the path with metric g_γ . This formulation may be interpreted as a front propagation with an image-dependent speed function $1/g_\gamma$ and implemented by using the level set formulation. The front arriving earlier than any of the other fronts becomes the current front. The force on the current front of ψ_i may be written in the same form as in equation (2):

$$G(\psi_i(\mathbf{x}, t), I) = -(g_d(\psi_i(\mathbf{x}, t), I) - g_d^c(\psi_i(\mathbf{x}, t), I)), \quad (6)$$

with

$$g_d(\psi_i(\mathbf{x}, t), I) = g_\gamma(I)H(\psi_i^{-1}(\mathbf{x}, t) - \min_{j=1, \dots, N} \{\psi_j^{-1}(\mathbf{x}, t)\}) \quad (7)$$

and $g_d^c = g_\gamma(I)$, where $\psi_i^{-1}(\mathbf{x}, t)$ denotes the distance value at point $\mathbf{x}$ and H is the Heaviside function. An example of the function g_γ is based on image gradient information and defined as $g_\gamma(I) = 1 + \|\nabla I\|_2^2$.

Feedback Control and Interactive Segmentation

Suppose the user has an ideal segmentation $\{\psi_i^*(\mathbf{x})\}, i = 1, \dots, N$ in mind. Given this, our goal is to design a feedback control system

$$\begin{aligned} \frac{\partial \psi_i(\mathbf{x}, t)}{\partial t} &= (G(\psi_i(\mathbf{x}, t), I) \\ &\quad + F(\psi_i(\mathbf{x}, t), \psi_i^*(\mathbf{x})))\delta(\psi_i(\mathbf{x}, t)), \\ \psi_i(\mathbf{x}, 0) &= \psi_i^0(\mathbf{x}) \end{aligned}$$

(8) for given constants $v > 0, \mu > 0$.

such that $\lim_{t \rightarrow \infty} \psi_i(\mathbf{x}, t) \rightarrow \psi_i^*(\mathbf{x})$ for $i = 1, \dots, N$, where $F(\psi_i(\mathbf{x}, t), \psi_i^*(\mathbf{x}))$ is the control signal to be determined.

Regulatory Control

Define point-wise error for each region $\xi_i(\mathbf{x}, t)$ and total error $V(\xi, t)$ as

$$\begin{aligned} \xi_i(\mathbf{x}, t) &= \varepsilon(\psi_i(\mathbf{x}, t), \psi_i^*(\mathbf{x})) \\ &\triangleq H(\psi_i(\mathbf{x}, t)) - H(\psi_i^*(\mathbf{x})) \end{aligned} \quad (9)$$

$$V(\xi, t) = \frac{1}{2} \sum_{i=1}^N \int_{\Lambda} \xi_i^2(\mathbf{x}, t) d\mathbf{x}, \quad (10)$$

where $\xi_i(\mathbf{x}, t) \in \{-1, 0, 1\}$ measures the point-wise difference between ψ_i and ψ_i^* at time t . If

ψ^* is given and $V(\xi, t) \in C^1$ with respect to t , the determination of $F(\cdot, \cdot)$ is straightforward by applying the Lyapunov's direct method for stabilization. The control signal uses the bounds of the image-dependent term $g(\cdot, \cdot)$ defined in equation (2):

$$g_M(\mathbf{x}) = \max_{i=1, \dots, N} \{g(\psi_i(\mathbf{x}), I), g^c(\psi_i(\mathbf{x}), I)\} \quad (11)$$

For the dynamical system defined in equation (8), we have the following theorem (Zhu et al. 2018):

Theorem 1 *The control law*

$$F(\psi_i(\mathbf{x}, t), \psi_i^*(\mathbf{x})) = -\alpha_i^2(\mathbf{x}, t)\xi_i(\mathbf{x}, t), \quad (12)$$

where $\alpha_i^2(\mathbf{x}, t) \geq g_M(\mathbf{x})$, asymptotically stabilizes the system (8) from $\{\psi_i(\mathbf{x}, t)\}$ to $\{\psi_i^*(\mathbf{x})\}, i = 1, \dots, N$.

Furthermore, the control law exponentially stabilizes the system with a convergence rate of e^{-vt} when ξ is large in the sense of

$$\begin{aligned} \mu \int_{\Lambda} \delta^2(\psi_i(\mathbf{x}, t))\xi_i^2(\mathbf{x}, t) d\mathbf{x} \\ \leq \int_{\Lambda} \xi_i^2(\mathbf{x}, t) d\mathbf{x}, i = 1, \dots, N \end{aligned} \quad (13)$$

for given constants $v > 0, \mu > 0$.

Examples of Feedback Control-Based Segmentation Methods

A large number of evolutionary methods can be augmented to include user interactions using the proposed design principles. Two representative methods are discussed in this section: (1) classic region-based active contour models and (2) distance-based clustering.

Controlled Region-Based Active Contour Models

The image force $G(\psi_i(\mathbf{x}, t), I)$ described in Sect. 1 is used. Our goal is then how to design the user input u_i^k and U_i based on principles from feedback control theory.

A kernel-based method is employed to model the user input for u_i^k as

$$u_i^k(\mathbf{x}, t_i^{k+}) = h_0(\|\mathbf{x} - \Lambda_{x_i^k}\|_g), \quad (14)$$

where $\|\mathbf{x} - \Lambda_{x_i^k}\|_g$ is the geodesic distance from $\mathbf{x}$ to $\Lambda_{x_i^k}$ in terms of an image based metric g and h_0 is a decreasing function with respect to this distance. Here, we employ $g = 1 + \|\nabla I\|_2^2$ and $h_0 = (d_{\max} - d)/(d_{\max} - d_{\min})$, where $d, d_{\max}, d_{\min}$ are the distance and its corresponding extrema, respectively. The value of $d_{\max}$ controls the maximal range the current input affects. The overall effect for label i is defined as

$$\mathbf{u}_i(\mathbf{x}, t) = \sum_k u_i^k(\mathbf{x}_i^k, t), \quad (15)$$

To propagate and smooth the effect of user input, a diffusion process is applied to $\mathbf{u}_i(\mathbf{x}, t)$ as in Karasev et al. (2013), where

$$\begin{cases} \frac{\partial \mathbf{u}_i}{\partial t} = \mathbf{u}_i + \nabla \cdot [H((\mathbf{u}_i/g_M)^2 - 1)\nabla \mathbf{u}_i] \\ \mathbf{u}_i(\mathbf{x}, 0) = 0 \end{cases} \quad (16)$$

The total user input effect is defined as

$$\begin{aligned} U_i(\mathbf{x}, t) &= J(\mathbf{u}_i(\mathbf{x}, t), \mathbf{u}_i^c(\mathbf{x}, t)) \\ &\triangleq \mathbf{u}_i(\mathbf{x}, t) - \sum_{j \neq i} \mathbf{u}_j(\mathbf{x}, t) \end{aligned} \quad (17)$$

The label error is approximated by

$$\hat{\xi}_i(\mathbf{x}, t) = S(U_i(\mathbf{x}, t)) \triangleq -\text{sign}(U_i(\mathbf{x}, t)). \quad (18)$$

Since we are only certain about the error at where the user scribbled, the strength of the control law is defined in a weighted sense as

$$F(\psi_i(\mathbf{x}, t), \psi_i^*(\mathbf{x})) = -g_M |U_i| \hat{\xi}_i(\mathbf{x}, t). \quad (19)$$

The purpose of weighting is that the user effect decreases as a function of distance to the actual input points. When this distance is large, we

have $g_M |U_i| \ll g_M$. Therefore, the control law drives the system to a balanced location between the force from G and F when the distance is large.

Controlled Distance-Based Clustering

Here we use the simple image force $G(\psi_i(\mathbf{x}, t), I)$ as described in Sect. 1. The control scheme described in the previous section may in principle be extended to the distance-based clustering, but it causes inconsistency between metrics used in the system equation and user input. To resolve this inconsistency, the effect of user input is defined to have the same metric as the system equations

$$\begin{aligned} u_i^k(\mathbf{x}, t) &= \min_{\mathbf{y} \in \Lambda_{x_i^k}} \min_{C \in \theta(\mathbf{x}, \mathbf{y})} \int_0^1 g_\gamma(C(l)) \|C'(l)\| dl \\ u_i^k(\Lambda_{x_i^k}, t_i^k) &= q, \quad u_i^k(\Lambda_{x_i^k}^c, t_i^k) = p \end{aligned} \quad (20)$$

with $q = 0$ and $p = \infty$, where $\theta(\mathbf{x}, \mathbf{y})$ is any path connecting points $\mathbf{x}$ and $\mathbf{y}$ and l is the parameterization of the path with metric g_γ .

The total user input effect is computed as

$$\begin{aligned} U_i(\mathbf{x}, t) &= J(\mathbf{u}_i(\mathbf{x}, t), \mathbf{u}_i^c(\mathbf{x}, t)) \\ &\triangleq \min_{u_k \in \mathbf{u}_i(\mathbf{x}, t)} u_k(\mathbf{x}, t) - \min_{u_k \in \mathbf{u}_i^c(\mathbf{x}, t)} u_k(\mathbf{x}, t). \end{aligned} \quad (21)$$

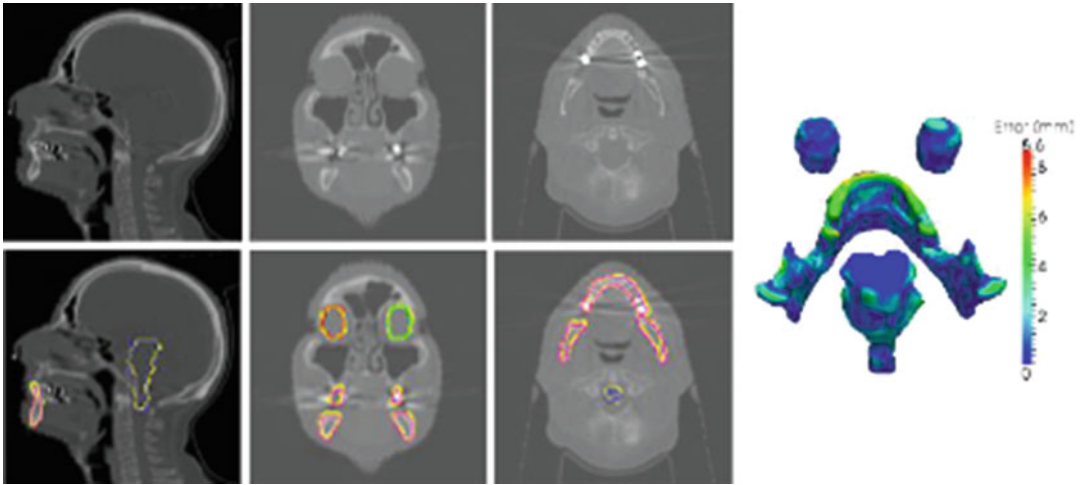
The effect of user input is propagated as $u_i^k(\mathbf{x}, t)$ evolving. The error estimate is defined as

$$\hat{\xi}_i(\mathbf{x}, t) = S(U_i(\mathbf{x}, t)) \triangleq \text{sign}(U_i(\mathbf{x}, t)). \quad (22)$$

The control law at continuous interval is defined as

$$F(\psi_i(\mathbf{x}, t), \psi_i^*(\mathbf{x})) = -g_M \hat{\xi}_i(\mathbf{x}, t). \quad (23)$$

For each $\mathbf{x}$ at the front $\delta(\psi_i(\mathbf{x}), t)$ with $\hat{\xi}_i(\mathbf{x}, t) = 0$, its distance value $\psi_i^{-1}(\mathbf{x}, t)$ is updated by the $\min\{\mathbf{u}_i(\mathbf{x}, t)\}$ if its distance is smaller. Thus, all distance values are up-to-date when there is no estimated segmentation error.



Control-Based Segmentation, Fig. 2 Region-based feedback control method to segment the left eye (red), right eye (green), brain stem (blue), and mandible (pink),

superimposed over manual segmentations (yellow), in axial, coronal, and sagittal views, respectively. The distribution of errors is shown on the right

We give an example of the region-based active contour control segmentation algorithm in Fig. 2.

Conclusion

In this work, we summarized the work of Karasev et al. (2013) and Zhu et al. (2018) for interactive (steerable) closed-loop control formulations for performing the central task of segmentation. This framework allows for a number of design choices; in particular, prior information can be incorporated in the estimators, and the automatic algorithms are replaceable, the control functions can be designed, and the user interactions can change. We should note that open source software is available for the implementation of the algorithms in 3D Slicer www.kitware.com that may be used in just about any platform.

Cross-References

- [H₂ Optimal Control](#)
- [Nonlinear Filters](#)

Acknowledgments This work was supported by AFOSR grant (FA9550-17-1-0435), a grant from National Institutes of Health (R01-AG048769), and a grant from Breast Cancer Research Foundation (BCRF-17-193).

Bibliography

- Boykov Y, Funka-Lea G (2006) Graph cuts and efficient n-d image segmentation. *Int J Comput Vis* 70(2): 109–131
- Chan T, Vese L (2001) Active contours without edges. *Trans Image Process* 10(2):266–277
- Gao Y, Kikinis R, Bouix S, Shenton M, Tannenbaum A (2012) A 3D interactive multi-object segmentation tool using local robust statistics driven active contours. *Med Image Anal* 16(6):1216–1227
- Karasev P, Kolesov I, Fritscher K, Vela P, Mitchell P, Tannenbaum A (2013) Interactive medical image segmentation using pde control of active contours. *IEEE Trans Med Imaging* 32(11):2127–2139
- Lankton S, Tannenbaum A (2008) Localizing region-based active contours. *IEEE Trans Image Process* 17(11):2029–2039
- Michailovich O, Rathi Y, Tannenbaum A (2007) Image segmentation using active contours driven by the Bhat-tacharyya gradient flow. *IEEE Trans Image Process* 16(11):2787–2801
- Osher S, Fedkiw R (2003) *Level set methods and dynamic implicit surfaces*, vol 153. Springer, New York
- Vázquez C, Mitiche A, Laganière R (2006) Joint multi-region segmentation and parametric estimation of image motion by basis function representation and level set evolution. *IEEE Trans Pattern Anal Mach Intell* 28(5):782–793
- Vezhnevets V, Konouchine V (2005) “Growcut” – interactive multi-label N-D image segmentation by cellular automata. In: *Proceedings of graphicon*, pp 150–156
- Zhu L, Kolesov I, Karasev P, Sandhu R, Tannenbaum A (2018) Guiding image segmentation on the fly: interactive segmentation from a feedback control perspective. *IEEE Trans Autom Control* 99. <https://doi.org/10.1109/TAC.2018.2792328>

Controllability and Observability

H. L. Trentelman

Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, Groningen, AV, The Netherlands

Abstract

State controllability and observability are key properties in linear input–output systems in state-space form. In the state-space approach, the relation between inputs and outputs is represented using the state variables of the system. A natural question is then to what extent it is possible to manipulate the values of the state vector by means of an appropriate choice of the input function. The concepts of controllability, reachability, and null controllability address this issue. Another important question is whether it is possible to uniquely determine the values of the state vector from knowledge of the input and output signals over a given time interval. This question is dealt with using the concept of observability.

Keywords

Controllability · Duality · Indistinguishability · Input–output systems in state-space form · Observability · Reachability

Introduction

In the state-space approach to input–output systems, the relation between input signals and output signals is represented by means of two equations. In the continuous-time case, the first of these equations is a first-order vector differential equation driven by the input signal and is often called *the state equation*. The second equation is an algebraic equation, often called the *output equation*. The unknown in the differential equation is called the *state vector* of the system. Given a particular input signal and initial value of the state vector, the state equation generates

a unique solution, called the state trajectory of the system. The output equation determines the corresponding output signal as a function of this state trajectory and the input signal. Thus, in the state space approach, the input–output behavior of the system is obtained using the state vector as an intermediate variable.

In the context of input–output systems in state-space form, the properties of controllability and observability characterize the interaction between the input, the state, and the output. In particular, controllability describes the ability to manipulate the state vector of the system by applying appropriate input signals. Observability describes the ability to determine the values of the state vector from knowledge of the input and output over a certain time interval. The properties of controllability and observability are fundamental properties that play a major role in the analysis and control of linear input–output systems in state-space form.

Systems with Inputs and Outputs

Consider a continuous-time, linear, time-invariant, input–output system in state-space form represented by the equations

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t).\end{aligned}\tag{1}$$

This system is referred to as Σ . In Eq. (1), A , B , C , and D are maps (or matrices), and the functions x , u , and y are considered to be defined on the real axis $\mathbb{R}$ or on any subinterval of it. In particular, one often assumes the domain of definition to be the nonnegative part of $\mathbb{R}$, which is without loss of generality since the system is time-invariant. The function u is called the *input*, and its values are assumed to be given. The class of admissible input functions is denoted by $\mathbf{U}$. Often, $\mathbf{U}$ is the class of piecewise continuous or locally integrable functions, but for most purposes, the exact class from which the input functions are chosen is not important. We assume that input functions take values in an m -dimensional space $\mathcal{U}$, which we often identify with $\mathbb{R}^m$. The first equation of Σ is an ordinary differential

equation for the variable x . For a given initial value of x and input function u , the function x is completely determined by this equation. The variable x is called the *state variable* and it is assumed to take values in an n -dimensional space $\mathcal{X}$. The space $\mathcal{X}$ is called the *state space*. It is usually identified with $\mathbb{R}^n$. Finally, y is called the *output* of the system and takes values in a p -dimensional space $\mathcal{Y}$, which we identify with $\mathbb{R}^p$. Since the system Σ is completely determined by the maps (or matrices) A , B , C , and D , we identify Σ with the quadruple (A, B, C, D) .

The solution of the differential equation of Σ with initial value $x(0) = x_0$ is denoted as $x_u(t, x_0)$. It can be given explicitly using the variation-of-constants formula, namely,

$$x_u(t, x_0) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau) d\tau. \quad (2)$$

The corresponding value of y is denoted by $y_u(t, x_0)$. As a consequence of (2), we have

$$y_u(t, x_0) = Ce^{At}x_0 + \int_0^t K(t-\tau)u(\tau) d\tau + Du(t), \quad (3)$$

where $K(t) := Ce^{At}B$. In the case $D = 0$, it is customary to call $K(t)$ the *impulse response*. In the general case, one would call the distribution $K(t) + D\delta(t)$ the impulse response.

Controllability

Controllability is concerned with the ability to manipulate the state by choosing an appropriate input signal, thus steering the current state to a desired future state in a given finite time. Thus, in particular, in the differential equation in (1), we study the relation between u and x . We investigate to what extent one can influence the state x by a suitable choice of the input u .

For this purpose, we introduce the (at time T) *reachable space* $\mathcal{W}_T$, defined as the space of points x_1 for which there exists an input u such that $x_u(T, 0) = x_1$, i.e., the set of points that can be reached from the origin at time T . It follows

from the linearity of the differential equation that $\mathcal{W}_T$ is a linear subspace of $\mathcal{X}$. In fact, (2) implies

$$\mathcal{W}_T = \left\{ \int_0^T e^{A(T-\tau)}Bu(\tau) d\tau \mid u \in \mathbf{U} \right\}. \quad (4)$$

We call system Σ *reachable at time T* if every point can be reached from the origin, i.e., if $\mathcal{W}_T = \mathcal{X}$. It follows from (2) that if the system is reachable at time T , every point can be reached from every point at time T , because the condition for the point x_1 to be reachable from x_0 at time T is

$$x_1 - e^{AT}x_0 \in \mathcal{W}_T.$$

The property that every point is reachable from any point in a given time interval $[0, T]$ is called *controllability (at T)*. Finally, we have the concept of *null controllability*, i.e., the possibility to reach the origin from an arbitrary initial point. According to (2), for a point x_0 to be null controllable at T , we must have

$$e^{AT}x_0 + \int_0^T e^{A(T-\tau)}Bu(\tau) d\tau = 0$$

for some $u \in \mathbf{U}$. We observe that x_0 is null controllable at T (by the control u) if and only if $-e^{AT}x_0$ is reachable at T (by the control u). Since e^{AT} is invertible, we see that Σ is null controllable at T if and only if Σ is reachable at T . Henceforth, we refer to the equivalent properties reachability, controllability, null controllability simply as controllability (at T). It should be remarked that the equivalence of these concepts does not hold in other situations, e.g., for discrete-time systems. We intend to obtain an explicit expression for the space $\mathcal{W}_T$ and, based on this, an explicit condition for controllability. This is provided by the following result.

Theorem 1 *Let η be an n -dimensional row vector and $T > 0$. Then the following statements are equivalent:*

1. $\eta \perp \mathcal{W}_T$ (i.e., $\eta x = 0$ for all $x \in \mathcal{W}_T$).
2. $\eta e^{tA}B = 0$ for $0 \leq t \leq T$.
3. $\eta A^k B = 0$ for $k = 0, 1, 2, \dots$
4. $\eta (B \ AB \ \dots \ A^{n-1}B) = 0$.

Proof.

- (i) $\Leftrightarrow$ (ii) If $\eta \perp \mathcal{W}_T$, then by Eq. (4):

$$\int_0^T \eta e^{A(T-\tau)} B u(\tau) d\tau = 0 \quad (5)$$

for every $u \in \mathbf{U}$. Choosing $u(t) = B^T e^{A^T(T-t)} \eta^T$ for $0 \leq t \leq T$ yields

$$\int_0^T \left\| \eta e^{A(T-\tau)} B \right\|^2 d\tau = 0,$$

from which (ii) follows. Conversely, assume that (ii) holds. Then (5) holds and hence (i) follows.

- (ii) $\Leftrightarrow$ (iii) This is obtained by power series expansion of $e^{At} \left(= \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k \right)$.
 (iii) $\Leftrightarrow$ (iv) This follows immediately from the evaluation of the vector-matrix product.
 (iv) $\Leftrightarrow$ (iii) This implication is based on the Cayley-Hamilton Theorem. According to this theorem, A^n is a linear combination of $I, A, \dots, A^{n-1}$. By induction, it follows that A^k ($k > n$) is a linear combination of $I, A, \dots, A^{n-1}$ as well. Therefore, $\eta A^k B = 0$ for $k = 0, 1, \dots, n-1$ implies that $\eta A^k B = 0$ for all $k \in \mathbb{N}$. $\square$

As an immediate consequence of the previous theorem, we find that at time T reachable subspace $\mathcal{W}_T$ can be expressed in terms of the maps A and B as follows. $\square$

Corollary 1

$$\mathcal{W}_T = \text{im} (B \quad AB \quad \dots \quad A^{n-1}B).$$

This implies that, in fact, $\mathcal{W}_T$ is independent of T , for $T > 0$. Because of this, we often use $\mathcal{W}$ instead of $\mathcal{W}_T$ and call this subspace the reachable subspace of Σ . This subspace of the state space has the following geometric characterization in terms of the maps A and B .

Corollary 2 $\mathcal{W}$ is the smallest A -invariant subspace containing $B := \text{im} B$. Explicitly, $\mathcal{W}$ is A -invariant, $B \subset \mathcal{W}$, and any A -invariant sub-

space $\mathcal{L}$ satisfying $B \subset \mathcal{L}$ also satisfies $\mathcal{W} \subset \mathcal{L}$. We denote the smallest A -invariant subspace containing B by $\langle A|B \rangle$, so that we can write $\mathcal{W} = \langle A|B \rangle$. For the space $\langle A|B \rangle$, we have the following explicit formula

$$\langle A|B \rangle = B + AB + \dots + A^{n-1}B.$$

Corollary 3 The following statements are equivalent.

1. There exists $T > 0$ such that system Σ is controllable at T .
2. $\langle A|B \rangle = \mathcal{X}$.
3. $\text{Rank} (B \quad AB \quad \dots \quad A^{n-1}B) = n$.
4. The system Σ is controllable at T for all $T > 0$.

We say that the matrix pair (A, B) is *controllable* if one of these equivalent conditions is satisfied.

Example 1 Let A and B be defined by

$$A := \begin{pmatrix} -2 & -6 \\ 2 & 5 \end{pmatrix}, \quad B := \begin{pmatrix} -3 \\ 2 \end{pmatrix}.$$

Then $(B \quad AB) = \begin{pmatrix} -3 & -6 \\ 2 & 4 \end{pmatrix}$, $\text{rank}(B \quad AB) = 1$, and consequently, (A, B) is not controllable. The reachable subspace is the span of $(B \quad AB)$, i.e., the line given by the equation $2x_1 + 3x_2 = 0$. This can also be seen as follows. Let $z := 2x_1 + 3x_2$, then $\dot{z} = z$. Hence, if $z(0) = 0$, which is the case if $x(0) = 0$, we must have $z(t) = 0$ for all $t \geq 0$.

Observability

In this section, we include the second of equations (1), $y = Cx + Du$, in our considerations. Specifically, we investigate to what extent it is possible to reconstruct the state x if the input u and the output y are known. The motivation is that we often can measure the output and prescribe (and hence know) the input, whereas the state variable is *hidden*.

Definition 1 Two states x_0 and x_1 in $\mathcal{X}$ are called *indistinguishable* on the interval $[0, T]$ if for any

input u we have $y_u(t, x_0) = y_u(t, x_1)$, for all $0 \leq t \leq T$.

Hence, x_0 and x_1 are indistinguishable if they give rise to the same output values for every input u . According to (3), for x_0 and x_1 to be indistinguishable on $[0, T]$, we must have that

$$\begin{aligned} Ce^{At}x_0 + \int_0^t K(t-\tau)u(\tau)d\tau + Du(t) \\ = Ce^{At}x_1 + \int_0^t K(t-\tau)u(\tau)d\tau + Du(t) \end{aligned}$$

for $0 \leq t \leq T$ and for any input signal u . We note that the input signal does not affect distinguishability, i.e., if one u is able to distinguish between two states, then any input is. In fact, x_0 and x_1 are indistinguishable if and only if $Ce^{At}x_0 = Ce^{At}x_1$ ($0 \leq t \leq T$). Obviously, x_0 and x_1 are indistinguishable if and only if $v := x_0 - x_1$ and 0 are indistinguishable. By applying Theorem 1 with $\eta = v^T$ nonzero and transposing the equations, it follows that $Ce^{At}x_0 = Ce^{At}x_1$ ($0 \leq t \leq T$) if and only if $Ce^{At}v = 0$ ($0 \leq t \leq T$) and hence if and only if $CA^k v = 0$ ($k = 0, 1, 2, \dots$). The Cayley-Hamilton Theorem implies that we need to consider the first n terms only, i.e.,

$$\begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{pmatrix} v = 0. \quad (6)$$

As a consequence, the distinguishability of two vectors does not depend on T . The space of vectors v for which (6) holds is denoted $\langle \ker C|A \rangle$ and called the *unobservable subspace*. It is equivalently characterized as the intersection of the spaces $\ker CA^k$ for $k = 0, \dots, n-1$, i.e.,

$$\langle \ker C|A \rangle = \bigcap_{k=0}^{n-1} \ker CA^k.$$

Equivalently, $\langle \ker C|A \rangle$ is the largest A -invariant subspace contained in $\ker C$. Finally, another characterization is “ $v \in \langle \ker C|A \rangle$ if and only if

$y_0(t, v)$ is identically zero,” where the subscript “0” refers to the zero input.

Definition 2 System Σ is called *observable* if any two distinct states are not indistinguishable. The previous considerations immediately lead to the result.

Theorem 2 The following statements are equivalent.

1. The system Σ is observable.
2. Every nonzero state is not indistinguishable from the origin.
3. $\langle \ker C|A \rangle = 0$.
4. $Ce^{At}v = 0$ ($0 \leq t \leq T$) $\Rightarrow v = 0$.

$$5. \text{Rank} \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{pmatrix} = n.$$

Since observability is completely determined by the matrix pair (C, A) , we will say “ (C, A) is observable” instead of “system Σ is observable.”

There is a remarkable relation between the controllability and observability properties, which is referred to as *duality*. This property is most conspicuous from the conditions (3) in Corollary 3 and (5) in Theorem 2, respectively. Specifically, (C, A) is observable if and only if (A^T, C^T) is controllable. As a consequence of duality, many theorems on controllability can be translated into theorems on observability and vice versa by mere transposition of matrices.

Example 2 Let

$$A := \begin{pmatrix} -11 & 3 \\ -3 & -5 \end{pmatrix}, \quad B := \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

$$C := (1 \quad -1),$$

Then

$$\text{rank} \begin{pmatrix} C \\ CA \end{pmatrix} = \text{rank} \begin{pmatrix} 1 & -1 \\ -8 & 8 \end{pmatrix} = 1,$$

hence, (C, A) is not observable. Notice that if $v \in \langle \ker C|A \rangle$ and $u = 0$, identically, then $y = 0$, identically. In this example, $\langle \ker C|A \rangle$ is the span of $(1, 1)^T$.

Summary and Future Directions

The property of controllability can be tested by means of a rank test on a matrix involving the maps A and B appearing in the state equation of the system. Alternatively, controllability is equivalent to the property that the reachable subspace of the system is equal to the state space. The property of observability allows a rank test on a matrix involving the maps A and C appearing in the system equations. An alternative characterization of this property is that the unobservable subspace of the system is equal to the zero subspace. Concepts of controllability and observability have also been defined for discrete-time systems and, more generally, for time-varying systems and for continuous-time and discrete-time nonlinear systems.

Cross-References

- [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)
- [Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions](#)
- [Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions](#)
- [Linear Systems: Discrete-Time, Time-Varying, State Variable Descriptions](#)
- [Realizations in Linear Systems Theory](#)

Recommended Reading

The description of linear systems in terms of a state space representation was particularly stressed by R. E. Kalman in the early 1960s (see Kalman 1960a,b, 1963), Kalman et al. (1963). See also Zadeh and Desoer (1963) and Gilbert (1963). In particular, Kalman introduced the concepts of controllability and observability and gave the conditions expressed in Corollary 3, time (3), and Theorem 5, item (5). Alternative conditions for controllability and observability have been introduced in Hautus (1969) and independently by a number of authors; see Popov

(1966) and Popov (1973). Other references are Belevitch (1968) and Rosenbrock (1970).

Bibliography

- Antsaklis PJ, Michel AN (2007) *A linear systems primer*. Birkhäuser, Boston
- Belevitch V (1968) *Classical network theory*. Holden-Day, San Francisco
- Gilbert EG (1963) Controllability and observability in multivariable control systems. *J Soc Ind Appl Math A* 2:128–151
- Hautus MLJ (1969) Controllability and observability conditions of linear autonomous systems. *Proc Nederl Akad Wetensch A* 72(5):443–448
- Kalman RE (1960a) Contributions to the theory of optimal control. *Bol Soc Mat Mex* 2:102–119
- Kalman RE (1960b) On the general theory of control systems. In: *Proceedings of the first IFAC congress*, London: Butterworth, pp 481–491
- Kalman RE (1963) Mathematical description of linear dynamical systems. *J Soc Ind Appl Math A* 1:152–192
- Kalman RE, Ho YC, Narendra KS (1963) Controllability of linear dynamical systems. *Contrib Diff Equ* 1:189–213
- Popov VM (1966) *Hiperstabilitatea sistemelor automate*. Editura Axcademiei Republicii Socialiste România (in Rumanian)
- Popov VM (1973) *Hyperstability of control systems*. Springer, Berlin. (Translation of the previous reference)
- Rosenbrock HH (1970) *State-space and multivariable theory*. Wiley, New York
- Trentelman HL, Stoorvogel AA, Hautus MLJ (2001) *Control theory for linear systems*. Springer, London
- Wonham WM (1979) *Linear multivariable control: a geometric approach*. Springer, New York
- Zadeh LA, Desoer CA (1963) *Linear systems theory – the state-space approach*. McGraw-Hill, New York

Controllability of Network Systems

Fabio Pasqualetti
Department of Mechanical Engineering,
University of California at Riverside, Riverside,
CA, USA

Abstract

Several natural and man-made systems are often represented as complex networks, since

their dynamic behavior depends to a large extent upon the properties of their interconnection structure. Electrical power grids, mass transportation systems, and cellular networks are instances of modern technological networks, while metabolic and brain networks are biological examples. The ability to control and reconfigure complex networks via external controls is fundamental to guarantee reliable and efficient network functionalities. In this note, we review recent advances and methods that characterize fundamental controllability properties and limitations of network systems.

Keywords

Network · Controllability · Structural analysis · Optimization

Introduction

Network systems are dynamical systems that arise from the interconnection of simpler dynamical units. Such systems are typically described by a graph representing the interconnections among the different units, a state containing the characteristic values of all units, and a map describing the dynamics of the state. In a simple and commonly used setting, the interconnections among the units are captured by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ are the vertex and edge sets, respectively, the state of each node is a real number, the network state is $x \in \mathbb{R}^n$, and the state dynamics are captured by the linear, discrete time, noiseless, and time-invariant recursion

$$x(t+1) = Ax(t). \quad (1)$$

In (1), the matrix $A = [a_{ij}]$ is a weighted adjacency matrix of $\mathcal{G}$, where $a_{ij} = 0$ if $(i, j) \notin \mathcal{E}$ and a_{ij} equals an arbitrary real number if $(i, j) \in \mathcal{E}$. Further, it is often assumed that a subset of control nodes $\mathcal{K} = \{k_1, \dots, k_m\} \subseteq \mathcal{V}$ can be independently manipulated, which leads to the controlled network dynamics

$$x(t+1) = Ax(t) + B_{\mathcal{K}}u_{\mathcal{K}}(t), \quad (2)$$

where $B_{\mathcal{K}} = [e_{k_1} \cdots e_{k_m}]$, e_i is the i -th canonical vector of dimension n , and $u_{\mathcal{K}} : \mathbb{N}_{\geq 0} \rightarrow \mathbb{R}^m$ is the control signal injected in the network via the nodes $\mathcal{K}$.

The network (2) is controllable in T steps by the control nodes $\mathcal{K}$ if there exists an input $u_{\mathcal{K}}$ such that $x(0) = x_0$ and $x(T) = x_d$, for any initial state x_0 and final state x_d . Several equivalent conditions ensuring controllability (Kalman et al. (1963) and Kailath (1980)) are known. For instance, the network (2) is controllable in T steps by the control nodes $\mathcal{K}$ if and only if the T -steps controllability matrix

$$C_{\mathcal{K},T} = [B_{\mathcal{K}} \ AB_{\mathcal{K}} \ A^2B_{\mathcal{K}} \ \cdots \ A^{T-1}B_{\mathcal{K}}]$$

is full rank or, equivalently, if and only if the T -steps controllability Gramian

$$W_{\mathcal{K},T} = \sum_{\tau=0}^{T-1} A^{\tau} B_{\mathcal{K}} B_{\mathcal{K}}^{\top} (A^{\top})^{\tau} = C_{\mathcal{K},T} C_{\mathcal{K},T}^{\top}$$

is invertible. When the final time satisfies $T \geq n$, network controllability is also equivalent to the matrix $[\lambda I - A \ B_{\mathcal{K}}]$ being full rank for every eigenvalue λ of A . Finally, when A is (Schur) stable and the final time satisfies $T = \infty$, network controllability is ensured by the existence of a unique positive-definite solution X to the Lyapunov equation $AXA^{\top} - X + B_{\mathcal{K}}B_{\mathcal{K}}^{\top} = 0$ (in fact, $X = W_{\mathcal{K},\infty}$). It should be noticed that the above tests for network controllability are descriptive, in the sense that they allow to test controllability of a network by a set of control nodes, but not prescriptive because they do not indicate how to select control nodes, or how to modify the network structure and weights, to ensure controllability of network systems.

Results on the Controllability of Network Systems

In this section, we review important approaches and results for the network controllability prob-

lem. We divide the results into three main categories, which differ in the required knowledge of the network matrices, in the type of information that they can provide, and in their numerical complexity.

Structural Controllability of Network Systems

A convenient method to select a set of control nodes to guarantee controllability of the network system (2) originates from the theory of structural systems. To formalize this discussion, let $a_{\mathcal{E}} = \{a_{ij} : (i, j) \in \mathcal{E}\}_{\text{ordered}}$ denote the set of nonzero entries of A in lexicographic order, and notice that the determinant $\det(W_{\mathcal{K}, T}) = \phi(a_{\mathcal{E}})$ is a polynomial function with variables $a_{\mathcal{E}}$. Then, the network (2) is uncontrollable when the weights $a_{\mathcal{E}}$ are chosen so that $\phi(a_{\mathcal{E}}) = 0$. Let $\mathcal{S}$ contain the choices of weights that render the network (2) uncontrollable; that is,

$$\mathcal{S} = \{z \in \mathbb{R}^d : \phi(z_1, \dots, z_d) = 0\}, \quad (3)$$

where $d = |\mathcal{E}| = |a_{\mathcal{E}}|$. Notice that $\mathcal{S}$ describes an algebraic variety of $\mathbb{R}^d$ (Wonham 1985). This implies that controllability of (2) is a *generic property*, because it fails to hold on an algebraic variety of the parameter space (Markus et al. 1962; Wonham 1985; Tchoń 1983). Consequently, when assessing controllability of the network (2) as a function of the weights, only two mutually exclusive cases are possible:

1. either there is *no* choice of weights a_{ij} , with $a_{ij} = 0$ if $(i, j) \notin \mathcal{E}$, rendering the network (2) controllable; or
2. the network (2) is controllable for all choices of weights a_{ij} , with $a_{ij} = 0$ if $(i, j) \notin \mathcal{E}$, except those belonging to the proper algebraic variety $\mathcal{S} \subset \mathbb{R}^d$. (The variety $\mathcal{S}$ of $\mathbb{R}^d$ is proper when $\mathcal{S} \neq \mathbb{R}^d$ (Wonham 1985).)

Loosely speaking, if one can find a choice of weights such that the network (2) is controllable, then *almost all* choices of weights yield a controllable network. In this case, the network is said to be *structurally controllable* (Davison and Wang 1973; Lin 1974; Reinschke 1988).

Following the above reasoning, necessary and sufficient graph-theoretic conditions on the network graph $\mathcal{G}$ and the set of control nodes $\mathcal{K}$ have been identified to guarantee controllability of network systems, as well as algorithms for the selection of control sets with minimum cardinality. For instance, Reinschke (1988) finds network controllability conditions using the notions of *cycle family* and *rooted path*, while the seminal work of Lin (1974) employs the notion of spanning *cacti* and Murota (2012) uses bipartite-graph criteria. Algorithms, and their complexity, to determine minimal sets of control nodes for structural controllability are presented, among others, in Olshevsky (2014) and Pequito et al. (2015). The case of strong structural controllability, where controllability must be satisfied for any nonzero choice of the network weights, is studied, for instance, in Mayeda and Yamada (1979), Menara et al. (2017), Jarczyk et al. (2011), and Jia et al. (2019), while the case of constrained entries is discussed in Menara et al. (2019), Mousavi et al. (2017), and Li et al. (2018). We conclude noting that structural controllability methods require the knowledge of only the network graph $\mathcal{G}$ and can only provide generic results that are independent of the network weights. Thus, even if a network system is structurally controllable, the actual numerical realization (i.e., the choice of network weights) may not be controllable.

Model-Based Controllability of Network Systems

The notion of structural controllability is qualitative, in the sense that it is independent of the network weights and unable to capture any property of the input signal that steers the network system to the desired state. As a matter of fact, most connected network systems are controllable even with a single control node (Reinschke 1988; Menara et al. 2019; Gu et al. 2015), although the properties of their control inputs may vary significantly as a function of the network size, weights, and number of control nodes. To quantify a degree of controllability, one convenient measure is the *energy* of the control input to

transfer the network state from $x_0 = 0$ to a desired state x_d . Specifically, define the energy of the control input $u_{\mathcal{K}}$ as

$$E(u_{\mathcal{K}}, T) = \sum_{\tau=0}^{T-1} \|u_{\mathcal{K}}(\tau)\|_2^2,$$

where T is the control horizon, and recall (Kailath 1980) that the unique control input that steers the network from $x(0) = 0$ to $x(T) = x_d$ with minimum energy is

$$u_{\mathcal{K}}^*(t) = B_{\mathcal{K}}^T (A^T)^{T-t-1} W_{\mathcal{K},T}^{-1} x_d. \quad (4)$$

Moreover, the energy of (4) satisfies $E(u_{\mathcal{K}}^*, T) = x_d^T W_{\mathcal{K},T}^{-1} x_d$. Different measures of the controllability degree of network systems can be obtained using the notion of control energy. For instance, the worst-case input energy is (In what follows, $\lambda_{\min}(M)$, $\text{Trace}(M)$, and $\text{Det}(M)$ denote the smallest eigenvalue, the trace, and the determinant of the square matrix M , respectively.)

$$E_{\max} = \max_{\|x_d\|=1} x_d^T W_{\mathcal{K},T}^{-1} x_d = \lambda_{\min}^{-1}(W_{\mathcal{K},T}), \quad (5)$$

while the average input energy (over all target states with unit norm) can be computed to be $E_{\text{average}} = \text{Trace}(W_{\mathcal{K},T}^{-1})$. The metrics $E_{\text{volume}} = \text{Det}(W_{\mathcal{K},T})$, which is proportional to the volume of the ellipsoid containing the states that can be reached with a unit-energy control input, and $E_{\text{impulse}} = \text{Trace}(W_{\mathcal{K},T})$, which measures the energy of the network response to a unit impulsive input, have also been adopted to measure the controllability degree of a network system (Müller and Weber 1972; Marx et al. 2004; Shaker and Tahavori 2012). The selection of control nodes for the optimization of these metrics is usually a computationally hard combinatorial problem (Olshevsky 2017, 2013), for which heuristics, typically without performance guarantees, and non-scalable optimization procedures have been proposed (Summers and Lygeros 2014;

Van De Wal and De Jager 2001; Sharon and Rex 1999).

Recent studies have demonstrated connections between the control energy of a network and its structure and parameters. For instance, it has been shown that most complex networks cannot be controlled by few nodes, because the control energy grows exponentially with the network cardinality (Pasqualetti et al. 2014). This property ensures that existing complex structures are in fact robust to targeted perturbations or failures. On the other hand, there exist topologies that violate this paradigm, where few controllers can arbitrarily reprogram large structures with little effort (Pasqualetti and Zampieri 2014; Zhao and Pasqualetti 2019, 2018). Control energy also grows significantly when the eigenvalues of the network matrix are clustered in a region of the complex plane (Olshevsky et al. 2016), when the diameter of the network grows linearly with its size (Bianchin et al. 2015), and also for a large class of networks with continuous-time dynamics (Zhao and Pasqualetti 2017). Finally, Bof et al. (2017) highlight an interesting connection between the controllability degree of a network and popular centrality measures.

Data-Driven Controllability of Network Systems

While the techniques described above require accurate knowledge of the network structure and, possibly, its weights, a recent line of research aims to assess and evaluate network controllability in a model-free and data-driven fashion. To this aim, the availability of a set of N control experiments is assumed, where the i -th control experiment consists of applying a randomly chosen input sequence $u_{\mathcal{K},i}$ to (2), and measuring the system state at time T , namely, $x_i = A^T x_0 + C_{\mathcal{K},T} u_{\mathcal{K},i}$. Thus, in vector form, the available data is assumed to be

$$X = [x_1 \cdots x_N] \quad \text{and} \quad U = [u_{\mathcal{K},1} \cdots u_{\mathcal{K},N}]. \quad (6)$$

In this setting, if the input matrix U is of full row rank (Baggio et al. 2019), then the minimum-

energy input (4) can equivalently be computed (in vector form) as

$$u_K^* = (I - UK(UK)^\dagger)UX^\dagger x_d = (XU^\dagger)^\dagger x_f \quad (7)$$

where $K = \text{Ker}(X)$ denotes the null space of the matrix X . Data-driven expressions of the controllability Gramian can be derived directly from (7), or following the approach described in Baggio et al. (2019). The data-driven approach to study network controllability problems, which has already been extended to different control settings (Baggio and Pasqualetti 2020; Coulson et al. 2019; Tabuada et al. 2017; Persis and Tesi 2020; Berberich and Allgöwer 2019; van Waarde et al. 2019), may provide solutions to the shortcomings of model-based techniques to study controllability of large networks (Sun and Motter 2013; Antoulas et al. 2002).

Summary and Future Directions

Network controllability is an active field of research with broad implications for natural, social, and technological systems (Gu et al. 2015; Acemoglu et al. 2010; Rajapakse et al. 2011; Skardal and Arenas 2015; Rahmani et al. 2009). Different methods and measures have been proposed and are currently being developed. Potentially interesting research directions include the study of controllability for networks with time-varying topologies, as well as a more flexible and dynamic choice of the control nodes, which could in principle overcome the energy limitations of a fixed selection of control nodes in large systems. The problem of defining controllability metrics and algorithms for closed-loop network systems is also fundamentally important, as it would allow us to design and operate network systems robustly against noise and modeling uncertainties. The design of efficient model-based and data-driven node selection algorithms is an important outstanding aspect, as the numerical limitations and assumptions of existing methods prevent their applicability to large systems. Finally, the development

of a controllability theory for networks with nonlinear dynamics is also critically lacking, and it remains necessary to ensure the applicability of this field of research to real-world network systems. To conclude, we stress that this note reflects specific advances and is not meant to be comprehensive.

Cross-References

- [Controllability and Observability](#)
- [Estimation and Control over Networks](#)

Bibliography

- Acemoglu D, Ozdaglar A, ParandehGheibi A (2010) Spread of (mis)information in social networks. *Games Econ Behav* 70(2):194–227
- Antoulas AC, Sorensen DC, Zhou Y (2002) On the decay rate of Hankel singular values and related issues. *Syst Control Lett* 46(5):323–342
- Baggio G, Pasqualetti F (2020) Learning minimum-energy controls from heterogeneous data. In: *American control conference*, Denver
- Baggio G, Katewa V, Pasqualetti F (2019) Data-driven minimum-energy controls for linear systems. *IEEE Control Syst Lett* 3(3):589–594
- Berberich J, Allgöwer F (2019) A trajectory-based framework for data-driven system analysis and control. *arXiv preprint arXiv:1903.10723*
- Bianchin G, Pasqualetti F, Zampieri S (2015) The role of diameter in the controllability of complex networks. In: *IEEE conference on decision and control*, Osaka, pp 980–985
- Bof N, Baggio G, Zampieri S (2017) On the role of network centrality in the controllability of complex networks. *IEEE Trans Control Netw Syst* 4(3):643–653
- Coulson J, Lygeros J, Dörfler F (2019) Data-enabled predictive control: in the shallows of the DeePC. In: *2019 18th European control conference (ECC)*, pp 307–312
- Davison E, Wang S (1973) Properties of linear time-invariant multivariable systems subject to arbitrary output and state feedback. *IEEE Trans Autom Control* 18(1):24–32
- Gu S, Pasqualetti F, Cieslak M, Telesford QK, Alfred BY, Kahn AE, Medaglia JD, Vettel JM, Miller MB, Grafton ST, Bassett DS (2015) Controllability of structural brain networks. *Nat Commun* 6, 8414
- Jarczyk JC, Svaricek F, Alt B (2011) Strong structural controllability of linear systems revisited. In: *IEEE conference on decision and control and European control conference*, Orlando, pp 1213–1218

- Jia J, van Waarde HJ, Trentelman HL, Camlibel KM (2019) A unifying framework for strong structural controllability. *arXiv preprint arXiv:1903.03353*
- Kailath T (1980) *Linear systems*. Prentice-Hall
- Kalman RE, Ho YC, Narendra SK (1963) Controllability of linear dynamical systems. *Contrib Diff Equ* 1(2):189–213
- Li J, Chen X, Pequito S, Pappas GJ, Preciado VM (2018) Structural target controllability of undirected networks. In: *IEEE conference on decision and control*, Miami, pp 6656–6661
- Lin CT (1974) Structural controllability. *IEEE Trans Autom Control* 19(3):201–208
- Markus L, Lee EB (1962) On the existence of optimal controls. *J Basic Eng* 84(1):13–20
- Marx B, Koenig D, Georges D (2004) Optimal sensor and actuator location for descriptor systems using generalized Gramians and balanced realizations. In: *American control conference*, Boston, pp 2729–2734
- Mayeda H, Yamada T (1979) Strong structural controllability. *SIAM J Control Optim* 17(1):123–138
- Menara T, Bianchin G, Innocenti M, Pasqualetti F (2017) On the number of strongly structurally controllable networks. In: *American control conference*, Seattle, pp 340–345
- Menara T, Bassett DS, Pasqualetti F (2019) Structural controllability of symmetric networks. *IEEE Trans Autom Control* 64(9):3740–3747
- Mousavi SS, Haeri M, Mesbahi M (2017) On the structural and strong structural controllability of undirected networks. *IEEE Trans Autom Control* 63(7):2234–2241
- Müller PC, Weber HI (1972) Analysis and optimization of certain qualities of controllability and observability for linear dynamical systems. *Automatica* 8(3):237–246
- Murota K (2012) *Systems analysis by graphs and matroids: structural solvability and controllability*, vol 3. Springer Science & Business Media
- Olshevsky A (2013) The minimal controllability problem. *ArXiv preprint arXiv:1304.3071*
- Olshevsky A (2014) Minimal controllability problems. *IEEE Trans Control Netw Syst* 1(3):249–258
- Olshevsky A (2016) Eigenvalue clustering, control energy, and logarithmic capacity. *Syst Control Lett* 96(1):45–50
- Olshevsky A (2017) On (non) supermodularity of average control energy. *IEEE Trans Control Netw Syst* 5(3):1177–1181
- Pasqualetti F, Zampieri S (2014) On the controllability of isotropic and anisotropic networks. In: *IEEE conference on decision and control*, Los Angeles, pp 607–612
- Pasqualetti F, Zampieri S, Bullo F (2014) Controllability metrics, limitations and algorithms for complex networks. *IEEE Trans Control Netw Syst* 1(1):40–52
- Pequito S, Kar S, Aguiar PA (2015) A framework for structural input/output and control configuration selection in large-scale systems. *IEEE Trans Autom Control* 61(2):303–318
- Persis CD, Tesi P (2020) Formulas for data-driven control: stabilization, optimality and robustness. *IEEE Trans Autom Control* 65(3):909–924
- Rahmani A, Ji M, Mesbahi M, Egerstedt M (2009) Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM J Control Optim* 48(1):162–186
- Rajapakse I, Groudine M, Mesbahi M (2011) Dynamics and control of state-dependent networks for probing genomic organization. *Proc Natl Acad Sci* 108(42):17257–17262
- Reinschke KJ (1988) *Multivariable control: a graph-theoretic approach*. Springer
- Shaker HR, Tahavori M (2012) Optimal sensor and actuator location for unstable systems. *J Vib Control*. <https://doi.org/10.1177/1077546312451302>. <http://jvc.sagepub.com/cgi/content/abstract/1077546312451302v1>
- Sharon PL, Rex KK (1999) Optimization strategies for sensor and actuator placement. Technical report, NASA Langley Technical Report
- Skardal PS, Arenas A (2015) Control of coupled oscillator networks with application to microgrid technologies. *Sci Adv* 1(7):e1500339
- Summers TH, Lygeros J (2014) Optimal sensor and actuator placement in complex dynamical networks. *IFAC Proc Vol* 47(3):3784–3789
- Sun J, Motter AE (2013) Controllability transition and nonlocality in network control. *Phys Rev Lett* 110(20):208701
- Tabuada P, Ma W, Grizzle J, Ames AD: Data-driven control for feedback linearizable single-input systems. In: *IEEE conference on decision and control*, Melbourne, pp 6265–6270 (2017)
- Tchoń K (1983) On generic properties of linear systems: an overview. *Kybernetika* 19(6):467–474
- Van De Wal M, De Jager B (2001) A review of methods for input/output selection. *Automatica* 37(4):487–510
- van Waarde HJ, Eising J, Trentelman HL, Camlibel MK (2019) Data informativity: a new perspective on data-driven analysis and control. *arXiv preprint arXiv:1908.00468*
- Wonham WM (1985) *Linear multivariable control: a geometric approach*, 3rd edn. Springer
- Zhao S, Pasqualetti F (2017) Discrete-time dynamical networks with diagonal controllability gramian. In: *IFAC world congress*, Toulouse, pp 8297–8302
- Zhao S, Pasqualetti F (2018) Controllability degree of directed line networks: nodal energy and asymptotic bounds. In: *European control conference*, Limassol, pp 1857–1862
- Zhao S, Pasqualetti F (2019) Networks with diagonal controllability gramians: analysis, graphical conditions, and design algorithms. *Automatica* 102:10–18

Controller Performance Monitoring

Sirish L. Shah

Department of Chemical and Materials Engineering, University of Alberta Edmonton, Edmonton, AB, Canada

Abstract

Process control performance is a cornerstone of operational excellence in a broad spectrum of industries such as refining, petrochemicals, pulp and paper, mineral processing, power systems, and wastewater treatment. Control performance assessment and monitoring applications have become mainstream in these industries and are changing the maintenance methodology surrounding control assets from predictive to condition based. The large numbers of these assets on most sites compared to the number of maintenance and control personnel have made monitoring and diagnosing control problems challenging. For this reason, automated controller performance monitoring technologies have been readily embraced by these industries.

This entry discusses the theory as well as practical application of controller performance monitoring tools as a requisite for monitoring and maintaining basic as well as advanced process control (APC) assets in the process industry. The section begins with the introduction to the theory of performance assessment as a technique for assessing the performance of the basic control loops in a plant. Performance assessment allows detection of performance degradation in the basic control loops in a plant by monitoring the variance in the process variable and comparing it to that of a minimum variance controller. Other metrics of controller performance are also reviewed. The resulting indices of performance give an indication of the level of performance of the controller and an indication

of the action required to improve its performance; the diagnosis of poor performance may lead one to look at remediation alternatives such as retuning controller parameters or process reengineering to reduce delays or implementation of feed-forward control or attribute poor performance to faulty actuators or other process nonlinearities.

Keywords

Time series analysis · Minimum variance control · Control loop performance assessment · Performance monitoring · Fault detection and diagnosis

Introduction

A typical industrial process, as in a petroleum refinery or a petrochemical complex, includes thousands of control loops. Instrumentation technicians and engineers maintain and service these loops but rather infrequently. However, industrial studies have shown that as many as 60% of control loops may have poor tuning or configuration or actuator problems and may therefore be responsible for suboptimal process performance. As a result, monitoring of such control strategies to detect and diagnose cause(s) of unsatisfactory performance has received increasing attention from industrial engineers. Specifically the methodology of data-based controller performance monitoring (CPM) is able to answer questions such as the following: Is the controller doing its job satisfactorily, and, if not, what is the cause of poor performance?

The performance of process control assets is monitored on a daily basis and compared with industry benchmarks. The monitoring system also provides diagnostic guidance for poorly performing control assets. Many industrial sites have established reporting and remediation workflows to ensure that improvement activities are carried out in an expedient manner. Plant-wide performance metrics can provide insight into

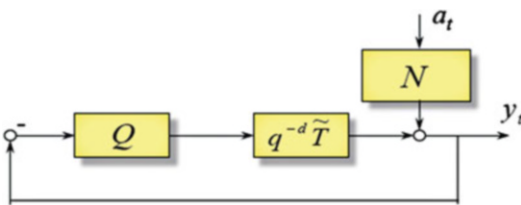
company-wide process control performance. Closed-loop tuning and modeling tools can also be deployed to aid with the improvement activities. Survey articles by Thornhill and Horch (2007) and Shardt et al. (2012) provide a good overview of the overall state of CPM and the related diagnosis issues. CPM software is now readily available from most DCS vendors and has already been implemented successfully at many large-scale industrial sites throughout the world.

Univariate Control Loop Performance Assessment with Minimum Variance Control as Benchmark

It has been shown by Harris (1989) that for a system with time delay d , a portion of the output variance is feedback control invariant and can be estimated from routine operating data. This is the so-called minimum variance output. Consider the closed-loop system shown in Fig. 1, where Q is the controller transfer function, $\tilde{T}$ is the process transfer function, d is the process delay (in terms of sample periods), and N is the disturbance transfer function driven by random white-noise sequence, a_t .

In the regulatory mode (when the set point is constant), the closed-loop transfer function relating the process output and the disturbance is given by

$$\text{Closed-loop response: } y_t = \left(\frac{N}{1 + q^{-d} \tilde{T} Q} \right) a_t$$



Controller Performance Monitoring, Fig. 1 Block diagram of a regulatory control loop

Note that all transfer functions are expressed for the discrete time case in terms of the backshift operator, q^{-1} . N represents the disturbance transfer function with numerator and denominator polynomials in q^{-1} . The division of the numerator by the denominator can be rewritten as: $N = F + q^{-d} R$, where the quotient term $F = F_0 + F_1 q^{-1} + \dots + F_{d-1} q^{-(d-1)}$ is a polynomial of order $(d-1)$ and the remainder R is a transfer function. The closed-loop transfer function can be reexpressed, after algebraic manipulation as

$$\begin{aligned} y_t &= \left(\frac{N}{1 + q^{-d} \tilde{T} Q} \right) a_t \\ &= \left(\frac{F + q^{-d} R}{1 + q^{-d} \tilde{T} Q} \right) a_t \\ &= \left(\frac{F(1 + q^{-d} \tilde{T} Q) + q^{-d} (R - F \tilde{T} Q)}{1 + q^{-d} \tilde{T} Q} \right) a_t \\ &= \left(F + q^{-d} \frac{R - F \tilde{T} Q}{1 + q^{-d} \tilde{T} Q} \right) a_t \\ &= \underbrace{F_0 a_t + F_1 a_{t-1} + \dots + F_{d-1} a_{t-d+1}}_{e_t} \\ &\quad + \underbrace{L_0 a_{t-d} + L_1 a_{t-d-1} + \dots}_{w_{t-d}} \end{aligned}$$

The closed-loop output can then be expressed as

$$y_t = e_t + w_{t-d}$$

where $e_t = F a_t$ corresponds to the first $d-1$ lags of the closed-loop expression for the output, y_t , and more importantly is independent of the controller, Q , or it is controller invariant, while w_{t-d} is dependent on the controller. The variance of the output is then given by

$$\text{Var}(y_t) = \text{Var}(e_t) + \text{Var}(w_{t-d}) \geq \text{Var}(e_t)$$

Since e_t is controller invariant, it provides the lower bound on the output variance. This is naturally achieved if $w_{t-d} = 0$, that is, when $R = F \tilde{T} Q$ or when the controller is a minimum

variance controller with $Q = \frac{R}{F\bar{T}}$. If the total output variance is denoted as $Var(y_t) = \sigma^2$, then the lowest achievable variance is $Var(e_t) = \sigma_{mv}^2$. To obtain an estimate of the lowest achievable variance from the time series of the process output, one needs to model the closed-loop output data y_t by a moving average process such as

$$y_t = \underbrace{f_0 a_t + f_1 a_{t-1} + \cdots + f_{d-1} a_{t-(d-1)}}_{e_t} + f_d a_{t-d} + f_{d+1} a_{t-(d+1)} + \cdots \quad (1)$$

The controller-invariant term e_t can then be estimated by time series analysis of routine closed-loop operating data and subsequently used as a benchmark measure of theoretically achievable absolute lower bound of output variance to assess control loop performance. Harris (1989), Desbor-

ough and Harris (1992), and Huang and Shah (1999) have derived algorithms for the calculation of this minimum variance term.

Multiplying Eq. (1) by $a_t, a_{t-1}, \dots, a_{t-d+1}$, respectively, and then taking the expectation of both sides of the equation yield the sample covariance terms:

$$\left. \begin{aligned} r_{ya}(0) &= E[y_t a_t] = f_0 \sigma_a^2 \\ r_{ya}(1) &= E[y_t a_{t-1}] = f_1 \sigma_a^2 \\ r_{ya}(2) &= E[y_t a_{t-2}] = f_2 \sigma_a^2 \\ &\vdots \\ r_{ya}(d-1) &= E[y_t a_{t-d+1}] = f_{d-1} \sigma_a^2 \end{aligned} \right\} \quad (2)$$

The minimum variance or the invariant portion of output variance is

$$\left. \begin{aligned} \sigma_{mv}^2 &= (f_0^2 + f_1^2 + f_2^2 + \cdots + f_{d-1}^2) \sigma_a^2 \\ &= \left[\left(\frac{r_{ya}(0)}{\sigma_a^2} \right)^2 + \left(\frac{r_{ya}(1)}{\sigma_a^2} \right)^2 + \left(\frac{r_{ya}(2)}{\sigma_a^2} \right)^2 + \left(\frac{r_{ya}(d-1)}{\sigma_a^2} \right)^2 \right] \sigma_a^2 \\ &= [r_{ya}^2(0) + r_{ya}^2(1) + r_{ya}^2(2) + \cdots + r_{ya}^2(d-1)] / \sigma_a^2 \end{aligned} \right\} \quad (3)$$

A measure of controller performance index can then be defined as

$$\eta(d) = \sigma_{mv}^2 / \sigma_y^2 \quad (4)$$

Substituting Eq. (3) into Eq. (4) yields

$$\begin{aligned} \eta(d) &= [r_{ya}^2(0) + r_{ya}^2(1) + r_{ya}^2(2) + \cdots + r_{ya}^2(d-1)] / \sigma_y^2 \sigma_a^2 \\ &= \rho_{ya}^2(0) + \rho_{ya}^2(1) + \rho_{ya}^2(2) + \cdots + \rho_{ya}^2(d-1) \\ &= ZZ^T \end{aligned}$$

where Z is the vector of cross correlation coefficients between y_t and a_t for lags 0 to $d-1$ and is denoted as

$$Z = [\rho_{ya}(0) \rho_{ya}(1) \rho_{ya}(2) \cdots \rho_{ya}(d-1)]$$

Although a_t is unknown, it can be replaced by the estimated innovation sequence $\hat{a}_t$. The estimate

$\hat{a}_t$ is obtained by whitening the process output variable y_t via time series analysis. This algorithm is denoted as the FCOR algorithm for Filtering and CORrelation analysis (Huang and Shah 1999). This derivation assumes that the delay, d , be known a priori. In practice, however, a priori knowledge of time delays may not always be available. It is therefore useful to

assume a range of time delays and then calculate performance indices over this range of the time delays. The indices over a range of time delays are also known as extended horizon performance indices (Thornhill et al. 1999). Through pattern recognition, one can tell the performance of the loop by visualizing the patterns of the performance indices versus time delays. There is a clear relation between performance indices curve and the impulse response curve of the control loop.

Consider a simple case where the process is subject to random disturbances. Figure 2 is one example of performance evaluation for a control loop in the presence of disturbances. This figure shows time series of process variable data for both loops in the top 2 plots, closed-loop impulse responses (left columns), and corresponding performance indices (labeled as PI on the right columns). From the impulse responses, one can see that the loop under the first set of tuning constants (denoted as TAG1.PV) has better performance; the loop under the second set of tuning constants (denoted as TAG5.PV) has oscillatory behavior, indicating a relatively poor control performance. With performance index “1” indicating the best possible performance and index “0” indicating the worst performance, performance indices for the first controller tuning (shown on the upper-right plot) approach “1” within four time lags, while performance indices for the second controller tuning (shown on the bottom-right plot) take ten time lags to approach “0.7.” In addition, performance indices for the second tuning show ripples as they approach an asymptotic limit, indicating a possible oscillation in the loop.

Notice that one cannot rank performance of these two controller settings from the noisy time series data. Instead, we can calculate performance indices over a range of time delays (from 1 to 10). The result is shown on the right column plots of Fig. 2. These simulations correspond to the same process with different controller tuning constants. It is clear from these plots that performance indices trajectory depends on dynamics of the disturbance and controller tuning.

It is important to note that the minimum variance is just one of several benchmarks for

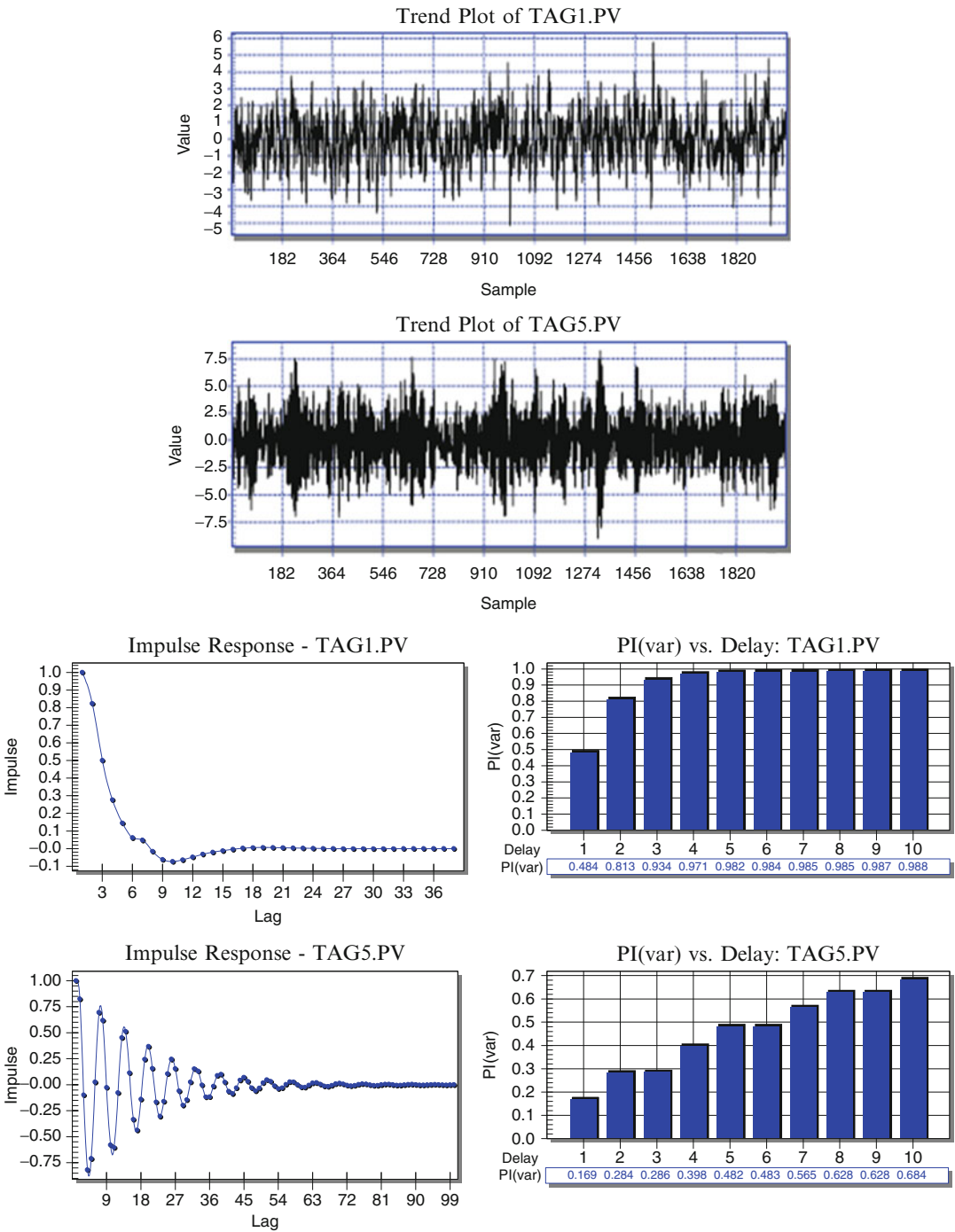
obtaining a controller performance metric. It is seldom practical to implement minimum variance control as it typically will require aggressive actuator action. However, the minimum variance benchmark serves to provide an indication of the opportunity in improving control performance; that is, should the performance index $\eta(d)$ be near or just above zero, then it gives the user an idea of the benefits possible in improving the control performance of that loop.

Performance Assessment and Diagnosis of Univariate Control Loop Using Alternative Performance Indicators

In addition to the performance index for performance assessment, there are several alternative indicators of control loop performance. These are discussed next.

Autocorrelation function The autocorrelation function (ACF) of the output error, shown in Fig. 3, is an approximate measure of how close the existing controller is to minimum variance condition or how predictable the error is over the time horizon of interest. If the controller is under minimum variance condition, then the autocorrelation function should decay to zero after “ $d - 1$ ” lags where “ d ” is the delay of the process. In other words, there should be no predictable information beyond time lag $d - 1$. The rate at which the autocorrelation decays to zero after “ $d - 1$ ” lags indicates how close the existing controller is to the minimum variance condition. Since it is straightforward to calculate autocorrelation using process data, the autocorrelation function is often used as a first-pass test before carrying out further performance analysis.

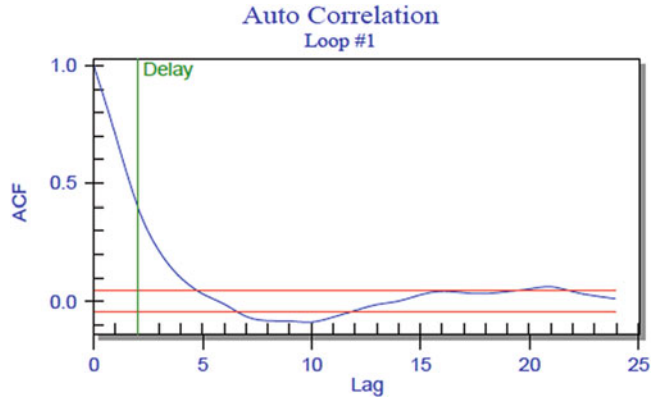
Impulse response An impulse response function curve represents the closed-loop impulse response between the whitened disturbance sequence and the process output. This function is a direct measure of how well the controller is performing in rejecting disturbances or tracking set-point changes. Under stochastic framework,



Controller Performance Monitoring, Fig. 2 Time series of process variable (top), corresponding impulse responses (left column) and their performance indices (right column)

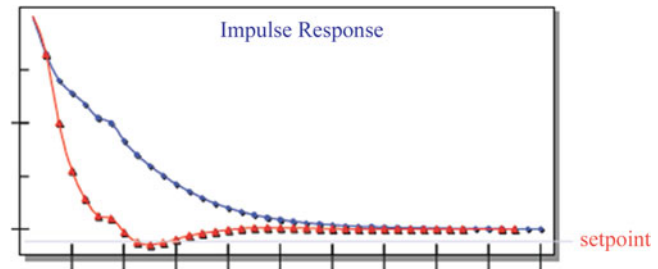
Controller Performance Monitoring,

Fig. 3 Autocorrelation function of the controller error



Controller Performance Monitoring,

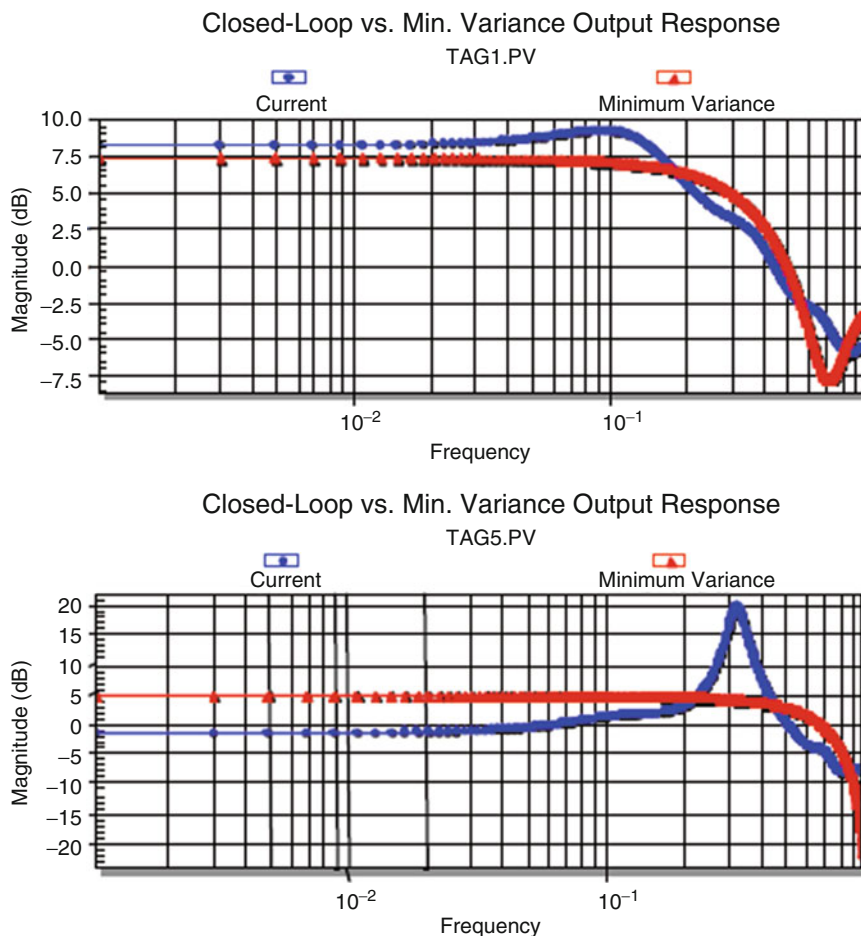
Fig. 4 Impulse responses estimated from routine operating data



this impulse response function may be calculated using time series model such as an autoregressive moving average (ARMA) or autoregressive integrated moving average (ARIMA) model. Once an ARMA type of time series model is estimated, the infinite-order moving average representation of the model shown in Eq. (1) can be obtained through a long division of the ARMA model. As shown in Huang and Shah (1999), the coefficients of the moving average model, Eq. (1), are the closed-loop impulse response coefficients of the process between whitened disturbances and the process output. Figure 4 shows closed-loop impulse responses of a control loop with two different control tunings. Clearly they denote two different closed-loop dynamic responses: one is slow and smooth, and the other one is relatively fast and slightly oscillatory. The sum of square of the impulse response coefficients is the variance of the data.

Spectral analysis The closed-loop frequency response of the process is an alternative way to assess control loop performance. Spectral analysis of output data easily allows one to

detect oscillations, offsets, and measurement noise present in the process. The closed-loop frequency response is often plotted together with the closed-loop frequency response under minimum variance control. This is to check the possibility of performance improvement through controller tunings. The comparison gives a measure of how close the existing controller is to the minimum variance condition. In addition, it also provides the frequency range in which the controller significantly deviates from minimum variance condition. Large deviation in the low-frequency range typically indicates lack of integral action or weak proportional gain. Large peaks in the medium-frequency range typically indicate an overtuned controller or presence of oscillatory disturbances. Large deviation in the high-frequency range typically indicates significant measurement noise. As an illustrative example, frequency responses of two control loops are shown in Fig. 5. The top graph of the figure shows that closed-loop frequency response of the existing controller is almost the same as the frequency response under minimum variance control. A peak at the mid-frequency indicates



Controller Performance Monitoring, Fig. 5 Frequency response estimated from routine operating data

possible overtuned control. The bottom graph of Fig. 5 shows that the frequency response of the existing controller is oscillatory, indicating a possible overtuned controller or the presence of an oscillatory disturbance at the peak frequency; otherwise the controller is close to minimum variance condition.

Segmentation of performance indices Most process data exhibit time-varying dynamics; i.e., the process transfer function or the disturbance transfer function is time variant. Performance assessment with a nonoverlapping sliding data window that can track time-varying dynamics is therefore often desirable. For example, segmentation of data may lead to some insight into any cyclical behavior of the process variation

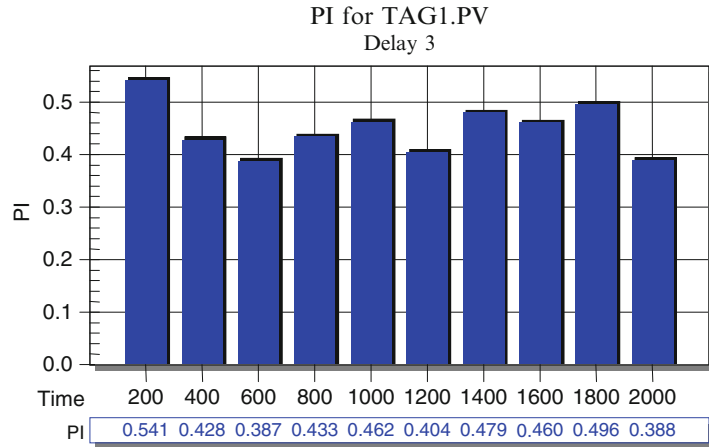
in controller performance during, e.g., due to day versus night process operation or due to shift changes. Figure 6 is an example of performance segmentation over a 200 data wide window.

Performance Assessment of Univariate Control Loops Using User-Specified Benchmarks

The increasing level of global competitiveness has pushed chemical plants into high-performance operating regions that require advanced process control technology. See the articles ► [Control Hierarchy of Large Processing Plants: An Overview](#) and ► [Control Structure Selection](#). Consequently, industry has

Controller Performance Monitoring,

Fig. 6 Performance indices for segmented data (each of window length 200 points)



an increasing need to upgrade the conventional PID controllers to advanced control systems. The most natural questions to ask for such an upgrading are as follows. Has the advanced controller improved performance as expected? If yes, where is the improvement and can it be justified? Has the advanced controller been tuned to its full capacity? Can this improvement also be achieved by simply retuning the existing traditional (e.g., PID) controllers? (see ► [PID Control](#)). In other words, what is the cost versus benefit of implementing an advanced controller? Unlike performance assessment using minimum variance control as benchmark, the solution to this problem does not require a priori knowledge of time delays. Two possible relative benchmarks may be chosen: one is the historical data benchmark or reference data set benchmark, and the other is a user-specified benchmark.

The purpose of reference data set benchmarking is to compare performance of the existing controller with the previous controller during the “normal” operation of the process. This reference data set may represent the process when the controller performance is considered satisfactory with respect to meeting the performance objectives. The reference data set should be representative of the normal conditions that the process is expected to operate at; i.e., the disturbances and set-point changes entering into the process should not be unusually different. This analysis provides the user with a relative performance index (RPI) which compares the existing control loop

performance with a reference control loop benchmark chosen by the user. The RPI is bounded by $0 \leq RPI \leq \infty$, with “<1” indicating deteriorated performance, “1” indicating no change of performance, and “>1” indicating improved performance. Figure 7 shows a result of reference data set benchmarking. The impulse response of the benchmark or reference data smoothly decays to zero, indicating good performance of the controller. After one increases the proportional gain of the controller, the impulse response shows oscillatory behavior, with an $RPI = 0.4$, indicating deteriorated performance due to the oscillation.

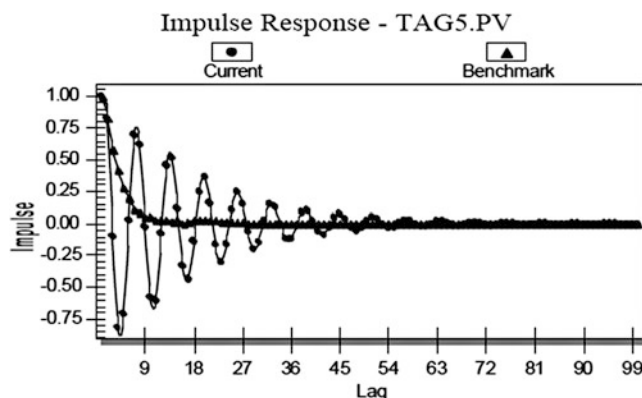
In some cases one may wish to specify certain desired closed-loop dynamics and carry out performance analysis with respect to such desired dynamics. One such desired dynamic benchmark is the closed-loop settling time. As an illustrative example, Figure 8 shows a system where a settling time of ten sampling units is desired for a process with a delay of five sampling units. The impulse responses show that the existing loop is close to the desired performance and the value of $RPI = 0.9918$ confirms this. Thus no further tuning of the loop is necessary.

Diagnosis of Poorly Performing Loops

Whereas detection of poorly performing loops is now relatively simple, the task of diagnosing

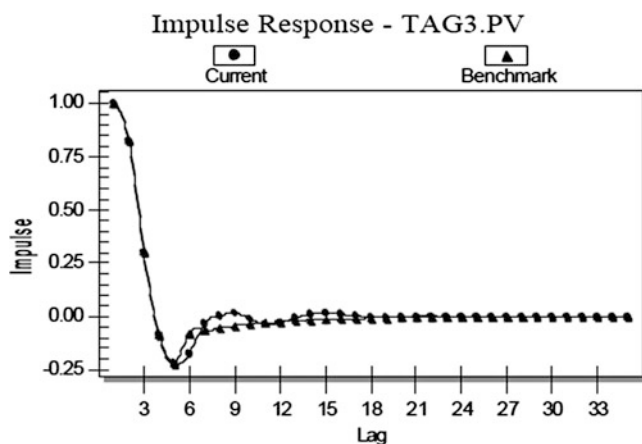
Controller Performance Monitoring,

Fig. 7 Reference benchmarking based on impulse responses



Controller Performance Monitoring,

Fig. 8 User-specified benchmark based on impulse responses



reason(s) for poor performance and how to “mend” the loop is generally not straightforward. The reasons for poor performance could be any one of interactions between various control loops, overtuned or undertuned controller settings, process nonlinearity, poor controller configuration (meaning poor choice of pairing a process (or controlled) variable with a manipulative variable loop), or actuator problems such as stiction, large delays, and severe disturbances. Several studies have focused on the diagnosis issues related to actuator problems (Håagglund 2002; Choudhury et al. 2008; Srinivasan and Rengaswamy 2008; Xiang and Lakshminarayanan 2009; de Souza et al. 2012). Shardt et al. (2012) has given an overview of the overall state of CPM and the related diagnosis issues.

Industrial Applications of CPM Technology

As remarked earlier, CPM software is now readily available from most DCS vendors and has already been implemented successfully at several large-scale industrial sites. A summary of just two of many large-scale industrial implementations of CPM technology appears below. It gives a clear evidence of the impact of this control technology and how readily it has been embraced by industry (Shah et al. 2014).

BASF Controller Performance Monitoring Application

As part of its excellence initiative OPAL 21 (Optimization of Production Antwerp and

Ludwigshafen), BASF has implemented the CPM strategy on more than 30,000 control loops at its Ludwigshafen site in Germany and on over 10,000 loops at its Antwerp production facility in Belgium. The key factor in using this technology effectively is to combine process knowledge, basic chemical engineering, and control expertise to develop solutions for the indicated control problems that are diagnosed in the CPM software (Wolff et al. 2012).

Saudi Aramco Controller Performance Monitoring Practice

As part of its process control improvement initiative, Saudi Aramco has deployed CPM on approximately 15,000 PID loops, 50 MPC applications, and 500 smart positioners across multiple operating facilities.

The operational philosophy of the CPM engine is incorporated in the continuous improvement process at BASF and Aramco, whereby all loops are monitored in real time and a holistic performance picture is obtained for the entire plant. Unit-wide performance metrics are displayed in effective color-coded graphic forms to effectively convey the analytics information of the process.

Concluding Remarks

In summary, industrial control systems are designed and implemented or upgraded with a particular objective in mind. The controller performance monitoring methodology discussed here will permit automated and repeated reviews of the design, tuning, and upgrading of the control loops. Poor design, tuning, or upgrading of the control loops can be detected, and repeated performance monitoring will indicate which loops should be retuned or which loops have not been effectively upgraded when changes in the disturbances, in the process, or in the controller itself occur. Obviously better design, tuning, and upgrading will mean that the process will operate at a point close to the economic optimum, leading to energy savings, improved safety, efficient utilization of raw materials, higher product

yields, and more consistent product qualities. This entry has summarized the major features available in recent commercial software packages for control loop performance assessment. The illustrative examples have demonstrated the applicability of this new technique when applied to process data.

This entry has also illustrated how controllers, whether in hardware or software form, should be treated like “capital assets” and how there should be routine monitoring to ensure that they perform close to the economic optimum and that the benefits of good regulatory control will be achieved.

Cross-References

- [Control Hierarchy of Large Processing Plants: An Overview](#)
- [Control Structure Selection](#)
- [Fault Detection and Diagnosis](#)
- [PID Control](#)
- [Statistical Process Control in Manufacturing](#)

Bibliography

- Choudhury MAAS, Shah SL, Thornhill NF (2008) Diagnosis of process nonlinearities and valve stiction: data driven approaches. Springer-Verlag, Sept. 2008, ISBN:978-3-540-79223-9
- Desborough L, Harris T (1992) Performance assessment measure for univariate feedback control. *Can J Chem Eng* 70:1186–1197
- Desborough L, Miller R (2002) Increasing customer value of industrial control performance monitoring: Honeywell’s experience. In: *AIChE symposium series*. American Institute of Chemical Engineers, New York, pp 169–189; 1998
- de Prada C (2014) Overview: control hierarchy of large processing plants. In: *Encyclopedia of systems and control*. Springer, London
- de Souza LCMA, Munaro CJ, Munareto S (2012) Novel model-free approach for stiction compensation in control valves. *Ind Eng Chem Res* 51(25):8465–8476
- Hågglund T (2002) A friction compensator for pneumatic control valves. *J Process Control* 12(8):897–904
- Harris T (1989) Assessment of closed loop performance. *Can J Chem Eng* 67:856–861
- Huang B, Shah SL (1999) Performance assessment of control loops: theory and applications. Springer-Verlag, Oct 1999, ISBN: 1-85233-639-0

- Shah SL, Nohr M, Patwardhan R (2014) Success stories in control: controller performance monitoring. In: Samad T, Annaswamy AM (eds) The impact of control technology, 2nd edn. www.ieeeccs.org
- Shardt YAW, Zhao Y, Lee KH, Yu X, Huang B, Shah SL (2012) Determining the state of a process control system: current trends and future challenges. *Can J Chem Eng* 90(2):217–245
- Srinivasan R, Rengaswamy R (2008) Approaches for efficient stiction compensation in process control valves. *Comput Chem Eng* 32(1):218–229
- Skogestad S (2014) Control structure selection and plantwide control. In: Encyclopedia of systems and control. Springer, London
- Thornhill NF, Horch A (2007) Advances and new directions in plant-wide controller performance assessment. *Control Eng Pract* 15(10):1196–1206
- Thornhill NF, Oettinger M, Fedenczuk P (1999) Refinery-wide control loop performance assessment. *J Process Control* 9(2):109–124
- Wolff F, Roth M, Nohr A, Kahrs O (2012) Software based control-optimization for the chemical industry. VDI, Tagungsband “Automation 2012”, 13/14.06 2012, Baden-Baden
- Xiang LZ, Lakshminarayanan S (2009) A new unified approach to valve stiction quantification and compensation. *Ind Eng Chem Res* 48(7):3474–3483

Controller Synthesis for CPS

Paulo Tabuada

Electrical and Computer Engineering
Department, UCLA, Los Angeles, CA, USA
Department of Electrical Engineering,
University of California, Los Angeles, CA, USA

Abstract

The analysis and design of Cyber-Physical Systems (CPS) remain a challenge in no small part due to the intricate interactions between cyber and physical components. In this entry we address the problem of synthesizing a cyber component so that its interaction with a CPS satisfies a desired specification by construction. This paradigm is often termed “formal controller synthesis”, or simply “synthesis”, and is an alternative to the more commonly used verification paradigm.

Keywords

Controller synthesis · Formal synthesis · Synthesis · CPS

What is Formal Controller Synthesis?

The analysis and design of Cyber-Physical Systems (CPS) remain a challenge in no small part due to the intricate interactions between cyber and physical components. In this entry we address the problem of synthesizing a cyber component so that its interaction with a CPS satisfies a desired specification by construction. This paradigm is often termed “formal controller synthesis,” or simply “synthesis,” and is an alternative to the more commonly used verification paradigm.

Each paradigm has its advantages and disadvantages. In the verification paradigm, one first constructs the cyber component and then verifies that its interaction with the CPS satisfies the desired specifications. When the verification step identifies that the specification is not satisfied, it is necessary to re-design the cyber component and to re-verify it. This results in a cycle, alternating between the design and verification phases that can be time consuming. In contrast, in the synthesis paradigm, a cyber component is produced along with a proof that it meets the desired specifications. Hence, iteration is not required, and for this reason, synthesis is an example of a *correct-by-construction* or *correct-by-design* methodology. Synthesis, however, is a much more difficult algorithmic problem than verification. In fact, one can view verification as a particular case of synthesis: if the result of synthesis for a CPS is that no cyber component is required, it constitutes a proof that the CPS already meets the specification. For this reason, verification is applicable to a much wider class of CPS than synthesis.

In this entry we review the existing approaches in the literature for CPS synthesis. Rather than attempting to be exhaustive, we focus on the key ideas that underlie this design methodology. In

particular, an important line of research that is not discussed is that of stochastic systems.

Problem Formulation

To formalize the synthesis problem, we need to introduce formal models for systems, specifications, and system composition.

Modeling Systems

We consider a fairly abstract notion of system that can be used to describe both cyber and physical components.

Definition 1 (System) A system Σ is a tuple $\Sigma = (X, X_0, U, F, Y, H)$ consisting of:

- a set of states X ;
- a set of initial states $X_0 \subseteq X$;
- a set of inputs U ;
- a set-valued map $F : X \times U \rightarrow 2^X$ where 2^X denotes the set of all subsets of X ;
- a set of outputs Y ;
- an output map $H : X \rightarrow Y$;

The dynamics is described by the map F , and its set-valued nature is used to model non-determinism that can arise from coarse modeling or from the effect of exogenous disturbances. Given a state $x \in X$ and input $u \in U$, the next state is known to belong to the set $F(x, u)$ and is only uniquely determined when $F(x, u)$ is a singleton. The dynamics of a system gives rise to its *behavior*.

Definition 2 (System behavior) An internal behavior ξ of a system Σ is an infinite sequence $\xi \in X^{\mathbb{N}}$ such that:

1. $\xi(0) \in X_0$;
2. There exists an infinite sequence of inputs $v \in U^{\mathbb{N}}$ such that $\xi(i+1) \in F(\xi(i), v(i))$ for all $i \in \mathbb{N}$.

Every internal behavior induces an external behavior $\zeta \in Y^{\mathbb{N}}$ defined by $\zeta(i) = H(\xi(i))$ for all $i \in \mathbb{N}$. We denote the set of external behaviors of a system Σ by $\mathcal{B}(\Sigma)$.

When modeling cyber components, the set of states X is typically finite, and thus we call such systems *finite systems*. However, models for physical components typically require (uncountably) infinite sets of states.

Modeling Specifications

Specification for the desired behavior of a system Σ are typically given as a set of desired behaviors $S \subseteq Y^{\mathbb{N}}$. In practice, the set S is given either as the output behavior of a finite system Σ_S , i.e., $S = \mathcal{B}(\Sigma_S)$, or as a temporal logic formula. It is outside the scope of this entry to venture into temporal logic. The interested reader can find all the relevant details in the literature, e.g., Emerson (1990) and Stirling (1992). For the purpose of understanding the CPS synthesis problem, it suffices to know that a temporal logic formula φ defines S as the set of all infinite sequences $\zeta \in Y^{\mathbb{N}}$ satisfying φ .

To make these ideas more concrete, let us assume we separate the set of states X into safe and unsafe states. We can then define Y to be the set **{safe, unsafe}** and define H by mapping safe states $x \in X$ to $H(x) = \mathbf{safe}$ and unsafe states $x \in X$ to $H(x) = \mathbf{unsafe}$. Consider now the formula $\varphi = \Diamond \mathbf{safe}$ requiring the output along an external behavior to eventually become **safe**. In other words, φ requires the state along an internal behavior to eventually become safe and thus produce the output **safe**. An external behavior $\zeta \in Y^{\mathbb{N}}$ satisfies φ , denoted by $\zeta \models \varphi$, if there exists an $i \in \mathbb{N}$ for which we have $\zeta(i) = \mathbf{safe}$. We could consider instead the formula $\varphi = \Box \mathbf{safe}$ requiring all the states visited along an internal behavior to be safe states. An external behavior $\zeta \in Y^{\mathbb{N}}$ satisfies $\Box \mathbf{safe}$ if for every $i \in \mathbb{N}$ we have $\zeta(i) = \mathbf{safe}$. The specification S defined by the formula $\Box \mathbf{safe}$ would then contain a single external behavior:

safe safe safe ...,

whereas for $\varphi = \Diamond \mathbf{safe}$, the set S would consist of all the external behaviors of the form:

$\underbrace{Y Y Y Y}_{k \text{ times}} \mathbf{safe} Y Y Y \dots$,

with $k \in \mathbb{N}$ and where Y represents an output that can either be **safe** or **unsafe**.

Composing Systems

The notion of composition describes the system that results from the concurrent execution of two systems mediated by their interaction. The specific way in which they interact is described by an interconnection relation.

Definition 3 Let $S_a = (X_a, X_{a0}, U_a, F_a, Y_a, H_a)$ and $S_b = (X_b, X_{b0}, U_b, F_b, Y_b, H_b)$ be two systems and let $\mathcal{I} \subseteq X_a \times X_b \times U_a \times U_b$ be a relation. The composition of S_a with S_b with interconnection relation $\mathcal{I}$, denoted by $S_a \times_{\mathcal{I}} S_b$, is the system $(X_{ab}, X_{ab0}, U_{ab}, F_{ab}, Y_{ab}, H_{ab})$ consisting of:

- $X_{ab} = \{(x_a, x_b) \in X_a \times X_b \mid \exists(u_a, u_b) \in U_a \times U_b (x_a, x_b, u_a, u_b) \in \mathcal{I}\};$
- $X_{ab0} = X_{ab} \cap (X_{a0} \times X_{b0});$
- $U_{ab} = U_a \times U_b;$
- $(x'_a, x'_b) \in F_{ab}(x_a, x_b, u_a, u_b)$ if the following three conditions hold:
 1. $x'_a \in F_a(x_a, u_a);$
 2. $x'_b \in F_b(x_b, u_b);$
 3. $(x_a, x_b, u_a, u_b) \in \mathcal{I};$
- $Y_{ab} = Y_a \times Y_b;$
- $H_{ab}(x_a, x_b) = (H_a(x_a), H_b(x_b)).$

To illustrate the notion of composition, consider an arbitrary system S_a that is to be controlled by a feedback control law $K : X_a \rightarrow U_a$. We can define a new system S_b by $(X_a, X_{a0}, U_a, F_a, U_a, K)$ where we used the output map to describe the control law. By connecting the output of S_b to the input of S_a , we will be forcing the inputs of S_a to be chosen according to the control law. This can be accomplished by defining the interconnection relation $\mathcal{I}$ as the set of tuples $(x_a, x_b, u_a, u_b) \in X_a \times X_b \times U_a \times U_b$ satisfying $x_a = x_b$ and $u_a = K(x_b) = H_b(x_b)$. By suitably defining $\mathcal{I}$, we can model series, parallel, and other types of interconnections between systems as detailed in Tabuada (2009).

The Formal Controller Synthesis Problem

We now mathematically define the synthesis problem discussed in this entry.

Problem 1 (Formal controller synthesis)

Given a system Σ and a specification S , synthesize a system C (the controller) and an interconnection relation $\mathcal{I}$ so that:

$$\mathcal{B}(C \times_{\mathcal{I}} \Sigma) \subseteq S.$$

When the specification S is given by a temporal logic formula, we write $C \times_{\mathcal{I}} \Sigma \models \varphi$ to denote that every $\zeta \in \mathcal{B}(C \times_{\mathcal{I}} \Sigma)$ satisfies φ .

This version of the synthesis problem is sometimes termed *behavioral inclusion* since it requires the external behavior of $C \times_{\mathcal{I}} \Sigma$ to be included in S . However, there are properties that cannot be specified via behavioral inclusion and other variations of the synthesis problem exist in the literature, e.g., Tabuada (2008, 2009) and Belta et al. (2017).

Abstraction-Based Synthesis

The synthesis problem is much better understood for cyber systems. Mathematically, this means that Σ is a finite system, and we are seeking a finite controller C . For this reason, CPS synthesis is often reduced to a synthesis problem for finite systems. Such reduction can be accomplished by constructing a *finite abstraction* $\tilde{\Sigma}$ of the system Σ modeling a CPS, so that a controller $\tilde{C}$ enforcing the specification S on $\tilde{\Sigma}$ can be refined to a controller C enforcing the specification S on Σ . In symbols, we have:

$$\mathcal{B}(\tilde{C} \times_{\tilde{\mathcal{I}}} \tilde{\Sigma}) \subseteq S \implies \mathcal{B}(C \times_{\mathcal{I}} \Sigma) \subseteq S. \quad (1)$$

This solution strategy implicitly assumes that:

1. One can synthesize $\tilde{C}$ when $\tilde{\Sigma}$ is finite.
2. One can construct the abstraction $\tilde{\Sigma}$ from Σ .
3. One can construct C from $\tilde{C}$.
4. It is easier to perform the preceding three steps than directly synthesizing C .

We now address each of these assumptions in detail.

Formal Controller Synthesis for Finite Systems

Synthesis for finite systems has a long history, and this problem is fairly well understood. Although the search for more efficient synthesis algorithms is still an active area of research, several software tools are available for this purpose. The interested readers can find an introduction to synthesis for finite systems in, e.g., Chapter 27 of Clarke et al. (2018), Chapter 5 of Belta et al. (2017), and a gentler introduction, for the special case of safety and reachability specifications, in Chapter 6 of Tabuada (2009). A different approach to synthesis for finite systems, based on discrete-event systems, can be found in, e.g., Kumar and Garg (1995) and Cassandras and Lafortune (2008).

Construction of Finite Abstractions

Finite abstractions of differential equation models for physical components can be categorized in two classes: *exact* and *approximate*. Both classes are based on the idea of aggregating several states of Σ into a single state in $\tilde{\Sigma}$ in such a way that any synthesized controller $\tilde{C}$ for $\tilde{\Sigma}$ can be refined to a controller C for Σ , i.e., implication (1) holds.

Historically, the idea of using finite abstractions for control was proposed in several papers, e.g., Caines and Wei (1998), Koutsoukos et al. (2000), Fřrstner et al. (2002), and Moor and Raisch (2002), but the first systematic construction of abstractions only appeared several years later. The first class of systems known to admit exact finite abstractions was discrete-time controllable linear systems Tabuada and Pappas (2006), and this enabled, for the first time, the synthesis of controllers enforcing temporal logic specifications on discrete-time linear systems. The key technical idea was to observe that the Brunovsky normal form of controllable linear systems behaves as a finite-state shift register on equivalence classes of states for a suitably defined equivalence relation. A different technique for piecewise linear control systems in continuous time was

first introduced in Habets and Schuppen (2004) to study control problems and then extended in Belta and Habets (2006) to a class on nonlinear systems called multi-affine. This construction exploited the observation that one can determine if all the solutions of a linear differential equation originating inside a simplex will remain within or leave the simplex, by analyzing the linear differential equation at the simplex's vertices. It was then proposed in Kloetzer and Belta (2008) to use this analysis technique as a constructive way to build finite abstractions of multi-affine control systems.

Since these earlier papers, several other extensions and improvements have been proposed in the literature. However, it was also recognized that the class of control systems admitting exact finite abstractions was limited. A major step forward occurred when the notion of *approximate bisimulation* was proposed for control systems in Girard and Pappas (2007) since it enabled the construction of *approximate* finite abstractions for smooth nonlinear systems Pola et al. (2008) and even switched nonlinear systems Girard et al. (2010). The property upon which these first constructions of finite abstractions were based is incremental input-to-state stability and requires, intuitively, that trajectories starting from close initial conditions remain close provided they are produced by inputs that also remain close. Based on this property, it is possible to approximately represent a tube of trajectories in Σ by a single trajectory inside this tube that is then used to build $\tilde{\Sigma}$. These ideas were later extended by showing that weaker forms of abstractions are still possible even in the absence of incremental input-to-state stability Zamani et al. (2012), Liu (2017). Further developments are described in Sect. 1.

Refinement of Controllers

The process of constructing C from $\tilde{C}$ is called controller refinement. Under suitable technical assumptions, the notions of approximate simulation and approximate alternating simulation relation Pola and Tabuada (2009), Tabuada (2009) guarantee that any controller $\tilde{C}$ satisfying $\mathcal{B}(\tilde{C} \times \bar{j}$

$\tilde{\Sigma}) \subseteq S$ can be refined to a controller C satisfying $B(C \times_I \Sigma) \subseteq S$. In general, C can be quite large as it contains $\tilde{C} \times_{\tilde{J}} \tilde{\Sigma}$. Simplifying the controller refinement process and reducing the resulting controller size were some of the motivations for the notion of *feedback refinement relation* introduced in Reissig et al. (2017). This notion results in a stronger requirement than the one imposed by approximate simulation relations but has the advantage of leading to controllers that only require a quantized version of the state. Similar ideas had already been investigated in Girard (2013) in the more specific context of safety and reachability specifications.

Abstraction-Based Synthesis vs Direct Synthesis

The advantage of abstraction-based synthesis is that once $\tilde{\Sigma}$ is available, it can be used to synthesize controllers for different specifications. In other words, $\tilde{\Sigma}$ is specification-agnostic. However, this is also a disadvantage since the construction of abstractions is the main computational bottleneck in abstraction-based synthesis and much of the abstraction may be irrelevant for enforcing a desired specification. This observation led several researchers to investigate specification-guided synthesis techniques. These techniques combine the construction of abstractions with the synthesis of the controller and have the advantage of only computing the part of the abstraction that is relevant for enforcing the given specification. This leads to considerable speedups in the synthesis process. Several different techniques are available in the literature and include the use of on-the-fly abstraction/synthesis techniques (Pola et al. 2012; Rungger and Stursberg 2012), multi-scale abstractions (Girard et al. 2016), and lazy abstraction/synthesis (Hsu et al. 2018; Hussien and Tabuada 2018).

Compositional Synthesis

A different approach to speedup synthesis is based on compositionality. Most CPS result from the composition of smaller modules. Hence, it is natural to exploit this structure for the construc-

tion of abstractions as well as for the synthesis of controllers. Consider a system $\Sigma = \Sigma_1 \times_I \Sigma_2$. In order to construct an abstraction $\tilde{\Sigma}$ of Σ , we can first abstract Σ_1 into $\tilde{\Sigma}_1$, abstract Σ_2 into $\tilde{\Sigma}_2$, and then compose these abstractions into $\tilde{\Sigma}_1 \times_{\tilde{J}} \tilde{\Sigma}_2$. Different technical assumptions and abstractions techniques would then guarantee that $\tilde{\Sigma}_1 \times_{\tilde{J}} \tilde{\Sigma}_2$ is an abstraction of Σ . Since the time complexity of computing $\tilde{\Sigma}_1 \times_{\tilde{J}} \tilde{\Sigma}_2$ is typically smaller than the time complexity of directly computing an abstraction $\tilde{\Sigma}$, the compositional approach is superior in this regard. However, existing compositional techniques result in abstractions that are more conservative than if directly computing an abstraction of Σ . This trade-off has to be carefully considered in the context of the specific system being controlled and the different techniques being used. Since the first investigations (Tazaki and Imura 2008; Reißig 2010) into compositionality in the context of finite abstractions, several techniques were developed including the use of feedback linearizability (Hussien et al. 2017), dissipative systems theory (Zamani and Arcak 2018), overlapping subsystems (Meyer et al. 2018), weak interaction between components (Mallik et al. 2019), and small gain principles (Pola et al. 2016; Swikir and Zamani 2019), among others.

Synthesis for Special Classes of Properties

Another research direction that has proved very useful is the use of specific structural properties of a system to simplify the construction of abstractions or the synthesis of controllers. Examples of such properties include large numbers of similar subsystems and mode counting specifications (Nilsson and Ozay 2016), monotonicity (Kim et al. 2017), and mixed monotonicity (Coogan and Arcak 2017). When specifications only restrict how many subsystems are in each mode of operation, termed mode counting specifications, the abstractions required for synthesis are simpler, thereby substantially reducing the complexity of synthesis. The authors of Nilsson and Ozay (2016) provide synthesis examples with thousands of subsystems. Monotonicity and mixed monotonicity are properties

of differential equations that can be leveraged for the same purpose of simplifying the construction of abstractions and reducing the complexity of synthesis.

Tools

The wide diversity of techniques used for synthesis is also reflected in the diversity of existing tools. Some are now of only historic significance, while others are still being actively maintained. An incomplete list includes PES-SOA (Mazo et al. 2010), LTLMoP (Finucane et al. 2010), TuLiP (Wongpiromsarn et al. 2011), CoSyMa (Mouelhi et al. 2013), SCOTS (Rungger and Zamani 2016), DSSynth (Abate et al. 2017), Sapo (Dreossi 2017), ROCS (Li and Liu 2018), and conPAS2 (Belta 2010).

Cross-References

- [Formal Methods for Controlling Dynamical Systems](#)
- [Validation and Verification Techniques and Tools](#)

Bibliography

- Abate A, Bessa I, Cattaruzza D, Chaves L, Cordeiro L, David C, Kesseli P, Kroening D, Polgreen E (2017) DSSynth: An automated digital controller synthesis tool for physical plants. In: 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE), pp 919–924
- Belta C (2010) conPAS2: temporal logic control of piecewise affine systems. <http://sites.bu.edu/hyness/conpas2/>. Accessed 23 Sept 2019
- Belta C, Habets LCGJM (2006) Controlling a class of nonlinear systems on rectangles. *IEEE Trans Autom Control* 51(11):1749–1759
- Belta C, Yordanov B, Gol EA (2017) Formal methods for discrete-time dynamical systems. Springer, Cham
- Caines PE, Wei YJ (1998) Hierarchical hybrid control systems: a lattice theoretic formulation. *IEEE Trans Autom Control* 43(4):501–508
- Cassandras CG, Lafortune S (2008) Introduction to discrete event systems. Springer, New York
- Clarke EM, Henzinger TA, Veith H, Bloem R (eds) (2018) Handbook of model checking. Springer, Cham
- Coogan S, Arcak M (2017) Finite abstraction of mixed monotone systems with discrete and continuous inputs. *Nonlinear Anal Hybrid Syst* 23:254–271
- Dreossi T (2017) Sapo: reachability computation and parameter synthesis of polynomial dynamical systems. In: Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control, HSCC'17. ACM, New York, pp 29–34
- Emerson EA (1990) Temporal and modal logic. In: Handbook of theoretical computer science, vol B. Elsevier Science, Amsterdam, pp 995–1072
- Finucane C, Jing G, Kress-Gazit H (2010) LTLMoP: experimenting with language, temporal logic and robot control. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 1988–1993
- Fřrstner D, Jung M, Lunze J (2002) A discrete-event model of asynchronous quantised systems. *Automatica* 38:1277–1286
- Girard A (2013) Low-complexity quantized switching controllers using approximate bisimulation. *Nonlinear Anal Hybrid Syst* 10:34–44. Special Issue Related to IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 12)
- Girard A, Pappas GJ (2007) Approximation metrics for discrete and continuous systems. *IEEE Trans Autom Control* 52(5):782–798
- Girard A, Pola G, Tabuada P (2010) Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Trans Autom Control* 55(1):116–126
- Girard A, Gřssler G, Mouelhi S (2016) Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models. *IEEE Trans Autom Control* 61(6):1537–1549
- Habets LCGJM, van Schuppen JH (2004) A control problem for affine dynamical systems on a full-dimensional polytope. *Automatica* 40(1):21–35
- Hsu K, Majumdar R, Mallik K, Schmuck A (2018) Lazy abstraction-based control for safety specifications. In: 2018 IEEE Conference on Decision and Control (CDC), pp 4902–4907
- Hussien O, Tabuada P (2018) Lazy controller synthesis using three-valued abstractions for safety and reachability specifications. In: 2018 IEEE Conference on Decision and Control (CDC), pp 3567–3572
- Hussien O, Ames A, Tabuada P (2017) Abstracting partially feedback linearizable systems compositionally. *IEEE Control Syst Lett* 1(2):227–232
- Kim ES, Arcak M, Seshia SA (2017) Symbolic control design for monotone systems with directed specifications. *Automatica* 83:10–19
- Kloetzer M, Belta C (2008) A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Trans Autom Control* 53(1):287–297
- Koutsoukos XD, Antsaklis PJ, Stiver JA, Lemmon MD (2000) Supervisory control of hybrid systems. *Proc IEEE* 88(7):1026–1049
- Kumar R, Garg VK (1995) Modeling and control of logical discrete event systems. Springer, Boston
- Li Y, Liu J (2018) Rocs: a robustly complete control synthesis tool for nonlinear dynamical systems. In: Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (Part of CPS Week), HSCC'18. ACM, New York, pp 130–135

- Liu J (2017) Robust abstractions for control synthesis: completeness via robustness for linear-time properties. In: Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control, HSCC'17. ACM, New York, pp 101–110
- Mallik K, Schmuck A, Soudjani S, Majumdar R (2019) Compositional synthesis of finite-state abstractions. *IEEE Trans Autom Control* 64(6):2629–2636
- Mazo M, Davitian A, Tabuada P (2010) PESSOA: a tool for embedded controller synthesis. In: Touili T, Cook B, Jackson P (eds) Computer aided verification. Springer, Berlin/Heidelberg, pp 566–569
- Meyer P, Girard A, Witrant E (2018) Compositional abstraction and safety synthesis using overlapping symbolic models. *IEEE Trans Autom Control* 63(6):1835–1841
- Moor T, Raisch J (2002) Abstraction based supervisory controller synthesis for high order monotone continuous systems. In: Engell S, Frehse G, Schnieder E (eds) Modelling, analysis, and design of hybrid systems. Springer, Berlin/Heidelberg, pp 247–265
- Mouelhi S, Girard A, Gössler G (2013) CoSyMA: a tool for controller synthesis using multi-scale abstractions. In: Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control, HSCC'13. ACM, New York, pp 83–88
- Nilsson P, Ozay N (2016) Control synthesis for large collections of systems with mode-counting constraints. In: Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control, HSCC'16. ACM, New York, pp 205–214
- Pola G, Tabuada P (2009) Symbolic models for nonlinear control systems: alternating approximate bisimulations. *SIAM J Control Optim* 48(2):719–733
- Pola G, Girard A, Tabuada P (2008) Approximately bisimilar symbolic models for nonlinear control systems. *Automatica* 44(10):2508–2516
- Pola G, Borri A, Di Benedetto MD (2012) Integrated design of symbolic controllers for nonlinear systems. *IEEE Trans Autom Control* 57(2):534–539
- Pola G, Pepe P, Di Benedetto MD (2016) Symbolic models for networks of control systems. *IEEE Trans Autom Control* 61(11):3663–3668
- Reiig G (2010) Abstraction based solution of complex attainability problems for decomposable continuous plants. In: 49th IEEE Conference on Decision and Control (CDC), Dec 2010, pp 5911–5917
- Reissig G, Weber A, Rungger M (2017) Feedback refinement relations for the synthesis of symbolic controllers. *IEEE Trans Autom Control* 62(4):1781–1796
- Rungger M, Stursberg O (2012) On-the-fly model abstraction for controller synthesis. In: 2012 American Control Conference (ACC), June 2012, pp 2645–2650
- Rungger M, Zamani M (2016) SCOTS: a tool for the synthesis of symbolic controllers. In: Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control, HSCC'16. ACM, New York, pp 99–104
- Stirling C (1992) Modal and temporal logics. In: Handbook of logic in computer science, vol 2. Oxford University Press, Oxford, pp 477–563
- Swikir A, Zamani M (2019) Compositional synthesis of finite abstractions for networks of systems: a small-gain approach. *Automatica* 107:551–561
- Tabuada P (2008) Controller synthesis for bisimulation equivalence. *Syst Control Lett* 57(6):443–452
- Tabuada P (2009) Verification and control of hybrid systems: a symbolic approach. Springer, New York
- Tabuada P, Pappas GJ (2006) Linear time logic control of discrete-time linear systems. *IEEE Trans Autom Control* 51(12):1862–1877
- Tazaki Y, Imura J-I (2008) Bisimilar finite abstractions of interconnected systems. In: Egerstedt M, Mishra B (eds) Hybrid systems: computation and control. Springer, Berlin/Heidelberg, pp 514–527
- Wongpiromsarn T, Topcu U, Ozay N, Xu H, Murray RM (2011) TuLiP: a software toolbox for receding horizon temporal logic planning. In: Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control, HSCC'11. ACM, New York, pp 313–314
- Zamani M, Arcak M (2018) Compositional abstraction for networks of control systems: a dissipativity approach. *IEEE Trans Control Netw Syst* 5(3):1003–1015
- Zamani M, Pola G, Mazo M, Tabuada P (2012) Symbolic models for nonlinear control systems without stability assumptions. *IEEE Trans Autom Control* 57(7):1804–1809

Controlling Collective Behavior in Complex Systems

Mario di Bernardo

University of Naples Federico II, Naples, Italy
Department of Electrical Engineering and ICT,
University of Naples Federico II, Napoli, Italy

Abstract

This entry presents a summarizing overview of the key problems in controlling complex network systems consisting of ensembles of interacting linear or nonlinear dynamical systems. After presenting a brief introduction and motivation to the problem, the main approaches to control the collective behavior of such ensembles are discussed together with the current open problems and challenges.

Keywords

Nonlinear control · Complex systems · Network control systems · Multi-agent systems · Control of complex networks

Introduction and Motivation

The traditional feedback paradigm where the output of a single system, e.g., the speed of a car or the level of water in a reservoir, needs to be adjusted as a function of its mismatch against some desired reference value has been increasingly challenged by modern applications. Examples include congestion control in communication networks, control of road traffic, energy balance and distribution in smart grids, and gene regulation in synthetic biological networks to name just a few (Newman et al. 2006; Strogatz 2001).

What makes all of these applications different from a traditional “plant” in control is their distributed, multi-agent nature and their ability to exhibit properties at the macroscopic level that cannot be predicted or analyzed by solely studying agents’ behavior at the microscopic level (Boccaletti et al. 2006; Chen et al. 2014). We name systems exhibiting emerging properties as “complex” to indicate that unfolding their behavior requires a multiscale approach bridging the gap between the scale at which agents operate (the microscopic level) with that where emerging properties arise (the macroscopic level).

A classical example of emerging property often observed in nature and technology is synchronization where an ensemble of interconnected (nonlinear) agents converges toward the same asymptotic solution (Pikovsky et al. 2001; Dörfler and Bullo 2014; Boccaletti et al. 2018). (This is related to the consensus problem in control where typically such a solution is some equilibrium point (Olfati-Saber et al. 2007); also see articles on ► “Consensus of Complex Multi-agent Systems” and ► “Averaging Algorithms and Consensus”). Synchronization is just an example of coordinated collective behavior. Other instances of relevance in applications are pattern formation, flocking behavior, platooning, cluster or group synchronization, etc.

In recent years, the need has become more and more pressing of devising feedback strategies to control the emergence of desired macroscopic properties and steer the collective behavior of

complex systems. This is due to epochal changes in traditional applications such as the transition from power grids to smart grids (Dobson 2013; Doerfler et al. 2013), the interest for new applications such as the control of synthetic biological systems (Del Vecchio et al. 2018), or the need to address pressing open challenges such as climate change, reducing congestion and fuel consumption in traffic applications, and controlling the behavior of flocks of drones or swarms of microrobots.

The challenge for control is to deal with large ensembles of dynamical systems, whose individual dynamics is often nonlinear, interacting with each other over a web of complex interconnections (Liu and Barabási 2016). As noted in the physics literature, such networks of interconnections can have complex structures and are often very far from being regular lattices (Newman 2003). Therefore, the structure of the graph describing the interconnections between the agents in the system plays an important role that cannot be neglected. Also the interconnections themselves between the agents can be time-varying, intermittent, affected by delays, and switched or described by nonlinear interaction functions (e.g., Meng et al. (2018) and Almeida et al. (2017)). Still in applications, collective behavior must be finely controlled, and coordination must be achieved in a distributed manner despite heterogeneities among the agents (think of biological applications!), disturbances, and noise.

Some crucial questions for control are therefore how to achieve, maintain, and steer the collective behavior of a complex network onto a desired solution, how to do so via distributed and decentralized estimation and control strategies, and, last but not least, how to formally prove convergence of the network toward the desired synchronous solution.

There is also a more fundamental challenge underlying all of the previous ones. Typically, the aim is to control the collective behavior of the complex system of interest at the macroscopic level by acting on the individual agents at the microscopic level. So how can we close the loop across these different scales?

The goal of this entry is to present a review of some existing solutions to these problems. Giving an exhaustive review of all the existing results is well beyond the scope of this contribution. Therefore, I have decided to focus on those approaches that rather than being specific to some application domain are more general in their formulation and can be proved formally to converge. Many other strategies have been proposed in the literature both in the control theoretic and physics communities. Exhaustive reviews can be found in many review papers and books in the literature (see, for example, Liu and Barabási 2016; Meyn 2007; Su and Wang 2013).

Modelling Complex Systems

To uncover their dynamics and control complex systems (also referred to as complex networks in the applied science literature), one needs to bring together three fundamental ingredients: (i) a model describing the dynamics of each individual agent in the ensemble, (ii) some communication (or interaction) protocol describing what information is being exchanged among neighboring agents, and (iii) the structure of the interconnections between different agents.

In a general framework, each agent can be thought of as a generic nonlinear control system of the form

$$\dot{x}_i = f_i(x_i) + g_i(x_i)u_i, \quad i = 1, \dots, N \quad (1)$$

where $x_i \in R^n$ is the state vector describing the agent dynamics and $u_i \in R^m$ is the input through which the agent communicates with its neighbors. Note that in consensus problems, the vector field is chosen so as to model networks of simple or higher-order integrators.

The interaction between neighboring agents can be modelled by means of an appropriate coupling law. The choice is application-dependent. A common choice, for example, in the consensus literature, is to assume linear diffusive coupling among the nodes, i.e., to set

$$u_i = \sigma \sum_{j=1}^N a_{ij}(x_j - x_i) \quad (2)$$

where σ is some common constant gain describing the strength of the interaction and a_{ij} is 1 if node i and node j are interconnected or 0 otherwise. Other models are also possible including delays, different gains associated with each link in the network, adaptive coupling laws, etc.

Finally, agents in the network communicate via some network structure encoding the topology of their interconnections. Networks associated with complex systems typically consist of many vertices (or nodes) linked together by relatively fewer edges (or links). The network structure can be represented by its associated graph, say $\mathcal{G}$, that using graph theoretic tools can be encoded by its corresponding adjacency matrix, $\mathcal{A}$, or more commonly via the graph Laplacian, $\mathcal{L}$ (for a review of the properties of these matrices, see Newman (2003) and references therein). To describe the network structures, we can use a set of characteristic features such as the degree k of a given vertex, defined as the number of its connections, the average degree, and the degree distributions. Different types of networks can then be identified such as random graphs, hierarchical networks, small-world networks, or scale-free networks (see Newman (2003) for an exhaustive review).

Often, in the literature on control of complex systems, some simplifying assumptions are made. In particular it is assumed that all agents are identical, i.e., $f_i = f_j = f$, that the interaction protocol is linear and diffusive, and that the structure of the network is static and undirected. Also, it is typically assumed that the input vector field $g_i(x)$ is the identity matrix and $m = n$. We will start by making the same assumptions and then discuss the implications of their relaxations and extensions proposed in the vast literature on controlling complex systems both from the control theoretic and the physics communities.

Under the assumptions mentioned above, the “open-loop” behavior of a multi-agent complex system can be described as

$$\dot{x}_i = f(x_i) + \sigma \sum_{j=1}^N \mathcal{L}_{ij} x_j, \quad i = 1, \dots, N \quad (3)$$

with $\mathcal{L}_{ij}$ being the elements of the network Laplacian describing the interconnection among nodes and where we have assumed linear diffusive coupling functions and nodes sharing identical nonlinear dynamics.

Controlling Collective Behavior

In general given a complex system of interest, because of its nature, control can be exerted via three different strategies or a combination of them. Specifically, control can be achieved by controlling (i) the network nodes or (ii) the edge dynamics (e.g., rendering the coupling function adaptive) or (iii) by varying in real time the network structure (e.g., switching on or off certain edges, rewiring the topology of the interconnections, etc.) (DeLellis et al. 2010). This opens a variety of possibilities to control the collective behavior of a complex system, with the latter approach being the most cumbersome and still little explored in the literature because of its computational complexity and the lack of a formal approach to prove convergence.

Here to illustrate some of the key aspects related to the control problem, we use the representative case of achieving synchronization in a network of the form (3) assuming that such convergence cannot be achieved in the absence of control or that one wishes to steer the synchronous behavior toward a desired solution. A typical example of such a scenario is that of ensembles of chaotic nonlinear systems, often used in the literature as a benchmark problem (Tang et al. 2014).

The crucial idea to achieve this goal is inspired from natural systems such as the heart tissue or swarms of honeybees where a relatively small number of agents in the ensemble take local control decisions that propagate to the entire group via the existing interconnections among nodes in the network. In the heart tissue, for instance, the so-called pacemaker cells are instructed to

increase or decrease the beating frequency. They are a relatively smaller number of all the cells in the tissue, but as information propagates to neighboring cells, the synchronous beating of the heart at the desired frequency is attained. Similarly, in a swarm of honeybees relocating from one location to another, a small number of *explorer bees* scout for the new location and, once decided, communicate its position to other individuals in the swarm causing information to propagate in the social network and the whole swarm to reach the desired site (Chen et al. 2014). *Pinning control* is the technique first proposed in Wang and Chen (2002) (also see the more recent review in Wang and Su 2014) that mimics this scenario in a feedback control setting. In particular, the idea consists in monitoring and controlling the dynamics of a relatively small fraction of the nodes in the network at the microscopic level to achieve synchronization at the macroscopic level by leveraging the existing interconnections among nodes in the network.

This is achieved by introducing a *master* node whose dynamics is not affected by that of the other nodes in the network and interconnecting it to a (small) fraction of other network nodes. In so doing, agents in the network are split into two communities: those that are directly reached by the control action (also known as *pinned* nodes) and those that are not.

The closed loop network equations can then be written in the simplest case as

$$\dot{x}_i = f(x_i) + \sigma \sum_{j=1}^N \mathcal{L}_{ij} x_j + u_i, \quad i = 1, 2, \dots, M \quad (4)$$

$$\dot{x}_i = f(x_i) + \sigma \sum_{j=1}^N \mathcal{L}_{ij} x_j, \quad i = M + 1, \dots, N \quad (5)$$

where, with a slight abuse of notation, u_i is now the input through which the master node can influence the dynamics of the pinned nodes. Note that typically $M \ll N$ as the number of pinned

nodes is much smaller than the total number of agents in the network (often less than 10~20%).

The control strategy to close the loop can be chosen in a number of different ways, and many approaches have been proposed in the literature since the first appearance of the idea back in 2002. From state feedback control to adaptive, robust, switched, and hybrid control, almost all techniques have been used and tested to achieve pinning control of a complex system (for a recent review, see Wang and Su 2014). Just to illustrate one example, we take the simplest state feedback pinning control law given by setting

$$u_i = \sigma k_i (s(t) - x_i(t))$$

where $s(t)$ is the desired asymptotic collective behavior we wish the complex system to achieve and k_i is a set of control gains determining the strength of the control action on the nodes being controlled.

With this choice of control law, the closed loop equations of the complex system become

$$\dot{x}_i = f(x_i) + \sigma \left[\sum_{j=1}^N \mathcal{L}_{ij} x_j + \delta_i k_i (s(t) - x_i(t)) \right], \quad i = 1, 2, \dots, N \quad (6)$$

with δ_i being equal to 1 if node i is pinned, 0 otherwise.

A number of key questions now arise. First and foremost, how many and what nodes should be picked as those that are directly controlled? For example, do we aim at controlling the large degree nodes in the network (the hubs) or those nodes with fewer links in the network periphery? What fraction of the nodes in the network must be controlled in order to achieve convergence? Also new technical issues arise, for example, selecting the best control strategy to be used given the nonlinear dynamics of the nodes or finding the right theoretical framework to prove convergence and, if possible, estimate transient performance such as the rate of convergence toward the synchronous solution (e.g., Zhang et al. 2011).

These problems are related to the problem of studying controllability of a complex system as highlighted in Liu et al. (2011), a paper that made the cover of an issue of *Nature* in 2011. To summarize, the problem is to determine whether a complex system of interest can be controlled by exerting the control action only on a number of nodes and identify what and how many nodes to control. This problem is still an open problem in the most general case of networks of nonlinear dynamical systems and, the existing approaches can be mostly classified into two categories: those that aim at deriving graph theoretic conditions on the network structure, given the dynamics, that guarantee controllability of the associated complex system (e.g., Mesbahi and Egerstedt 2010; Liu et al. 2011) and those, often based on local approaches, that aim at finding conditions on the number of nodes and the strength of the coupling and control gains that guarantee convergence to the synchronous solution (e.g., Sorrentino et al. (2007) and Porfiri and di Bernardo (2008)). The methods proposed in Mesbahi and Egerstedt (2010) and Liu et al. (2011) and later extensions achieve the former goal, while the concept of pinning controllability first presented in Sorrentino et al. (2007) the latter. While the former set of tools relies on graph theoretic methods and the use of the theory of structural controllability, pinning controllability is studied using an extension of the semi-analytical approach known as the master stability function (MSF) in the physics literature (Pecora and Carroll 1998, 2015). Later extensions based on Lyapunov-based arguments were presented in Porfiri and di Bernardo (2008).

In all cases, it is often assumed that the network structure is static, that the node dynamics is linear or if nonlinear that it can be linearized, and that the controlled nodes can be selected at will among all nodes in the network (when this is not the case, the concept of partial controllability should be used instead; see, for example, (DeLellis et al. 2018)). So far, more complex scenarios have only been analyzed numerically. A thorough review of results on controllability of complex multi-agent systems is beyond the scope

of this entry (for further information, see Liu and Barabási 2016).

Despite their limitations, the existing tools allowed the analysis of a number of real-world scenarios finding evidence that in general only a relatively small fraction of nodes needs to be controlled and that, strikingly, in many natural and technological networks *control avoids the hubs* but tends to be localized on nodes with smaller degrees (Liu et al. 2011). This result was obtained by only looking at the structure of the graph assuming linear dynamics at the nodes. Recent studies of controllability problems for complex network systems have highlighted the importance of taking into account the control energy required to attain control and the nonlinear dynamics of the nodes (see, for instance, the results presented in Pasqualetti et al. (2014) and the recent paper by Jiang and Lai (2019)).

Still many open problems remain. For example, how to analyze controllability of time-varying network structures, or complex systems characterized by nonlinear heterogeneous systems, or those affected by noise or disturbances. Also, the problem of observability of complex systems still deserves further attention (e.g., Liu et al. 2013).

Evolving Networks and Adaptive Complex Systems

As discussed so far, the earlier models of complex systems neglected some important aspects in applications such as that agents are typically nonidentical, that the strength of the coupling and the control gains might not be the same for all edges/nodes in the network, and that the topology of the interconnections is fixed (and selected off-line). Moreover, noise or external disturbances are hardly taken into account. The pioneering work of Siljak on dynamic graphs (Siljak 1978, 2008) in the early 1970s needs to be mentioned here as one of the earlier attempts to formalize the idea of complex systems with dynamical nodes and edges (but a fixed structure) (Zecevic and Siljak 2010). Around the same time, J. Holland proposed the idea in information science of a

complex adaptive system as one adapting both its structure and dynamics to respond to environmental stimuli (Holland 1975). More recently, the concepts of adaptive networks (Gross and Sayama 2009) and evolving dynamical networks (Gorochowski et al. 2010) were proposed as a possible way forward to incorporate both aspects.

From the control viewpoint, the presence of dynamical links allows the design of distributed and decentralized adaptive strategies to control the system collective behavior, for example, toward synchronization. In this case, the network structure is still fixed, but the edges are endowed with their own adaptive dynamics which is chosen as a function of the mismatch between the dynamics of the agents they interconnect. Namely, in its simplest form, an adaptive network can be written as

$$\dot{x}_i = f(x_i) + \left[\sum_{j=1}^N \sigma_{ij}(t) \mathcal{L}_{ij} x_j \right] \quad (7)$$

where the coupling gains are now modelled by the set of time-varying functions $\sigma_{ij}(t)$ which can be selected by appropriate adaptive laws (see, for example, the strategies presented in DeLellis et al. (2009a), Lu et al. (2006), Yu et al. (2012), Das and Lewis (2012), and Kim and De Persis (2017)).

Adaptive strategies together with distributed estimation strategies of macroscopic network observables can also be used, as recently shown in Kempton et al. (2017a, b), to close the loop across scales and achieve control of macroscopic variables, such as the network connectivity or synchronizability, by acting on the edges at the microscopic level. It is also possible to consider adaptive pinning strategies able to self-tune the control and coupling gains among nodes required to achieve control (see Kempton et al. 2017a, b).

A further step is to endow the network structure with the ability of rewiring itself dynamically or evolving by adding or removing some links in order to achieve the desired control goal. This, of course, is a much harder problem that can be solved under restricting assumptions by performing off-line or online numerical optimization

(e.g., Gorochowski et al. 2010), but it is much harder to study analytically. An example of evolving strategies for controlling the collective behavior of complex systems is the *edge-snapping control technique* first proposed in DeLellis et al. (2009b). Here, the network has the same structure as in (7), but the edge dynamics is given by a nonlinear bistable dynamics driven by some function, say g , of the mismatch between the nodes at its endpoints, i.e.,

$$\ddot{\sigma}_{ij} + d\dot{\sigma}_{ij} + \frac{d}{d\sigma_{ij}} V(\sigma_{ij}) = g(x_i, x_j) \quad (8)$$

with V being a bistable potential function and d a tuning parameter that can be used to control the local transient properties of the edge dynamics.

Other more advanced approaches for the control of collective behavior in complex systems include those based on the use of multilayer and multiplex control structures such as the distributed PI strategy presented in Burbano-Lombana and di Bernardo (2015) and Lombana and Di Bernardo (2016) where the control layers implementing proportional and integral coupling among nodes can have different structures that can be exploited to enhance control performance. Other recently proposed methods include those in Scardovi et al. (2010), Seyboth et al. (2016), Liuzza et al. (2016), Montenbruck et al. (2015), and Yang and Lu (2015).

Proving Convergence

When controlling complex systems, a pressing open problem is how to prove convergence toward the desired collective behavior and certify stability, possibly globally, of the entire ensemble of interacting dynamical systems. An exhaustive review of the existing results is beyond the scope of this entry, so here I will just pinpoint some of the key approaches presented in the literature highlighting their differences.

Early attempts to prove convergence to coordinated collective behavior in networks goes back to the idea of using local methods to prove transversal stability of the synchronization

manifold. These methods are based on the master stability function (MSF) approach first presented in Pecora and Carroll (1998) which is based on linearization and block diagonalization of the network equations and the computation of the MSF defined as the magnitude of the largest Lyapunov exponent expressed as a function of a parameter encompassing information on both the network structure and the agent dynamics. Despite being a semi-analytical approach (the computation of the MSF requires the use of numerical algorithms to compute Lyapunov exponents), this method has the merit of returning conditions combining the strength of the coupling, the network structure (through the Laplacian eigenvalues), and the agent dynamics (Pecora and Carroll 2015).

Another approach often used in control theory is the use of Lyapunov-based tools yielding global but often extremely conservative results. The use of Lyapunov methods allows to deal with nonlinear vector fields, but additional conditions are required such as the QUAD condition which assumes the vector field verifies a sort of one-sided Lipschitz condition (for further details, see DeLellis et al. 2011). Other methods based on a variety of nonlinear control approaches can be found in Wieland et al. (2011), Andrieu et al. (2018), and Khong et al. (2016).

An alternative strategy to proving asymptotic stability is the use of contraction theory (Lohmiller and Slotine 1998) as explained in Slotine et al. (2004). This is particularly useful for certain types of complex systems such as biological systems where matrix measures other than the Euclidean one can be used to study the problem (Russo and Slotine 2010; Russo et al. 2010). A review with further pointers to other references can be found in di Bernardo et al. (2016).

Summary, Open Problems, and Future Directions

We have seen that many complex systems in applications can be modelled as large networks of interconnected (nonlinear) dynamical systems. The model consists of three key ingredients: a

model of the agent dynamics, a function describing the nature of the interaction among agents, and a graph capturing the structure of the interconnections among the nodes. Such systems are able to exhibit emergent properties such as synchronization, the simplest type of coordinated collective behavior often observed in natural and artificial systems.

To control the collective behavior of a complex system, we can act on each of the three aspects it comprises: we can control the nodes, the edges, or the structure itself of the node interconnections. In this entry, we reviewed some of the key approaches, particularly those based on steering the dynamics of a small fraction of the agents in the system to achieve global, macroscopic effects. We reviewed some of the existing implementations of pinning control and discussed their limitations and possible extensions.

Despite current advances, controlling collective behavior of complex systems remains a pressing open problem in many application areas, specifically, those where heterogeneity abounds such as biological networks or the network structure changes in time such as in robotics, power grids, or animal behavior. Moreover, uncertainties and noise are unavoidable in applications, and communication among nodes might take place over all sort of different media (Wi-Fi, cloud, mobile networks, etc.). Then a key challenge is how to design control strategies able to endow the system with the ability to self-organize, rewire the network of interconnections between agents in order to cope with noise, accommodate for intermittent communications or packet drops, and achieve the desired control goal. Also recently, it has been pointed out that, intriguingly, noise can be beneficial to control the collective behavior of a complex system (see, for example, Russo and Shorten 2018 and Burbano-L et al. 2017).

Ultimately, the goal is to go beyond synchronization toward other more complex types of coordinated collective behavior and devise strategies to mimic natural complex systems, such as school of fish or flocks of birds, that are able to perform collective maneuvers at relatively high speeds with limited communication and in the absence of any centralized intelligence.

These systems are characterized by the ability to evolve and self-organize, changing their structure to accommodate environmental changes. A hurdle for control system design is the lack of a theoretical framework to prove convergence in these scenarios, particularly those where the agents are strongly heterogeneous or stochastic, noise is present, and the agents are strongly nonlinear or even hybrid or discontinuous.

This is an area full of opportunities for exciting research. If we succeed, we can mimic natural systems and design self-organizing, robust, and flexible control strategies for complex multi-agent systems. In so doing, we will be able to take control into new emerging applications such as those coming from biology and health or swarm robotics where thousands of individual cells or microrobots need to finely coordinate their collective behavior in order to achieve the desired control goal.

Cross-References

- [Averaging Algorithms and Consensus](#)
- [Consensus of Complex Multi-agent Systems](#)

Bibliography

- Almeida J, Silvestre C, Pascoal A (2017) Synchronization of multiagent systems using event-triggered and self-triggered broadcasts. *IEEE Trans Autom Control* 62(9):4741–4746
- Andrieu V, Jayawardhana B, Tarbouriech S (2018) Some results on exponential synchronization of nonlinear systems. *IEEE Trans Autom Control* 63(4):1213–1219
- Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang DU (2006) Complex networks: structure and dynamics. *Phys Rep* 424(4–5):175–308
- Boccaletti S, Pisarchik AN, Del Genio CI, Amann A (2018) Synchronization: from coupled systems to complex networks. Cambridge University Press
- Burbano-L DA, Russo G, di Bernardo M (2017) Pinning controllability of complex stochastic networks. In: *IFAC World Congress, Toulouse*, vol 50, pp 8327–8332
- Burbano-Lombana DA, di Bernardo M (2015) Distributed PID control for consensus of homogeneous and heterogeneous networks. *IEEE Trans Control Netw Syst* 2(2):154–163
- Chen G, Wang X, Li X (2014) Fundamentals of complex networks: models, structures and dynamics. Wiley

- Das A, Lewis FL (2012) Distributed adaptive control for synchronization of unknown nonlinear networked systems. *Automatica* 46(12):2014–2021
- DeLellis P, diBernardo M, Garofalo F (2009a) Novel decentralized adaptive strategies for the synchronization of complex networks. *Automatica* 45(5):1312–1319
- DeLellis P, diBernardo M, Garofalo F, Porfiri M (2009b) Evolution of complex networks via edge snapping. *IEEE Trans Circuits Syst I* 57(8):2132–2143. <https://doi.org/10.1109/TCSI.2009.2037393>
- DeLellis P, diBernardo M, Gorochowski TE, Russo G (2010) Synchronization and control of complex networks via contraction, adaptation and evolution. *IEEE Circuits Syst Mag* 10(3):64–82
- DeLellis P, diBernardo M, Russo G (2011) On QUAD, lipschitz, and contracting vector fields for consensus and synchronization of networks. *IEEE Trans Circuits Syst I Reg Pap* 58(3):576–583
- DeLellis P, Garofalo F, Iudice FL (2018) The partial pinning control strategy for large complex networks. *Automatica* 89:111–116
- Del Vecchio D, Qian Y, Murray RM, Sontag ED (2018) Future systems and control research in synthetic biology. *Ann Rev Control* 45:5–17
- diBernardo M, Fiore D, Russo G, Scafuti F (2016) Convergence, consensus and synchronization of complex networks via contraction theory. In: *Complex systems and networks*. Springer, pp 313–339
- Dobson I (2013) Complex networks: synchrony and your morning coffee. *Nat Phys* 9(3):133
- Doerfler F, Chertkov M, Bullo F (2013) Synchronization in complex oscillator networks and smart grids. In: *Proc Natl Acad Sci* 110(6):2005–2010
- Dörfler F, Bullo F (2014) Synchronization in complex networks of phase oscillators: a survey. *Automatica* 50(6):1539–1564
- Gorochowski TE, diBernardo M, Grierson CS (2010) Evolving enhanced topologies for the synchronization of dynamical complex networks. *Phys Rev E* 81(5):056212. <https://doi.org/10.1103/PhysRevE.81.056212>
- Gross T, Sayama H (2009) *Adaptive networks: theory, models and applications*. Springer, p 332
- Holland JH (1975) *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor
- Jiang J, Lai YC (2019) Irrelevance of linear controllability to nonlinear dynamical networks. *Nat Commun* 10(1):1–10
- Kempton L, Herrmann G, diBernardo M (2017a) Distributed optimisation and control of graph Laplacian eigenvalues for robust consensus via an adaptive multilayer strategy. *Int J Robust Nonlinear Control* 27(9):1499–1525
- Kempton L, Herrmann G, diBernardo M (2017b) Self-organization of weighted networks for optimal synchronizability. *IEEE Trans Control Netw Syst* 5(4):1541–1550
- Khong SZ, Lovisari E, Rantzer A (2016) A unifying framework for robust synchronization of heterogeneous networks via integral quadratic constraints. *IEEE Trans Autom Control* 61(5):1297–1309
- Kim H, De Persis C (2017) Adaptation and disturbance rejection for output synchronization of incrementally output–feedback passive systems. *Int J Robust Nonlinear Control* 27(17):4071–4088
- Liu YY, Barabási AL (2016) Control principles of complex systems. *Rev Modern Phys* 88(3):035006
- Liu YY, Slotine JJ, Barabási AL (2011) Controllability of complex networks. *Nature* 473:167–173
- Liu YY, Slotine JJ, Barabási AL (2013) Observability of complex systems. *Proc Natl Acad Sci* 110(7):2460–2465
- Liuzza D, Dimarogonas DV, DiBernardo M, Johansson KH (2016) Distributed model based event-triggered control for synchronization of multi-agent systems. *Automatica* 73:1–7
- Lohmiller W, Slotine JJ (1998) On contraction analysis for nonlinear systems. *Automatica* 34(6):683–696
- Lombana DAB, DiBernardo M (2016) Multiplex pi control for consensus in networks of heterogeneous linear agents. *Automatica* 67:310–320
- Lu JA, Lü J, Zhou J (2006) Adaptive synchronization of an uncertain complex dynamical network. *IEEE Trans Autom Control* 51(4):652–656
- Meng Z, Yang T, Li G, Ren W, Wu D (2018) Synchronization of coupled dynamical systems: tolerance to weak connectivity and arbitrarily bounded time-varying delays. *IEEE Trans Autom Control* 63(6):1791–1797
- Mesbahi M, Egerstedt M (2010) *Graph theoretic methods in multiagent networks*, vol 33. Princeton University Press
- Meyn S (2007) *Control techniques for complex networks*. Cambridge University Press, Cambridge
- Montenbruck JM, Bürger M, Allgöwer F (2015) Practical synchronization with diffusive couplings. *Automatica* 53:235–243
- Newman MEJ (2003) The structure and function of complex networks. *SIAM Rev* 45(2):167–256
- Newman MEJ, Barabási AL, Watts DJ (2006) *The structure and dynamics of networks*. Princeton University Press, Princeton
- Olfati-Saber R, Fax AA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. *Proc IEEE* 95(1):215–233
- Pasqualetti F, Zampieri S, Bullo F (2014) Controllability metrics, limitations and algorithms for complex networks. *IEEE Trans Control Netw Syst* 1(1):40–52
- Pecora LM, Carroll TL (1998) Master stability functions for synchronized coupled systems. *Phys Rev Lett* 80:2109–2112
- Pecora L, Carroll T (2015) Synchronization of chaotic systems. *Chaos* 25(9):097611
- Pikovsky A, Rosenblum M, Kurths J (2001) *Synchronization: a universal concept in nonlinear science*. Cambridge University Press, Cambridge
- Porfiri M, diBernardo M (2008) Criteria for global pinning-controllability of complex networks. *Automatica* 44(12):3100–3106
- Russo G, Shorten R (2018) On common noise-induced synchronization in complex networks with

- state-dependent noise diffusion processes. *Phys D Nonlinear Phenom* 369:47–54
- Russo G, Slotine J (2010) Global convergence of quorum-sensing networks. *Phys Rev E* 82:041919
- Russo G, diBernardo M, Sontag ED (2010) Global entrainment of transcriptional systems to periodic inputs. *PLoS Comput Biol* 6:e1000739
- Scardovi L, Arcak M, Sontag ED (2010) Synchronization of interconnected systems with applications to biochemical networks: an input-output approach. *IEEE Trans Autom Control* 55(6):1367–1379
- Seyboth GS, Ren W, Allgöwer F (2016) Cooperative control of linear multi-agent systems via distributed output regulation and transient synchronization. *Automatica* 68:132–139
- Siljak DD (1978) *Large-scale dynamic systems: stability and Structure*. North Holland, New York
- Siljak D (2008) *Dynamic graphs. Nonlinear analysis: hybrid systems*
- Slotine JJE, Wang W, Rifai KE (2004) Contraction analysis of synchronization of nonlinearly coupled oscillators. In: *International symposium on mathematical theory of networks and systems*
- Sorrentino F, diBernardo M, Garofalo F, Chen G (2007) Controllability of complex networks via pinning. *Phys Rev E* 75(4):046103
- Strogatz SH (2001) Exploring complex networks. *Nature* 410:268–276
- Su H, Wang X (2013) *Pinning control of complex networked systems: synchronization, consensus and flocking of networked systems via pinning*. Springer
- Tang Y, Qian F, Gao H, Kurths J (2014) Synchronization in complex networks and its applications: a survey of recent advances and challenges. *Ann Rev Control* 38(2):184–198
- Wang XF, Chen G (2002) Pinning control of scale-free dynamical networks. *Phys A Stat Mech Appl* 310(3–4):521–531
- Wang X, Su H (2014) Pinning control of complex networked systems: a decade after and beyond. *Ann Rev Control* 38(1):103–111
- Wieland P, Sepulchre R, Allgöwer F (2011) An internal model principle is necessary and sufficient for linear output synchronization. *Automatica* 47(5):1068–1074
- Yang X, Lu J (2015) Finite-time synchronization of coupled networks with Markovian topology and impulsive effects. *IEEE Trans Autom Control* 61(8):2256–2261
- Yu W, DeLellis P, Chen G, di Bernardo M, Kurths J (2012) Distributed adaptive control of synchronization in complex networks. *IEEE Trans Autom Control* 57(8):2153–2158
- Zecevic A, Siljak D (2010) Control of dynamic graphs. *SICE J Control Meas Syst Integration* 2(1):001–009
- Zhang H, Lewis FL, Das A (2011) Optimal design for synchronization of cooperative systems: state feedback, observer and output feedback. *IEEE Trans Autom Control* 56(8):1948–1952

Cooperative Manipulators

Fabrizio Caccavale

School of Engineering, Università degli Studi della Basilicata, Viale dell'Ateneo Lucano 10, Potenza, Italy

Abstract

This chapter presents an overview of the main modeling and control issues related to cooperative manipulation of a common object by means of two or more robotic manipulators. A historical path is followed to present the main research results on cooperative manipulation. Kinematics and dynamics of robotic arms cooperatively manipulating a tightly grasped rigid object are briefly discussed. Then, the main strategies for force/motion control of cooperative systems are briefly reviewed.

Keywords

Cooperative task space · Coordinated motion · Force/motion control · Grasping · Manipulation · Multi-arm systems

Introduction

Since the early 1970s, it has been recognized that many tasks, which are difficult or even impossible to execute by a single robotic manipulator, become feasible when two or more manipulators work in a cooperative way. Examples of typical cooperative tasks are the manipulation of heavy and/or large payloads, assembly of multiple parts, and handling of flexible and articulated objects. (see Fig. 1 for an example of cooperative robotic work cell).

In the 1980s research achieved several theoretical results related to modeling and control of single-arm robots; this further fostered research on multi-arm robotic systems. Dynamics and control as well as force control issues have been widely explored along the decade.

Cooperative

Manipulators, Fig. 1 An example of a cooperative robotic work cell composed by two industrial robot arms



C

In the 1990s parametrization of the constraint forces/moments acting on the object has been recognized as a key to solving control problems and has been studied in several papers (see, e.g., contributions in Uchiyama and Dauchez 1993; Walker et al. 1991; Williams and Khatib 1993, and Sang et al. 1995). Several control schemes for cooperative manipulators based on the sought parameterizations have been designed, including force/motion control (Wen and Kreutz-Delgado 1992) and impedance control (Schneider and Cannon 1992; Bonitz and Hsia 1996). Other approaches are adaptive control (Hu et al. 1995), kinematic control (Chiacchio et al. 1996), task-space regulation (Caccavale et al. 2000), and model-based coordinated control (Hsu 1993). Other important topics investigated in the 1990s were the definition of user-oriented task-space variables for coordinated control (Chiacchio et al. 1996; Caccavale et al. 2000), the development of meaningful performance measures (Chiacchio et al. 1991a,b) for multi-arm systems, and the problem of load sharing (Walker et al. 1989).

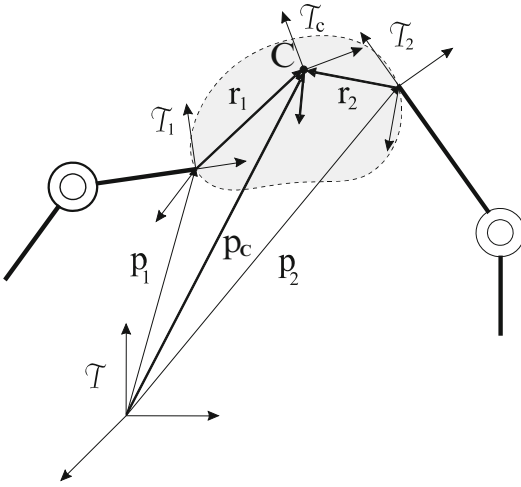
Most of the abovementioned works assume that the cooperatively manipulated object is rigid and tightly grasped. However, since 1990s, several research efforts have been focused on the control of cooperative flexible manipulators (Yamano et al. 2004), since flexible-arm robots merits (lightweight structure, intrinsic compliance, and hence safety) can be conveniently exploited in cooperative manipulation. Other research efforts have been focused

on the control of cooperative systems for the manipulation of flexible objects (Yukawa et al. 1996) and, more recently, on the cooperative transportation and manipulation of objects via multiple mobile robotic system (Khatib et al. 1996; Sugar and Kumar 2002; Bai and Wen 2010).

Modeling, Load Sharing, and Performance Evaluation

The first modeling goal is the definition of suitable variables describing the kinetostatics of a cooperative system. Hereafter, the main results available are summarized for a dual-arm system composed by two cooperative manipulators grasping a common object.

The kinetostatic formulation proposed by Uchiyama and Dauchez (1993), i.e., the so-called symmetric formulation, is based on kinematic and static relationships between generalized forces/velocities acting at the object and their counterparts acting at the manipulators end effectors. To this aim, the concept of *virtual stick* is defined as the vector which determines the position of an object-fixed coordinate frame with respect to the frame attached to each robot end effector (Fig. 2). When the object grasped by the two manipulators can be considered rigid and tightly attached to each end effector, then the virtual stick behaves as a rigid stick fixed to each end effector. According to the



Cooperative Manipulators, Fig. 2 Grasp geometry for a two-manipulator cooperative system manipulating a common object. The vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ are the *virtual sticks*, $\mathcal{T}_c$ is the coordinate frame attached to the object, and $\mathcal{T}_1$ and $\mathcal{T}_2$ are the coordinate frames attached to each end effector

symmetric formulation, the vector, $\mathbf{h}$, collecting the generalized forces (i.e., forces and moments) acting at each end effector is given by:

$$\mathbf{h} = \mathbf{W}^\dagger \mathbf{h}_E + \mathbf{V} \mathbf{h}_I, \quad (1)$$

where $\mathbf{W}$ is the so-called grasp matrix; the columns of $\mathbf{V}$ span the null space of the grasp matrix; $\mathbf{h}_I$ is the generalized force vector which does not contribute to the object's motion, i.e., it represents internal loading of the object (mechanical stresses) and is termed as *internal forces*; while $\mathbf{h}_E$ represents the vector of *external forces*, i.e., forces and moments causing the object's motion. Later, a *task-oriented formulation* has been proposed (Chiacchio et al. 1996), aimed at defining a *cooperative task space* in terms of *absolute* and *relative* motion of the cooperative system, which can be directly computed from the position and orientation of the end-effector coordinate frames.

The dynamics of a cooperative multi-arm system can be written as the dynamics of the single manipulators together with the closed-chain constraints imposed by the grasped object.

By eliminating the constraints, a reduced-order model can be obtained (Koivo and Unseren 1991).

Strongly related to kinetostatics and dynamics of cooperative manipulators is the load sharing problem, i.e., distributing the load among the arms composing the system (Walker et al. 1989). A very relevant problem related to the load sharing is that of robust holding, i.e., the problem of determining forces/moments applied to object by the arms, in order to keep the grasp even in the presence of disturbing forces/moments (Uchiyama and Yamashita 1991).

A major issue in robotic manipulation is the performance evaluation via suitably defined indexes (e.g., manipulability ellipsoids). These concepts have been extended to multi-arm robotic systems in Chiacchio et al. (1991a,b). Namely, by exploiting the kinetostatic formulations described above, velocity and force manipulability ellipsoids can be defined, by regarding the whole cooperative system as a mechanical transformer from the joint space to the cooperative task space. The manipulability ellipsoids can be seen as performance measures aimed at determining the attitude of the system to cooperate in a given configuration.

Finally, it is worth mentioning the strict relationship between problems related to grasping of objects by fingers/hands and those related to cooperative manipulation. In fact, in both cases, multiple manipulation structures grasp a commonly manipulated object. In multifingered hands only some motion components are transmitted through the contact point to the manipulated object (unilateral constraints), while cooperative manipulation via robotic arms is achieved by rigid (or near-rigid) grasp points, and interaction takes place by transmitting all the motion components through the grasping points (bilateral constraints). While many common problems between the two fields can be tackled in a conceptually similar way (e.g., kinetostatic modeling, force control), many others are specific of each of the two application fields (e.g., form and force closure for multifingered hands).

Control

When a cooperative multi-arm system is employed for the manipulation of a common object, it is important to control both the absolute motion of the object and the internal stresses applied to it. Hence, most of the control approaches to cooperative robotic systems can be classified as force/motion control schemes.

Early approaches to the control of cooperative systems were based on the *master/slave* concept. Namely, the cooperative system is decomposed in a position-controlled master arm, in charge of imposing the absolute motion of the object, and the force-controlled slave arms, which are to follow (as smoothly as possible) the motion imposed by the master. A natural evolution of the above-described concept has been the so-called *leader/follower* approach, where the follower arms reference motion is computed via closed-chain constraints. However, such approaches suffered from implementation issues, mainly due to the fact that the compliance of the slave arms has to be very large, so as to smoothly follow the motion imposed by the master arm. Moreover, the roles of master and slave (leader and follower) may need to be changed during the task execution.

Due to the abovementioned limitations, more natural non-master/slave approaches have been pursued later, where the cooperative system is seen as a whole. Namely, the reference motion of the object is used to determine the motion of all the arms in the system, and the interaction forces are measured and fed back so as to be directly controlled. To this aim, the mappings between forces and velocities at the end effector of each manipulator and their counterparts at the manipulated object are considered in the design of the control laws.

An approach, based on the classical hybrid force/position control scheme, has been proposed in Uchiyama and Dauchez (1993), by exploiting the symmetric formulation described in the previous section.

In Wen and Kreutz-Delgado (1992) a Lyapunov-based approach is pursued to devise force/position PD-type control laws. Namely, the

joints torques inputs to each arm are computed as the combination of two contributions:

$$\tau_m = \tau_p + \tau_h, \quad (2)$$

where τ_p is a PD-type term (eventually including a feedback/feedforward model-based compensation term), taking care of position control, while τ_h is in charge of internal forces/moments control. This approach has been extended in Caccavale et al. (2000), where kinetostatic filtering of the control action is performed, so as to eliminate all the components of the control input which contribute to internal stresses at the object.

A further improvement of the PD plus gravity compensation control approach has been achieved by introducing a full model compensation, so as to achieve feedback linearization of the closed loop system. The feedback linearization approach formulated at the operational space level is the base of the so-called *augmented object* approach (Sang et al. 1995). In this approach the system is modeled in the operational space as a whole, by suitably expressing its inertial properties via a single augmented inertia matrix M_O , i.e.,

$$M_O(x_E)\ddot{x}_E + c_O(x_E, \dot{x}_E) + g_O(x_E) = h_E. \quad (3)$$

where M_O , c_O , and g_O are the operational space terms modeling, respectively, the inertial properties of the whole system (manipulators and object); the Coriolis, centrifugal, and friction terms; and the gravity terms, while x_E is the operational space vector describing the position and orientation of the coordinate frame attached to the grasped object. In the framework of feedback linearization (formulated in the operational space), the problem of controlling the internal forces can be solved, e.g., by resorting to the *virtual linkage* model (Williams and Khatib 1993) or according to the scheme proposed in Hsu (1993), i.e.,

$$\tau_m = J^T W^\dagger [M_O(\ddot{x}_{E,d} + K_v \dot{e}_E + K_p e_E) + c_O + g_O] + J^T V \left(h_{I,d} + K_h \int e_{h,I} \right), \quad (4)$$

where the subscript “d” denotes the desired variables; $\mathbf{e}_E$ is the tracking error computed in terms of object’s position/orientation variables; $\mathbf{e}_{h,i}$ is the internal force tracking error; $\mathbf{K}_p$, $\mathbf{K}_v$, and $\mathbf{K}_h$ are positive-definite matrix gains; and $\mathbf{J}$ is a block-diagonal matrix collecting the individual Jacobian matrices of the manipulators; the above control law yields a linear and decoupled closed-loop dynamics, such that motion and force tracking errors asymptotically vanish.

An alternative control approach is based on the well-known impedance concept (Schneider and Cannon 1992; Bonitz and Hsia 1996). In fact, when a manipulation system interacts with an external environment and/or other manipulators, large values of the contact forces and moments can be avoided by enforcing a compliant behavior with suitable dynamic features. In detail, the following mechanical impedance behavior between the object displacements and the forces due to the object-environment interaction can be enforced (*external impedance*):

$$\mathbf{M}_E \tilde{\mathbf{a}}_E + \mathbf{D}_E \tilde{\mathbf{v}}_E + \mathbf{K}_E \mathbf{e}_E = \mathbf{h}_{env}, \quad (5)$$

where $\mathbf{e}_E$ represents the vector of displacements between object’s desired and actual pose, $\tilde{\mathbf{v}}_E$ is the difference between the object’s desired and actual velocities, $\tilde{\mathbf{a}}_E$ is the difference between the object’s desired and actual accelerations, and $\mathbf{h}_{env}$ is the generalized force acting on the object, due to the interaction with the environment. The impedance dynamics is characterized in terms of given positive definite mass, damping, and stiffness matrices ($\mathbf{M}_E$, $\mathbf{D}_E$, $\mathbf{K}_E$). A mechanical impedance behavior between the i th end-effector displacements and the internal forces can be imposed as well (*internal impedance*):

$$\mathbf{M}_{I,i} \tilde{\mathbf{a}}_i + \mathbf{D}_{I,i} \tilde{\mathbf{v}}_i + \mathbf{K}_{I,i} \mathbf{e}_i = \mathbf{h}_{I,i}, \quad (6)$$

where $\mathbf{e}_i$ is the vector expressing the displacement between the commanded and the actual pose of the i th end effector, $\tilde{\mathbf{v}}_i$ is the vector expressing the difference between commanded and actual velocities of the i th end effector, $\tilde{\mathbf{a}}_i$ is the vector expressing the difference between commanded and actual accelerations of the i th

end effector, and $\mathbf{h}_{I,i}$ is the contribution of the i th end effector to the internal force. Again, the impedance dynamics is characterized in terms of given positive definite mass, damping, and stiffness matrices ($\mathbf{M}_{I,i}$, $\mathbf{D}_{I,i}$, $\mathbf{K}_{I,i}$). An impedance scheme designed to control of both external forces and internal forces has been developed in Caccavale et al. (2008).

Summary and Future Directions

This entry provides a brief survey of the main issues related to cooperative robots, with special emphasis on modeling and control problems. Among several open research topics in cooperative manipulation, it is worth mentioning the problem of cooperative transportation and manipulation of objects via multiple mobile manipulators. In fact, although notable results have been already devised in Khatib et al. (1996), Sugar and Kumar (2002), and Bai and Wen (2010), the foreseen use of robotic teams in industrial settings (hyperflexible robotic work cells) and/or in collaboration with humans (robotic coworker concept) raises new challenges related to autonomy and safety of multiple mobile manipulators. Also, an emerging application field is given by cooperative systems composed by multiple aerial vehicle-manipulator systems (see, e.g., Fink et al. 2011).

Cross-References

- [Force Control in Robotics](#)
- [Robot Grasp Control](#)
- [Robot Motion Control](#)

Recommended Reading

An overview of the field of cooperative manipulation can be found also in Caccavale and Uchiyama (2016), where a more extended literature review and further technical details are provided. Seminal contributions to control of cooperative manipulators can be found in

Chiacchio et al. (1991a), Koivo and Unseren (1991), Sang et al. (1995), Uchiyama and Dauchez (1993), Walker et al. (1989), Wen and Kreutz-Delgado (1992), and Williams and Khatib (1993).

Bibliography

- Bai H, Wen JT (2010) Cooperative load transport: a formation-control perspective. *IEEE Trans Robot* 26: 742–750
- Bonitz RG, Hsia TC (1996) Internal force-based impedance control for cooperating manipulators. *IEEE Trans Robot Autom* 12:78–89
- Caccavale F, Uchiyama M (2016) Cooperative manipulation. In: Siciliano B, Khatib O (eds) *Springer handbook of robotics*. Springer handbooks. Springer, Cham
- Caccavale F, Chiacchio P, Chiaverini S (2000) Task-space regulation of cooperative manipulators. *Automatica* 36:879–887
- Caccavale F, Chiacchio P, Marino A, Villani L (2008) Six-Dof impedance control of dual-arm cooperative manipulators. *IEEE/ASME Trans Mechatron* 13: 576–586
- Chiacchio P, Chiaverini S, Sciavicco L, Siciliano B (1991a) Global task space manipulability ellipsoids for multiple arm systems. *IEEE Trans Robot Autom* 7:678–685
- Chiacchio P, Chiaverini S, Sciavicco L, Siciliano B (1991b) Task space dynamic analysis of multiarm system configurations. *Int J Robot Res* 10:708–715
- Chiacchio P, Chiaverini S, Siciliano B (1996) Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system. *ASME J Dyn Syst Meas Control* 118:691–697
- Fink J, Michael N, Kim S, Kumar V (2011) Planning and control for cooperative manipulation and transportation with aerial robots. *Int J Robot Res* 30:324–334
- Hsu P (1993) Coordinated control of multiple manipulator systems. *IEEE Trans Robot Autom* 9:400–410
- Hu Y-R, Goldenberg AA, Zhou C (1995) Motion and force control of coordinated robots during constrained motion tasks. *Int J Robot Res* 14:351–365
- Khatib O, Yokoi K, Chang K, Ruspini D, Holmberg R, Casal A (1996) Coordination and decentralized cooperation of multiple mobile manipulators. *J Robot Syst* 13:755–764
- Koivo AJ, Unseren MA (1991) Reduced order model and decoupled control architecture for two manipulators holding a rigid object. *ASME J Dyn Syst Meas Control* 113:646–654
- Sang KS, Holmberg R, Khatib O (1995) The augmented object model: cooperative manipulation and parallel mechanisms dynamics. In: *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, pp 470–475
- Schneider SA, Cannon RH Jr (1992) Object impedance control for cooperative manipulation: theory and experimental results. *IEEE Trans Robot Autom* 8: 383–394
- Sugar TG, Kumar V (2002) Control of cooperating mobile manipulators. *IEEE Trans Robot Autom* 18:94–103
- Uchiyama M, Dauchez P (1993) Symmetric kinematic formulation and non-master/slave coordinated control of two-arm robots. *Adv Robot* 7:361–383
- Uchiyama M, Yamashita T (1991) Adaptive load sharing for hybrid controlled two cooperative manipulators. In: *Proceedings of 1991 IEEE International Conference on Robotics and Automation (Sacramento 1991)*, pp 986–991
- Walker ID, Marcus SI, Freeman RA (1989) Distribution of dynamic loads for multiple cooperating robot manipulators. *J Robot Syst* 6:35–47
- Walker ID, Freeman RA, Marcus SI (1991) Analysis of motion and internal force loading of objects grasped by multiple cooperating manipulators. *Int J Robot Res* 10:396–409
- Wen JT, Kreutz-Delgado K (1992) Motion and force control of multiple robotic manipulators. *Automatica* 28:729–743
- Williams D, Khatib O (1993) The virtual linkage: a model for internal forces in multi-grasp manipulation. In: *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, Atlanta, pp 1025–1030
- Yamano M, Kim J-S, Konno A, Uchiyama M (2004) Cooperative control of a 3D dual-flexible-arm robot. *J Intell Robot Syst* 39:1–15
- Yukawa T, Uchiyama M, Nenchev DN, Inooka H (1996) Stability of control system in handling of a flexible object by rigid arm robots. In: *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, pp 2332–2339

Cooperative Solutions to Dynamic Games

Alain Haurie
ORDECSYS and University of Geneva,
Geneva, Switzerland
GERAD-HEC, Montréal, PQ, Canada

Abstract

This article presents the fundamental elements of the theory of cooperative games in the context of dynamic systems. The concepts of Pareto optimality, Nash bargaining solution, characteristic function, cores,

and C-optimality are discussed, and some fundamental results are recalled.

Keywords

Cores · Nash equilibrium · Pareto optimality

Introduction

Solution concepts in game theory are regrouped in two main categories called noncooperative and cooperation solutions, respectively. In the seminal book of von Neumann and Morgenstern (1944) this categorization is already made. These authors discuss zero-sum (matrix) games in normal form, where the noncooperative solution concept of saddle-point was defined and characterized, and games in characteristic function form, where solution concepts for games of coalitions were introduced. In this article we present the fundamental solution concepts of the theory of cooperative games in the context of dynamical systems. The article is organized as follows: we first recall the papers, which mark the origin of development of a theory of dynamic games; then we recall the basic concept of Pareto optimality proposed as a cooperative solution concept; we present the scalarization technique and the necessary or sufficient optimality conditions for Pareto optimality in mathematical programming and optimal control settings; we then explore the difficulties encountered when one tried to extend the Nash bargaining solution, characteristic function and cores concept to dynamic games; we show the links that exist with the theory of reachability for perturbed dynamic systems.

The Origins

One may consider that the first introduction of a cooperative game solution concept in systems and control science is due to L.A. Zadeh (1963). Two-player zero-sum dynamic games have been studied by R. Isaacs (1954) in a deterministic continuous time setting and by L. Shapley (1953)

in a discrete time stochastic setting. Nonzero-sum and m player differential games were introduced by Y.C. Ho and A.W. Starr (1969) and J.H. Case (1969). For these games cooperative solutions can be looked for to complement the noncooperative Nash equilibrium concept.

Cooperation Solution Concept

In cooperative games one is interested in nondominated solution. This solution type is related to a concept introduced by the well-known economist V. Pareto (1869) in the context of welfare economics. Consider a system with decision variables $x \in X \subset \mathbb{R}^n$ and m performance criteria $x \rightarrow \psi_j(x) \in \mathbb{R}$, $j = 1, \dots, m$ that one tries to maximize.

Definition 1 The decision $x^* \in X$ is nondominated or Pareto optimal if the following condition holds:

$$\begin{aligned} \psi_j(x) &\geq \psi_j(x^*) \quad \forall j = 1, \dots, m \\ \implies \psi_j(x) &= \psi_j(x^*) \quad \forall j = 1, \dots, m. \end{aligned}$$

In other words it is impossible to give one criterion j a value greater than $\psi_j(x^*)$ without decreasing the value of another criterion, say ℓ , which then takes a value lower than $\psi_\ell(x^*)$.

This vector-valued optimization framework corresponds to a situation where m players are engaged in a game, described in its normal form, where the strategies of the m players constitute the decision vector x and their respective payoffs are given by the m performance criteria $\psi_j(x)$, $j = 1, \dots, m$. One assumes that these players jointly take a decision that is cooperatively optimal, in the sense that no player can improve his/her payoff without deteriorating the payoff of at least one other player.

The Scalarization Technique

Let $\mathbf{r} = (r_1, r_2, \dots, r_m)$ be a given m -vector composed of normalized weights that satisfy $r_j > 0$, $j = 1, \dots, m$ and $\sum_{j=1, \dots, m} r_j = 1$.

Lemma 1 Let $x^* \in X$ be a maximum in X for the scalarized criterion $\Psi(x; \mathbf{r}) = \sum_{j=1}^m r_j \psi_j(x)$. Then x^* is a nondominated solution for the multi-objective problem.

The proof is very simple. Suppose x^* is dominated, then there exists $x^\circ \in X$ such that $\psi_j(x^\circ) \geq \psi_j(x^*)$, $\forall j = 1, \dots, m$, and $\psi_i(x^\circ) > \psi_i(x^*)$ for one $i \in \{1, \dots, m\}$. Since all the r_j are > 0 , this yields $\sum_{j=1}^m r_j \psi_j(x^\circ) > \sum_{j=1}^m r_j \psi_j(x^*)$, which contradicts the maximizing property of x^* . This result shows that it will be very easy to find many Pareto optimal solutions by varying a strictly positive weighting of the criteria. But this procedure will not find all of the nondominated solutions.

Conditions for Pareto Optimality in Mathematical Programming

N.O. Da Cunha and E. Polak (1967b) have obtained the first necessary conditions for multi-objective optimization. The problem they consider is

$$\begin{aligned} &\text{Pareto Opt. } \psi_j(x) \quad j = 1, \dots, m \\ &\text{s.t.} \\ &\varphi_k(x) \leq 0 \quad k = 1, \dots, p \end{aligned}$$

where the functions $x \in \mathbb{R}^n \mapsto \psi_j(x) \in \mathbb{R}$, $j = 1, \dots, m$, and $x \mapsto \varphi_k(x) \in \mathbb{R}$, $k = 1, \dots, p$ are continuously differentiable (C^1) and where we assume that the constraint qualification conditions of mathematical programming hold for this problem too. They proved the following theorem.

Theorem 1 Let x^* be a Pareto optimal solution of the problem defined above. Then there exists a vector λ of p multipliers λ_k , $k = 1, \dots, p$, and a vector $\mathbf{r} \neq 0$ of m weights $r_j \geq 0$, such that the following conditions hold

$$\begin{aligned} \frac{\partial}{\partial x} \mathcal{L}(x^*; \mathbf{r}; \lambda) &= 0 \\ \varphi_k(x^*) &\leq 0 \\ \lambda_k \varphi_k(x^*) &= 0 \\ \lambda_k &\geq 0, \end{aligned}$$

where $\mathcal{L}(x^*; \mathbf{r}; \lambda)$ is the weighted Lagrangian defined by

$$\mathcal{L}(x; \mathbf{r}; \lambda) = \sum_{j=1}^m r_j \psi_j(x) + \sum_{k=1}^p \lambda_k \varphi_k(x).$$

So there is a local scalarization principle for Pareto optimality.

Maximum Principle

The extension of Pareto optimality concept to control systems was done by several authors (Binmore et al. 1986; Zadeh 1963; Basile and Vincent 1970; Vincent and Leitmann 1970; Salukvadze 1971; Leitmann et al. 1972; Blaqui re et al. 1972; Bellassali and Jourani 2004), the main result being an extension of the maximum principle of Pontryagin. Let a system be governed by state equations:

$$\dot{x}(t) = f(x(t), u(t)) \quad (1)$$

$$u(t) \in U \quad (2)$$

$$x(0) = x_0 \quad (3)$$

$$t \in [0, T] \quad (4)$$

where $x \in \mathbb{R}^n$ is the state variable of the system, $u \in U \subset \mathbb{R}^p$ with U compact is the control variable, and $[0, T]$ is the control horizon. The system is evaluated by m performance criteria of the form

$$\psi_j(x(\cdot), u(\cdot)) = \int_0^T g_j(x(t), u(t)) dt + G_j(x(T)), \quad (5)$$

for $j = 1, \dots, m$. Under the usual assumptions of control theory, i.e., $f(\cdot, \cdot)$ and $g_j(\cdot, \cdot)$, $j = 1, \dots, m$, being C^1 in x and continuous in u , $G_j(\cdot)$ being C^1 in x , one can prove the following.

Theorem 2 Let $\{x^*(t) : t \in [0, T]\}$ be a Pareto optimal trajectory, generated at initial state x° by the Pareto optimal control $\{u^*(t) : t \in [0, T]\}$. Then there exist costate vectors $\{\lambda^*(t) : t \in [0, T]\}$ and a vector of positive weights $\mathbf{r} \neq 0 \in \mathbb{R}^m$, with components $r_j \geq 0$, $\sum_{j=1}^m r_j = 1$, such that the following relations hold:

$$\dot{x}^*(t) = \frac{\partial}{\partial \lambda} H(x^*(t), u^*(t); \lambda(t); \mathbf{r}) \quad (6)$$

$$\dot{\lambda}(t) = -\frac{\partial}{\partial x} H(x^*(t), u^*(t); \lambda(t); \mathbf{r}) \quad (7)$$

$$x^*(0) = x_o \quad (8)$$

$$\lambda(T) = \sum_{j=1}^m r_j \frac{\partial}{\partial x} G_j(x(T)) \quad (9)$$

with

$$\begin{aligned} & H(x^*(t), u^*(t); \lambda(t); \mathbf{r}) \\ &= \max_{u \in U} H(x^*(t), u; \lambda(t); \mathbf{r}) \end{aligned}$$

where the weighted Hamiltonian is defined by

$$H(x, u; \lambda; \mathbf{r}) = \sum_{j=1}^m r_j g_j(x, u) + \lambda^T f(x, u).$$

The proof of this result necessitates some additional regularity assumptions. Some of these conditions imply that there exist differentiable Bellman value functions (see, e.g., Blaquièrre et al. 1972); some others use the formalism of nonsmooth analysis (see, e.g., Bellassali and Jourani 2004).

The Nash Bargaining Solution

Since Pareto optimal solutions are numerous (actually since a subset of Pareto outcomes are indexed over the weightings $\mathbf{r}$, $r_j > 0$, $\sum_{j=1}^m r_j = 1$), one can expect, in the payoff m -dimensional space, to have a manifold of Pareto outcomes. Therefore, the problem that we must solve now is *how to select the “best” Pareto outcome?* “Best” is a misnomer here, because, by their very definition, two Pareto outcomes cannot be compared or gauged. The choice of a Pareto outcome that satisfies each player must be the result of some bargaining. J. Nash addressed this problem very early, in 1951, using a two-player game setting. He developed an axiomatic approach where he proposed four behavior axioms which, if accepted, would determine a

unique choice for the bargaining solution. These axioms are called respectively, (i) invariance to affine transformations of utility representations, (ii) Pareto optimality, (iii) independence of irrelevant alternatives, and (iv) symmetry. Then the bargaining point is the Pareto optimal solution that maximizes the product

$$x^* = \operatorname{argmax}_x (\psi_1(x) - \psi_1(x^\circ))(\psi_2(x) - \psi_2(x^\circ))$$

where x° is the status quo decision, in case bargaining fails, and $(\psi_j(x^\circ))$, $j = 1, 2$ are the payoffs associated with this no-accord decision (this defines the so-called threat point). It has been proved (Binmore et al. 1986) that this solution could be obtained also as the solution of an auxiliary dynamic game in which a sequence of claims and counterclaims is made by the two players when they bargain.

When extended directly to the context of differential or multistage games, the Nash bargaining solution concept proved to lack the important property of time consistency. This was first noticed in Haurie (1976). Let a dynamic game be defined by Eqs. (1)–(5), with $j = 1, 2$. Suppose the status quo decision, if no agreement is reached at initial state ($t = 0$, $x(0) = x^\circ$), consists in playing an open-loop Nash equilibrium, defined by the controls $u_j^N(\cdot) : [0, T] \rightarrow U_j$, $j = 1, 2$ and generating the trajectory $x^N(\cdot) : [0, T] \rightarrow \mathbb{R}^n$, with $x^N(0) = x_o$. Now applying the Nash bargaining solution scheme to the data of this differential game played at time $t = 0$ and state $x(0) = x_o$, one identifies a particular Pareto optimal solution, associated with the controls $u^*(\cdot) : [0, T] \rightarrow U_j$, $j = 1, 2$ and generating the trajectory $x^*(\cdot) : [0, T] \rightarrow \mathbb{R}^n$, with $x^*(0) = x_o$. Now assume the two players renegotiate the agreement to play $u_j^*(\cdot)$ at an intermediate point of the Pareto optimal trajectory $(\tau, x^*(\tau))$, $\tau \in (0, T)$. When computed from that point, the status quo strategies are in general not the same as they were at $(0, x_o)$; furthermore, the shape of the Pareto frontier, when the game is played from $(\tau, x^*(\tau))$, is different from what it is when the game is played at $(0, x_o)$. For these two reasons the bargaining solution at $(\tau, x^*(\tau))$

will not coincide in general with the restriction to the interval $[\tau, T]$ of the bargaining solution from $(0, x_0)$. This implies that the solution concept is not *time consistent*. Using feedback strategies, instead of open-loop ones, does not help, as the same phenomena (change of status quo and change of Pareto frontier) occur in a feedback strategy context.

This shows that the cooperative game solutions proposed in the classical theory of games cannot be applied without precaution in a dynamic setting when players have the possibility to renegotiate agreements at any intermediary point $(t, x^*(t))$ of the bargained solution trajectory.

Cores and C-Optimality in Dynamic Games

Characteristic functions and the associated solution concept of core are important elements in the classical theory of cooperative games. In two papers (Haurie and Delfour 1974; Haurie 1975) the basic definitions and properties of the concept of core in dynamic cooperative games were presented. Consider the multistage system, controlled by a set M of m players and defined by

$$\begin{aligned} x(k+1) &= f^k(x(k), u_M(k)), \\ k &= 0, 1, \dots, K-1 \\ x(i) &= x^i, \quad i \in \{0, 1, \dots, K-1\} \\ u_M(k) &\triangleq (u_j(k))_{j \in M} \in U_M(k) \triangleq \prod_{j \in M} U_j(k). \end{aligned}$$

From the initial point (i, x^i) a control sequence $(u_M(i), \dots, u_M(K-1))$ generates for each player j a payoff defined as follows:

$$\begin{aligned} J_j(i, x^i; u_M(i), \dots, u_M(K-1)) &\triangleq \\ \sum_{k=i}^{K-1} \Phi_j(x(k), u_M(k)) &+ \Upsilon_j(x(K)). \end{aligned}$$

A subset S of M is called a coalition. Let $\mu_S^k : x(k) \mapsto u_S(k) \in \prod_{j \in S} U_j(k)$ be a feedback control for the coalition defined at each stage k . A player $j \in S$ considers then, from any initial point (i, x^i) , his guaranteed payoff:

$$\begin{aligned} \Psi_j(i, x^i; \mu_S^i, \dots, \mu_S^{K-1}) &\triangleq \\ \inf_{u_{M-S}(i) \in U_{M-S}(i), \dots, u_{M-S}(K-1) \in U_{M-S}(K-1)} & \\ \sum_{k=i}^{K-1} \Phi_j(x(k), [\mu_S^k(x(k)), u_{M-S}(k)]) & \\ + \Upsilon_j(x(K)). & \end{aligned}$$

Definition 2 The characteristic function at stage i for coalition $S \subset M$ is the mapping $v^i : (S, x^i) \mapsto v^i(S, x^i) \subset \mathbb{IR}^S$ defined by

$$\begin{aligned} \omega_S &\triangleq (\omega_j)_{j \in S} \in v^i(S, x^i) \Leftrightarrow \\ \exists \mu_S^i, \dots, \mu_S^{K-1} : \forall j \in S & \\ \Psi_j(i, x^i; \mu_S^i, \dots, \mu_S^{K-1}) &\geq \omega_j. \end{aligned}$$

In other words, there is a feedback law for the coalition S which guarantees at least ω_j to each player j in the coalition.

Suppose that in a cooperative agreement, at point (i, x^i) , the coalition S is proposed a gain vector ω_S which is interior to $v^i(S, x^i)$. Then coalition S will block this agreement, because using an appropriate feedback, the coalition can guarantee a better payoff to each of its members. We can now extend the definition of the core of a cooperative game to the context of dynamic games, as the set of agreement gains that cannot be blocked by any coalition.

Definition 3 The core $\Omega(i, x^i)$ at point (i, x^i) is the set of gain vectors $\omega_M \triangleq (\omega_j)_{j \in M}$ such that:

1. There exists a Pareto optimal control $u_M^*(i), \dots, u_M^*(K-1)$ for which $\omega_j = J_j(i, x^i; u_M^*(i), \dots, u_M^*(K-1))$,
2. $\forall S \subset M$ the projection of ω_M in $\mathbb{IR}^S$ is not interior to $v^i(S, x^i)$

Playing a cooperative game, one would be interested in finding a solution where the gain-to-go remains in the core at each point of the trajectory. This leads us to define the following.

Definition 4 A control $\tilde{u}^o \triangleq (u_M^o(0), \dots, u_M^o(K-1))$ is C -optimal at $(0, x^0)$ if $\tilde{u}^o$ is Pareto optimal generating a state trajectory

$$\{x^o(0) = x^0, x^o(1), \dots, x^o(K)\}$$

and a sequence of gain-to-go values

$$\begin{aligned} \omega_j^o(i) &= J_j(i, x^o(i); u_M^o(i), \dots, u_M^o(K-1)), \\ i &= 0, \dots, K-1 \end{aligned}$$

such that $\forall i = 0, 1, \dots, K-1$, the m -vector $\omega_M^o(i)$ is element of the core $\Omega(i, x^o(i))$.

A C -optimal control generates an agreement which cannot be blocked by any coalition along the Pareto optimal trajectory. It can be shown on examples that a Pareto optimal trajectory which has the gain-to-go vector in the core at initial point $(0, x_0)$ is not C -optimal.

Links with Reachability Theory for Perturbed Systems

The computation of characteristic functions can be made using the techniques developed to study reachability of dynamic systems with set constrained disturbances (see Bertsekas and Rhodes 1971). Consider the particular case of a linear system

$$x(k+1) = A^k x(k) + \sum_{j \in M} B_j^k u_j(k) \quad (10)$$

where $x \in \mathbb{R}^n$, $u_j \in U_j^k \subset \mathbb{R}^{p_j}$, where U_j^k is a convex-bound set and A^k, B_j^k are matrices of appropriate dimensions. Let the payoff to player j be defined by:

$$\begin{aligned} J_j(i, x^i; u_M(i), \dots, u_M(K-1)) &\triangleq \\ \sum_{k=i}^{K-1} \phi_j^k(x(k)) + \gamma_j^k(u_j(k)) + \Upsilon_j(x(K)). \end{aligned}$$

Algorithm Here we use the notations $\phi_S^k \triangleq (\phi_j^k)_{j \in S}$ and $B_S^k u_S \triangleq \sum_{j \in S} B_j^k u_j$. Also we denote $\{u + V\}$, where u is a vector in $\mathbb{R}^m$ and $V \subset \mathbb{R}^m$, the set of vectors $u + v$, $\forall v \in V$. Then

1. $\forall x^K v^K(S, x^K) \triangleq \{\omega_S \in \mathbb{R}^S : \Upsilon_S(x^K) \geq \omega_S\}$
2. $\forall x \mathcal{E}^{k+1}(S, x) \triangleq \cap_{v \in U_{M-S}} v^{k+1}(S, x + B_{M-S}^k v)$
3. $\forall x^k \mathcal{H}^k(S, x^k) \triangleq \bigcup_{u \in U_S} \{\gamma_S^k(u) + \mathcal{E}^{k+1}(S, A^k x^k + B_S^k u)\}$
4. $\forall x^k v^k(S, x^k) = \{\phi_S^k(x^k) + \mathcal{H}^k(S, x^k)\}.$

In an open-loop control setting, the calculation of characteristic function can be done using the concept of Pareto optimal solution for a system with set constrained disturbances, as shown in Goffin and Haurie (1973, 1976) and Haurie (1973).

Conclusion

Since the foundations of a theory of cooperative solutions to dynamic games, recalled in this article, the research has evolved toward the search for cooperative solutions that could be also equilibrium solution, using for that purpose a class of memory strategies Haurie and Towinski (1985), and has found a very important domain of application in the assessment of environmental agreements, in particular those related to the climate change issue. For example, the sustainability of solutions in the core of a dynamic game modeling international environmental negotiations is studied in Germain et al. (2003). A more encompassing model of dynamic formation of coalitions and stabilization of solutions through the use of threats is proposed in Breton et al. (2010). These references are indicative of the trend of research in this field.

Cross-References

► [Dynamic Noncooperative Games](#)

- [Game Theory: A General Introduction and a Historical Overview](#)
- [Strategic Form Games and Nash Equilibrium](#)

Bibliography

- Basile G, Vincent TL (1970) Absolutely cooperative solution for a linear, multiplayer differential game. *J Optim Theory Appl* 6:41–46
- Bellassali S, Jourani A (2004) Necessary optimality conditions in multiobjective dynamic optimization. *SIAM J Control Optim* 42:2043–2061
- Bertsekas DP, Rhodes IB (1971) On the minimax reachability of target sets and target tubes. *Automatica* 7: 23–247
- Binmore K, Rubinstein A, Wolinsky A (1986) The Nash bargaining solution in economic modelling. *Rand J Econ* 17(2):176–188
- Blaquière A, Juricek L, Wiese KE (1972) Geometry of Pareto equilibria and maximum principle in n -person differential games. *J Optim Theory Appl* 38:223–243
- Breton M, Sbragia L, Zaccour G (2010) A dynamic model for international environmental agreements. *Environ Resour Econ* 45:25–48
- Case JH (1969) Toward a theory of many player differential games. *SIAM J Control* 7(2):179–197
- Da Cunha NO, Polak E (1967a) Constrained minimization under vector-valued criteria in linear topological spaces. In: Balakrishnan AV, Neustadt LW (eds) *Mathematical theory of control*. Academic, New York, pp 96–108
- Da Cunha NO, Polak E (1967b) Constrained minimization under vector-valued criteria in finite dimensional spaces. *J Math Anal Appl* 19:103–124
- Germain M, Toint P, Tulkens H, Zeeuw A (2003) Transfers to sustain dynamic core-theoretic cooperation in international stock pollutant control. *J Econ Dyn Control* 28:79–99
- Goffin JL, Haurie A (1973) Necessary conditions and sufficient conditions for Pareto optimality in a multicriterion perturbed system. In: Conti R, Ruberti A (eds) *5th conference on optimization techniques*, Rome. Lecture notes in computer science, vol 4 Springer
- Goffin JL, Haurie A (1976) Pareto optimality with nondifferentiable cost functions. In: Thiriez H, Zions S (eds) *Multiple criteria decision making*. Lecture notes in economics and mathematical systems, vol 130. Springer, Berlin/New York, pp 232–246
- Haurie A (1973) On Pareto optimal decisions for a coalition of a subset of players. *IEEE Trans Autom Control* 18:144–149
- Haurie A (1975) On some properties of the characteristic function and the core of a multistage game of coalition. *IEEE Trans Autom Control* 20(2):238–241
- Haurie A (1976) A note on nonzero-sum differential games with bargaining solutions. *J Optim Theory Appl* 13:31–39
- Haurie A, Delfour MC (1974) Individual and collective rationality in a dynamic Pareto equilibrium. *J Optim Appl* 13(3):290–302
- Haurie A, Towinski B (1985) Definition and properties of cooperative equilibria in a two-player game of infinite duration. *J Optim Theory Appl* 46(4):525–534
- Isaacs R (1954) *Differential games I: introduction*. Rand Research Memorandum, RM-1391-30. Rand Corporation, Santa Monica
- Leitmann G, Rocklin S, Vincent TL (1972) A note on control space properties of cooperative games. *J Optim Theory Appl* 9:379–390
- Nash J (1950) The bargaining problem. *Econometrica* 18(2):155–162
- Pareto V (1896) *Cours d'Economie Politique*. Rogue, Lausanne
- Salukvadze ME (1971) On the optimization of control systems with vector criteria. In: *Proceedings of the 11th all-union conference on control*, Part 2. Nauka
- Shapley LS (1953) Stochastic games. *PNAS* 39(10):1095–1100
- Starr AW, Ho YC (1969) Nonzero-sum differential games. *J Optim Theory Appl* 3(3):184–206
- Vincent TL, Leitmann G (1970) Control space properties of cooperative games. *J Optim Theory Appl* 6(2): 91–113
- von Neumann J, Morgenstern O (1944) *Theory of Games and Economic Behavior*, Princeton University Press
- Zadeh LA (1963) Optimality and non-scalar-valued performance criteria. *IEEE Trans Autom Control* AC-8:59–60

Coordination of Distributed Energy Resources for Provision of Ancillary Services: Architectures and Algorithms

Alejandro D. Domínguez-García¹ and Christoforos N. Hadjicostis²

¹University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, USA

²Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus

Abstract

We discuss the utilization of distributed energy resources (DERs) to provide active and reactive power support for ancillary services. Though the amount of active and/or

reactive power provided individually by each of these resources can be very small, their presence in large numbers in power distribution networks implies that, under proper coordination mechanisms, they can collectively provide substantial active and reactive power regulation capacity. In this entry, we provide a simple formulation of the DER coordination problem for enabling their utilization to provide ancillary services. We also provide specific architectures and algorithmic solutions to solve the DER coordination problem, with focus on decentralized solutions.

Keywords

Ancillary services · Consensus · Distributed algorithms · Distributed energy resources (DERs)

Introduction

On the distribution side of a power system, there are many distributed energy resources (DERs), e.g., photovoltaic (PV) installations, plug-in hybrid electric vehicles (PHEVs), and thermostatically controlled loads (TCLs), that can be potentially used to provide ancillary services, e.g., reactive power support for voltage control (see, e.g., Turitsyn et al. (2011) and the references therein) and active power up and down regulation for frequency control (see, e.g., Callaway and Hiskens (2011) and the references therein). To enable DERs to provide ancillary services, it is necessary to develop appropriate control and coordination mechanisms. One potential solution relies on a centralized control architecture in which each DER is directly coordinated by (and communicates with) a central decision maker. An alternative approach is to distribute the decision making, which obviates the need for a central decision maker to coordinate the DERs. In both cases, the decision making involves solving a

resource allocation problem for coordinating the DERs to collectively provide a certain amount of a resource (e.g., active or reactive power).

In a practical setting, whether a centralized or a distributed architecture is adopted, the control of DERs for ancillary services provision will involve some aggregating entity that will gather together and coordinate a set of DERs, which will provide certain amount of active or reactive power in exchange for monetary benefits. In general, these aggregating entities are the ones that interact with the ancillary services market, and through some market-clearing mechanism, they enter a contract to provide some amount of resource, e.g., active and/or reactive power over a period of time. The goal of the aggregating entity is to provide this amount of resource by properly coordinating and controlling the DERs, while ensuring that the total monetary compensation to the DERs for providing the resource is below the monetary benefit that the aggregating entity obtains by selling the resource in the ancillary services market.

In the context above, a household with a solar PV rooftop installation and a PHEV might choose to offer the PV installation to a renewable aggregator so it is utilized to provide reactive power support (this can be achieved as long as the PV installation power electronics-based grid interface has the correct topology Domínguez-García et al. 2011). Additionally, the household could offer its PHEV to a battery vehicle aggregator to be used as a controllable load for energy peak shaving during peak hours and load leveling at night (Guille and Gross 2009). Finally, the household might choose to enroll in a demand response program in which it allows a demand response provider to control its TCLs to provide frequency regulation services (Callaway and Hiskens 2011). In general, the renewable aggregator, the battery vehicle aggregator, and the demand response provider can be either separate entities or they can be the same entity. In this entry, we will refer to these aggregating entities as *aggregators*.

The Problem of DER Coordination

Without loss of generality, denote by x_j the amount of resource provided by DER i without specifying whether it is active or reactive power. [However, it is understood that each DER provides (or consumes) the same type of resource, i.e., all the x_i 's are either active or reactive power.] Let $0 < \underline{x}_i < \bar{x}_i$, for $i = 1, 2, \dots, n$, denote the minimum ($\underline{x}_i$) and maximum ($\bar{x}_i$) capacity limits on the amount of resource x_i that node i can provide. Denote by X the total amount of resource that the DERs must collectively provide to satisfy the aggregator request. Let $\pi_i(x_i)$ denote the price that the aggregator pays DER i per unit of resource x_i that it provides. Then, the objective of the aggregator in the DER coordination problem is to minimize the total monetary amount to be paid to the DERs for providing the total amount of resource X while satisfying the individual capacity constraints of the DERs. Thus, the DER coordination problem can be formulated as follows:

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^n x_i \pi_i(x_i) \\ & \text{subject to} && \sum_{i=1}^n x_i = X \\ & && 0 < \underline{x}_i \leq x_i \leq \bar{x}_i, \forall i. \end{aligned} \quad (1)$$

By allowing heterogeneity in the price per unit of resource that the aggregator offers to each DER, we can take into account the fact that the aggregator might value classes of DERs differently. For example, the downregulation capacity provided by a residential PV installation (which is achieved by curtailing its power) might be valued differently from the downregulation capacity provided by a TCL or a PHEV (both would need to absorb additional power in order to provide downregulation).

It is not difficult to see that if the price functions $\pi_i(\cdot)$, $i = 1, 2, \dots, n$, are convex

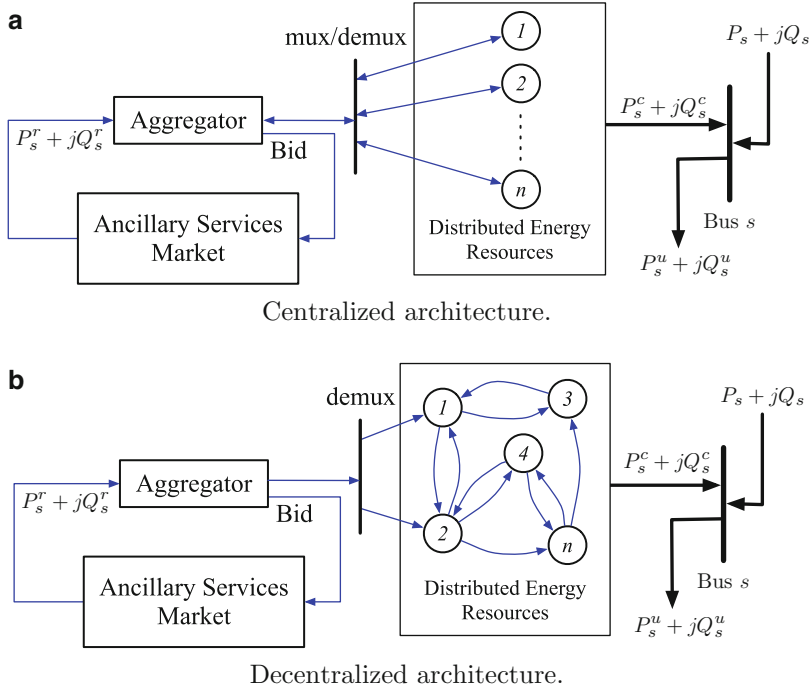
and nondecreasing, then the cost function $\sum_{i=1}^n x_i \pi_i(x_i)$ is convex; thus, if the problem in (1) is feasible, then there exists a globally optimal solution. Additionally, if the price per unit of resource is linear with the amount of resource, i.e., $\pi_i(x_i) = c_i x_i$, $i = 1, 2, \dots, n$, then $x_i \pi_i(x_i) = c_i x_i^2$, $i = 1, 2, \dots, n$, and the problem in (1) reduces to a quadratic program. Also, if the price per unit of resource is constant, i.e., $\pi_i(x_i) = c_i$, $i = 1, 2, \dots, n$, then $x_i \pi_i(x_i) = c_i x_i$, $i = 1, 2, \dots, n$, and the problem in (1) reduces to a linear program. Finally, if $\pi_i(x_i) = \pi(x_i) = c$, $i = 1, 2, \dots, n$, for some constant $c > 0$, i.e., the price offered by the aggregator is constant and the same for all DERs, then the optimization problem in (1) becomes a feasibility problem of the form

$$\begin{aligned} & \text{find} && x_1, x_2, \dots, x_n \\ & \text{subject to} && \sum_{i=1}^n x_i = X \\ & && 0 < \underline{x}_i \leq x_i \leq \bar{x}_i, \forall i. \end{aligned} \quad (2)$$

If the problem in (2) is indeed feasible (i.e., $\sum_{i=1}^n \underline{x}_i \leq X \leq \sum_{i=1}^n \bar{x}_i$), then there is an infinite number of solutions. One such solution, which we refer to as *fair splitting*, is given by

$$x_i = \underline{x}_i + \frac{X - \sum_{l=1}^n \underline{x}_l}{\sum_{l=1}^n (\bar{x}_l - \underline{x}_l)} (\bar{x}_i - \underline{x}_i), \forall i. \quad (3)$$

The formulation to the DER coordination problem provided in (2) is not the only possible one. In this regard, and in the context of PHEVs, several recent works have proposed game-theoretic formulations to the problem (Tushar et al. 2012; Ma et al. 2013; Gharesifard et al. 2013). For example, in Gharesifard et al. (2013), the authors assume that each PHEV is a decision maker and can freely choose to participate after receiving a request from the aggregator. The decision that each PHEV is faced with depends on its own utility function, along with some pricing strategy designed by the aggregator. The



Coordination of Distributed Energy Resources for Provision of Ancillary Services: Architectures and Algorithms, Fig. 1 Control architecture alternatives. (a) Centralized architecture. (b) Decentralized architecture

PHEVs are assumed to be price anticipating in the sense that they are aware of the fact that the pricing is designed by the aggregator with respect to the average energy available. Another alternative is to formulate the DER coordination problem as a scheduling problem (Subramanian et al. 2012; Chen et al. 2012), where the DERs are treated as tasks. Then, the problem is to develop real-time scheduling policies to service these tasks.

Architectures

Next, we describe two possible architectures that can be utilized to implement the proper algorithms for solving the DER coordination problem as formulated in (1). Specifically, we describe a centralized architecture that requires the aggregator to communicate bidirectionally with each DER and a distributed architecture that requires the aggregator to only unidirectionally communicate with a limited number of DERs but requires

some additional exchange of information (not necessarily through bidirectional communication links) among the DERs.

Centralized Architecture

A solution can be achieved through the completely centralized architecture of Fig. 1a, where the aggregator can exchange information with each available DER. In this scenario, each DER can inform the aggregator about its active and/or reactive capacity limits and other operational constraints, e.g., maintenance schedule. After gathering all this information, the aggregator solves the optimization program in (1), the solution of which will determine how to allocate among the resources the total amount of active power P_s^r and/or reactive power Q_s^r that it needs to provide. Then, the aggregator sends individual commands to each DER so they modify their active and or reactive power generation according to the solution of (1) computed by the aggregator. In this centralized solution, however, it is necessary to overlay a communication network

connecting the aggregator with each resource and to maintain knowledge of the resources that are available at any given time.

Decentralized Architecture

An alternative is to use the decentralized control architecture of Fig. 1b, where the aggregator relays information to a limited number of DERs that it can directly communicate with and each DER is able to exchange information with a number of other close-by DERs. For example, the aggregator might broadcast the prices to be paid to each type of DER. Then, through some distributed protocol that adheres to the communication network interconnecting the DERs, the information relayed by the aggregator to this limited number of DERs is disseminated to all other available DERs. This dissemination process may rely on flooding algorithms, message-passing protocols, or linear-iterative algorithms as proposed in Domínguez-García and Hadjicostis (2010, 2011). After the dissemination process is complete and through a distributed computation over the communication network, the DERs can solve the optimization program in (1) and determine its active and/or reactive power contribution.

A decentralized architecture like the one in Fig. 1b may offer several advantages over the centralized one in Fig. 1a, including the following. First, a decentralized architecture may be more economical because it does not require communication between the aggregator and the various DERs. Also, a decentralized architecture does not require the aggregator to have a complete knowledge of the DERs available. Additionally, a decentralized architecture can be more resilient to faults and/or unpredictable behavioral patterns by the DERs. Finally, the practical implementation of such decentralized architecture can rely on inexpensive and simple hardware. For example, the testbed described in Domínguez-García et al. (2012a), which is used to solve a particular instance of the problem in (1), uses Arduino microcontrollers (see Arduino for a description) outfitted with wireless transceivers implementing a ZigBee protocol (see ZigBee for a description).

Algorithms

Ultimately, whether a centralized or a decentralized architecture is adopted, it is necessary to solve the optimization problem in (1). If a centralized architecture is adopted, then solving (1) is relatively straightforward using, e.g., standard gradient-descent algorithms (see, e.g., Bertsekas and Tsitsiklis 1997). Beyond the DER coordination problem and the specific formulation in (1), solving an optimization problem is challenging if a decentralized architecture is adopted (especially if the communication links between DERs are not bidirectional); this has spurred significant research in the last few years (see, e.g., Bertsekas and Tsitsiklis 1997, Xiao et al. 2006, Nedic et al. 2010, Zanella et al. 2011, Gharesifard and Cortes 2012, and the references therein).

In the specific context of the DER coordination problem as formulated in (1), when the cost functions are assumed to be quadratic and the communication between DERs is not bidirectional, an algorithm amenable for implementation in a decentralized architecture like the one in Fig. 1b has been proposed in Domínguez-García et al. (2012a). Also, in the context of Fig. 1b, when the communication between DERs are bidirectional, the DER coordination problem, as formulated in (1), can be solved using an algorithm proposed in Kar and Hug (2012).

As mentioned earlier, when the price offered by the aggregator is constant and identical for all DERs, the problem in (1) reduces to the feasibility problem in (2). One possible solution to this feasibility problem is the fair-splitting solution in (3). Next, we describe a linear-iterative algorithm – originally proposed in Domínguez-García and Hadjicostis (2010, 2011) and referred to as *ratio consensus* – that allows the DERs to individually determine its contribution so that the fair-splitting solution is achieved.

Ratio Consensus: A Distributed Algorithm for Fair Splitting

We assume that each DER is equipped with a processor that can perform simple computations and can exchange information with neighboring

DERs. In particular, the information exchange between DERs can be described by a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, 2, \dots, n\}$ is the vertex set (each vertex – or node – corresponds to a DER) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, where $(i, j) \in \mathcal{E}$ if node i can receive information from node j . We require $\mathcal{G}$ to be *strongly connected*, i.e., for any pair of vertices l and l' , there exists a path that starts in l and ends in l' . Let $\mathcal{L}^+ \subseteq \mathcal{V}$, $\mathcal{L}^+ \neq \emptyset$ denote the set of nodes that the aggregator is able to directly communicate with.

The processor of each DER i maintains two values y_i and z_i , which we refer to as internal states, and updates them (independently of each other) to be, respectively, a linear combination of DER i 's own previous internal states and the previous internal states of all nodes that can possibly transmit information to node i (including itself). In particular, for all $k \geq 0$, each node i updates its two internal states as follows:

$$y_i[k+1] = \sum_{j \in \mathcal{N}_i^-} \frac{1}{\mathcal{D}_j^+} y_j[k], \quad (4)$$

$$z_i[k+1] = \sum_{j \in \mathcal{N}_i^-} \frac{1}{\mathcal{D}_j^+} z_j[k], \quad (5)$$

where $\mathcal{N}_i^- = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$, i.e., all nodes that can possibly transmit information to node i (including itself); and $\mathcal{D}_i^+$ is the out-degree of node i , i.e., the number of nodes to which node i can possibly transmit information (including itself). The initial conditions in (4) are set to $y_i[0] = X/m - \underline{x}_i$ if $i \in \mathcal{L}^+$, and $y_i[0] = -\underline{x}_i$ otherwise and the initial conditions in (5) are set to $z_i[0] = \bar{x}_i - \underline{x}_i$. Then, as shown in Domínguez-García and Hadjicostis (2011), as long as $\sum_{l=1}^n \bar{x}_l \leq X \leq \sum_{l=1}^n \underline{x}_l$, each DER i can asymptotically calculate its contribution as

$$x_i = \underline{x}_i + \gamma(\bar{x}_i - \underline{x}_i) \quad (6)$$

where for all i

$$\lim_{k \rightarrow \infty} \frac{y_i[k]}{z_i[k]} = \frac{X - \sum_{l=1}^n \underline{x}_l}{\sum_{l=1}^n (\bar{x}_l - \underline{x}_l)} := \gamma. \quad (7)$$

It is important to note that the algorithm in (4)–(7) also serves as a primitive for the algorithm proposed in Domínguez-García et al. (2012a), which solves the problem in (1) when the cost function is quadratic. Also, the algorithm in (4)–(7) is not resilient to packet-dropping communication links or imperfect synchronization among the DERs, which makes it difficult to implement in practice; however, there are robustified variants of this algorithm that address these issues Domínguez-García et al. (2012b) and have been demonstrated to work in practice (Domínguez-García et al. 2012a).

Cross-References

- Averaging Algorithms and Consensus
- Distributed Optimization
- Electric Energy Transfer and Control via Power Electronics
- Flocking in Networked Systems
- Graphs for Modeling Networked Interactions
- Network Games
- Networked Systems

Bibliography

- Arduino [Online]. Available: <http://www.arduino.cc>
- Bertsekas DP, Tsitsiklis JN (1997) Parallel and distributed computation. Athena Scientific, Belmont
- Callaway DS, Hiskens IA (2011) Achieving controllability of electric loads. Proc IEEE 99(1): 184–199
- Chen S, Ji Y, Tong L (2012) Large scale charging of electric vehicles. In: Proceedings of the IEEE power and energy society general meeting, San Diego
- Domínguez-García AD, Hadjicostis CN (2010) Coordination and control of distributed energy resources for

- provision of ancillary services. In: Proceedings of the IEEE SmartGridComm, Gaithersburg
- Domínguez-García AD, Hadjicostis CN (2011) Distributed algorithms for control of demand response and distributed energy resources. In: Proceedings of the IEEE conference on decision and control, Orlando
- Domínguez-García AD, Hadjicostis CN, Krein PT, Cady ST (2011) Small inverter-interfaced distributed energy resources for reactive power support. In: Proceedings of the IEEE applied power electronics conference and exposition, Fort Worth
- Domínguez-García AD, Cady ST, Hadjicostis CN (2012a) Decentralized optimal dispatch of distributed energy resources. In: Proceedings of the IEEE conference on decision and control, Maui
- Domínguez-García AD, Hadjicostis CN, Vaidya N (2012b) Resilient networked control of distributed energy resources. *IEEE J Sel Areas Commun* 30(6):1137–1148
- Gharesifard B, Cortes J (2012) Continuous-time distributed convex optimization on weight-balanced digraphs. In: Proceedings of the IEEE conference on decision and control, Maui
- Gharesifard B, Domínguez-García AD, Başar T (2013) Price-based distributed control for networked plug-in electric vehicles. In: Proceedings of the American control conference, Washington, DC
- Guille C, Gross G (2009) A conceptual framework for the vehicle-to-grid (V2G) implementation. *Energy Policy* 37(11):4379–4390
- Kar S, Hug G (2012) Distributed robust economic dispatch in power systems: a consensus + innovations approach. In: Proceedings of the IEEE power and energy society general meeting, San Diego
- Ma Z, Callaway DS, Hiskens IA (2013) Decentralized charging control of large populations of plug-in electric vehicles. *IEEE Trans Control Syst Technol* 21:67–78
- Nedic A, Ozdaglar A, Parrilo PA (2010) Constrained consensus and optimization in multi-agent networks. *IEEE Trans Autom Control* 55(4):922–938
- Subramanian A, García M, Domínguez-García AD, Callaway DC, Poolla K, Varaiya P (2012) Real-time scheduling of deferrable electric loads. In: Proceedings of the American control conference, Montreal
- Turitsyn K, Sulc P, Backhaus S, Chertkov M (2011) Options for control of reactive power by distributed photovoltaic generators. *Proc IEEE* 99(6):1063–1073
- Tushar W, Saad W, Poor HV, Smith DB (2012) Economics of electric vehicle charging: a game theoretic approach. *IEEE Trans Smart Grids* 3(4):1767–1778
- Xiao L, Boyd S, Tseng CP (2006) Optimal scaling of a gradient method for distributed resource allocation. *J Optim Theory Appl* 129(3):469–488

Zanella F, Varagnolo D, Cenedese A, Pillonetto G, Schenato L (2011) Newton-Raphson consensus for distributed convex optimization. In: Proceedings of the IEEE conference on decision and control, Orlando

ZigBee Alliance [Online]. Available: <http://www.zigbee.org>

Credit Risk Modeling

Tomasz R. Bielecki

Department of Applied Mathematics, Illinois
Institute of Technology, Chicago, IL, USA

Abstract

Modeling of credit risk is concerned with constructing and studying formal models of time evolution of credit ratings (credit migrations) in a pool of credit names, and with studying various properties of such models. In particular, this involves modeling and studying default times and their functionals.

Keywords

Credit risk · Credit migrations · Default time · Markov copulae

Introduction

Modeling of credit risk is concerned with constructing and studying formal models of time evolution of credit ratings (credit migrations) in a pool of N credit names (obligors), and with studying various properties of such models. In particular, this involves modeling and studying default times and their functionals. In many ways, modeling techniques used in credit risk are similar to modeling techniques used in reliability theory. Here, we focus on modeling in continuous time.

Models of credit risk are used for the purpose of valuation and hedging of credit derivatives, for valuation and hedging of counter-party risk, for

assessment of systemic risk in an economy, or for constructing optimal trading strategies involving credit-sensitive financial instruments, among other uses.

Evolution of credit ratings for a single obligor, labeled as i , where $i \in \{1, \dots, N\}$, can be modeled in many possible ways. One popular possibility is to model credit migrations in terms of a jump process, say $C^i = (C_t^i)_{t \geq 0}$, taking values in a finite set, say $\mathcal{K}^i := \{0, 1, 2, \dots, K^i - 1, K^i\}$, representing credit ratings assigned to obligor i . Typically, the rating state K^i represents the state of default of the i -th obligor, and typically it is assumed that process C^i is absorbed at state K^i .

Frequently, the case when $K^i = 1$, that is $\mathcal{K}^i := \{0, 1\}$, is considered. In this case, one is only concerned with jump from the pre-default state 0 to the default state 1, which is usually assumed to be absorbing – the assumption made here as well. It is assumed that process C^i starts from state 0. The (random) time of jump of process C^i from state 0 to state 1 is called the default time, and is denoted as τ^i . Process C^i is now the same as the indicator process of τ^i , which is denoted as H^i and defined as $H_t^i = \mathbb{1}_{\{\tau^i \leq t\}}$, for $t \geq 0$. Consequently, modeling of the process C^i is equivalent to modeling of the default time τ^i .

The ultimate goal of credit risk modeling is to provide a feasible mathematical and computational methodology for modeling the evolution of the multivariate credit migration process $\mathbf{C} := (C^1, \dots, C^N)$, so that relevant functionals of such processes can be computed efficiently. The simplest example of such functional is $P(\mathbf{C}_{t_j} \in A_j, j = 1, 2, \dots, J | \mathcal{G}_s)$, representing the conditional probability, given the information $\mathcal{G}_s$ at time $s \geq 0$, that process $\mathbf{C}$ takes values in the set A_j at time $t_j \geq 0, j = 1, 2, \dots, J$. In particular, in case of modeling of the default times $\tau^i, i = 1, 2, \dots, N$, one is concerned with computing conditional survival probabilities $P(\tau^1 > t_1, \dots, \tau^N > t_N | \mathcal{G}_s)$, which are the same as probabilities $P(H_{t_i}^i = 0, i = 1, 2, \dots, N | \mathcal{G}_s)$.

Based on that, one can compute more complicated functionals, that naturally occur in the context of valuation and hedging of credit risk-sensitive financial instruments, such as corporate

(defaultable) bonds, credit default swaps, credit spread options, collateralized bond obligations, and asset-based securities, for example.

Modeling of Single Default Time Using Conditional Density

Traditionally, there were two main approaches to modeling default times: the structural approach and the reduced approach, also known as the hazard process approach. The main features of both these approaches are presented in Bielecki and Rutkowski (2004).

We focus here on modeling a single default time, denoted as τ , using the so-called conditional density approach of El Karoui et al. (2010). This approach allows for extension of results that can be derived using reduced approach.

The default time τ is a strictly positive random variable defined on the underlying probability space $(\Omega, \mathcal{F}, P)$, which is endowed with a reference filtration, say $\mathbb{F} = (\mathcal{F}_t)_{t \geq 0}$, representing flow of all (relevant) market information available in the model, not including information about occurrence of τ . The information about occurrence of τ is carried by the (right continuous) filtration $\mathbb{H}$ generated by the indicator process $H := (H_t = \mathbb{1}_{\{\tau \leq t\}})_{t \geq 0}$. The full information in the model is represented by filtration $\mathbb{G} := \mathbb{F} \vee \mathbb{H}$.

It is postulated that

$$P(\tau \in d\theta | \mathcal{F}_t) = \alpha_t(\theta) d\theta,$$

for some random field $\alpha(\cdot)$, such that $\alpha_t(\cdot)$ is $\mathcal{F}_t \otimes \mathcal{B}(\mathbb{R}_+)$ measurable for each t . The family $\alpha_t(\cdot)$ is called $\mathcal{F}_t$ -conditional density of τ . In particular, $P(\tau > \theta) = \int_\theta^\infty \alpha_0(u) du$. The following survival processes are associated with τ ,

- $S_t(\theta) := P(\tau > \theta | \mathcal{F}_t) = \int_\theta^\infty \alpha_t(u) du$, which is an $\mathbb{F}$ -martingale,
- $S_t := S_t(t) = P(\tau > t | \mathcal{F}_t)$, which is an $\mathbb{F}$ -supermartingale (Azéma supermartingale).

In particular, $S_0(\theta) = P(\tau > \theta) = \int_\theta^\infty \alpha_0(u) du$, and $S_t(0) = S_0 = 1$.

As an example of computations that can be done using the conditional density approach we give the following result, in which notation “bd” and “ad” stand for before default and at-or-after default, respectively.

Theorem 1 *Let $Y_T(\tau)$ be a $\mathcal{F}_T \vee \sigma(\tau)$ measurable and bounded random variable. Then*

$$E(Y_T(\tau)|\mathcal{F}_t) = Y_t^{\text{bd}} \mathbb{1}_{t < \tau} + Y_t^{\text{ad}}(T, \tau) \mathbb{1}_{t \geq \tau},$$

where

$$Y_t^{\text{bd}} = \frac{\int_t^\infty Y_T(\theta) \alpha_t(\theta) d\theta}{S_t} \mathbb{1}_{S_t > 0},$$

and

$$Y_t^{\text{ad}}(T, \theta) = \frac{E(Y_T(\theta) \alpha_T(\theta) | \mathcal{F}_t)}{\alpha_t(\theta)} \mathbb{1}_{\alpha_t(\theta) > 0}.$$

There is an interesting connection between the conditional density process and the so-called default intensity processes, which are ones of the main objects used in the reduced approach. This connection starts with the following result,

Theorem 2 (i) *The Doob-Meyer (additive) decomposition of the survival process S is given as*

$$S_t = 1 + M_t^{\mathbb{F}} - \int_0^t \alpha_u(u) du,$$

where $M_t^{\mathbb{F}} = -\int_0^t (\alpha_t(u) - \alpha_u(u)) du = E(\int_0^\infty \alpha_u(u) du | \mathcal{F}_t) - 1$.

(ii) *Let $\xi := \inf\{t \geq 0 : S_{t-} = 0\}$. Define $\lambda_t^{\mathbb{F}} = \frac{\alpha_t(t)}{S_t}$ for $t < \xi$ and $\lambda_t^{\mathbb{F}} = \lambda_\xi^{\mathbb{F}}$ for $t \geq \xi$. Then, the multiplicative decomposition of S is given as*

$$S_t = L_t^{\mathbb{F}} e^{-\int_0^t \lambda_u^{\mathbb{F}} du},$$

where

$$dL_t^{\mathbb{F}} = e^{\int_0^t \lambda_u^{\mathbb{F}} du} dM_t^{\mathbb{F}}, \quad L_0^{\mathbb{F}} = 1.$$

The process $\lambda^{\mathbb{F}}$ is called the $\mathbb{F}$ intensity of τ .

The $\mathbb{G}$ -compensator of τ is the $\mathbb{G}$ -predictable increasing process $\Lambda^{\mathbb{G}}$ such that the process

$$M_t^{\mathbb{G}} = H_t - \Lambda_t^{\mathbb{G}}$$

is a $\mathbb{G}$ -martingale. If $\Lambda^{\mathbb{G}}$ is absolutely continuous, the $\mathbb{G}$ -adapted process $\lambda^{\mathbb{G}}$ such that

$$\Lambda_t^{\mathbb{G}} = \int_0^t \lambda_u^{\mathbb{G}} du$$

is called the $\mathbb{G}$ -intensity of τ . The $\mathbb{G}$ -compensator is stopped at τ , i.e., $\Lambda_t^{\mathbb{G}} = \Lambda_{t \wedge \tau}^{\mathbb{G}}$. Hence, $\lambda_t^{\mathbb{G}} = 0$ when $t > \tau$. In particular, we have

$$\lambda_t^{\mathbb{G}} = \mathbb{1}_{t < \tau} \lambda_t^{\mathbb{F}} = (1 - H_t) \lambda_t^{\mathbb{F}}.$$

The conditional density process and the $\mathbb{G}$ -intensity of τ are related as follows: For any $t < \xi$ and $\theta \geq t$ we have

$$\alpha_t(\theta) = E(\lambda_\theta^{\mathbb{G}} | \mathcal{F}_t).$$

Example 1 This is a structural-model-like example

- Suppose $\mathbb{F} = \mathbb{F}^X$ is a filtration of a default driver process, say X , and Θ is the default barrier assumed to be independent of X . Denote $G(t) = P(\Theta > t)$.
- Define

$$\tau := \inf\{t \geq 0 : \Gamma_t \geq \Theta\},$$

with $\Gamma_t := \sup_{s \leq t} X_s$. We then have $S_t(\theta) = G(\Gamma_\theta)$ if $\theta \leq t$ and $S_t(\theta) = E(G(\Gamma_\theta) | \mathcal{F}_t^X)$ if $\theta > t$

- Assume that $F = 1 - G$ and Γ are absolutely continuous w.r.t. Lebesgue measure, with respective densities f and γ . We then have

$$\alpha_t(\theta) = f(\Gamma_\theta) \gamma_\theta = \alpha_\theta, \quad t \geq \theta,$$

and $\mathbb{F}^X$ intensity of τ is

$$\lambda_t = \frac{\alpha_t(t)}{G(\Gamma_t)} = \frac{\alpha_t(t)}{S_t}.$$

- In particular, if Θ is a unit exponential r.v., that is, if $G(t) = e^{-t}$ for $t \geq 0$, then we have that $\lambda_t = \gamma_t = \frac{\alpha_t(t)}{S_t}$.

Example 2 This is a reduced-form-like example.

- Suppose S is a strictly positive process. Then, the $\mathbb{F}$ -hazard process of τ is denoted by $\Gamma^\mathbb{F}$ and is given as

$$\Gamma_t^\mathbb{F} = -\ln S_t, \quad t \geq 0.$$

In other words,

$$S_t = e^{-\Gamma_t^\mathbb{F}}, \quad t \geq 0.$$

- In particular, if $\Gamma^\mathbb{F}$ is absolutely continuous, that is, $\Gamma_t^\mathbb{F} = \int_0^t \gamma_u^\mathbb{F} du$ then

$$S_t = e^{-\int_0^t \gamma_u^\mathbb{F} du}, \quad t \geq 0 \quad \text{and}$$

$$\alpha_t(\theta) = \gamma_\theta^\mathbb{F} S_\theta, \quad t \geq \theta.$$

Modeling Evolution of Credit Ratings Using Markov Copulae

The key goal in modeling of the joint migration process $\mathbf{C}$ is that the distributional laws of the individual migration components C^i , $i \in \{1, \dots, N\}$, agree with given (predetermined) laws. The reason for this is that the marginal laws of $\mathbf{C}$, that is, the laws of C^i , $i \in \{1, \dots, N\}$, can be calibrated from market quotes for prices of individual (as opposed to basket) credit derivatives, such as the credit default swaps, and thus, the marginals of $\mathbf{C}$ should have laws agreeing with the market data.

One way of achieving this goal is to model $\mathbf{C}$ as a Markov chain satisfying the so-called Markov copula property. For brevity we present here the simplest such model, in which the reference filtration $\mathbb{F}$ is trivial, assuming additionally, but without loss of generality, that $N = 2$ and that $\mathcal{K}^1 = \mathcal{K}^2 = \mathcal{K} := \{0, 1, \dots, K\}$.

Here we focus on the case of the so-called strong Markov copula property, which is reflected in Theorem 3.

Let us consider two Markov chains Z^1 and Z^2 on $(\Omega, \mathcal{F}, P)$, taking values in a finite state space $\mathcal{K}$, and with the infinitesimal generators $A^1 := [a_{ij}^1]$ and $A^2 := [a_{hk}^2]$, respectively.

Consider the system of linear algebraic equations in unknowns $a_{ih,jk}^{\mathbf{C}}$,

$$\sum_{k \in \mathcal{K}} a_{ih,jk}^{\mathbf{C}} = a_{ij}^1, \quad \forall i, j, h \in \mathcal{K}, \quad i \neq j, \quad (1)$$

$$\sum_{j \in \mathcal{K}} a_{ih,jk}^{\mathbf{C}} = a_{hk}^2, \quad \forall i, h, k \in \mathcal{K}, \quad h \neq k, \quad (2)$$

It can be shown that this system admits at least one positive solution.

Theorem 3 Consider an arbitrary positive solution of the system (1) and (2). Then the matrix $A^{\mathbf{C}} = [a_{ih,jk}^{\mathbf{C}}]_{i,h,j,k \in \mathcal{K}}$ (where diagonal elements are defined appropriately) satisfies the conditions for a generator matrix of a bivariate time-homogeneous Markov chain, say $\mathbf{C} = (C^1, C^2)$, whose components are Markov chains in the filtration of $\mathbf{C}$ and with the same laws as Z^1 and Z^2 .

Consequently, the system (1) and (2) serves as a Markov copula between the Markovian margins C^1, C^2 and the bivariate Markov chain $\mathbf{C}$.

Note that the system (1) and (2) can contain more unknowns than the number of equations, therefore being underdetermined, which is a crucial feature for ability of calibration of the joint migration process $\mathbf{C}$ to marginal market data.

Example 3 This example illustrates modeling joint defaults using strong Markov copula theory.

Let us consider two processes, Z^1 and Z^2 , that are time-homogeneous Markov chains, each taking values in the state space $\{0, 1\}$, with respective generators

$$A^1 = \begin{matrix} & \begin{matrix} 0 & 1 \end{matrix} \\ \begin{matrix} 0 \\ 1 \end{matrix} & \begin{pmatrix} -(a+c) & a+c \\ 0 & 0 \end{pmatrix} \end{matrix} \quad (3)$$

and

$$A^2 = \begin{matrix} & \begin{matrix} 0 & 1 \end{matrix} \\ \begin{matrix} 0 \\ 1 \end{matrix} & \begin{pmatrix} -(b+c) & b+c \\ 0 & 0 \end{pmatrix} \end{matrix}, \quad (4)$$

for $a, b, c \geq 0$.

The off-diagonal elements of the matrix $A^{\mathbf{C}}$ below satisfy the system (1) and (2),

$$A^C = \begin{matrix} & \begin{matrix} (0, 0) & (0, 1) & (1, 0) & (1, 1) \end{matrix} \\ \begin{matrix} (0, 0) \\ (0, 1) \\ (1, 0) \\ (1, 1) \end{matrix} & \begin{pmatrix} -(a + b + c) & b & a & c \\ 0 & -(a + c) & 0 & a + c \\ 0 & 0 & -(b + c) & b + c \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix} \tag{5}$$

C

Thus, matrix A^C generates a Markovian joint migration process $C = (C^1, C^2)$, whose components C^1 and C^2 model individual default with prescribed default intensities $a + c$ and $b + c$, respectively.

For more information about Markov copulae and about their applications in credit risk we, refer to Bielecki et al. (2013).

Bielecki TR, Jakubowski J, Niewęglowski M (2013) Intricacies of dependence between components of multivariate Markov chains: weak Markov consistency and Markov copulae. *Electron J Probab* 18(45):1–21

Bluhm Ch, Overbeck L, Wagner Ch (2010) An introduction to credit risk modeling. Chapman & Hall, Boca Raton

El Karoui N, Jeanblanc M, Jiao Y (2010) What happens after a default: the conditional density approach. *SPA* 120(7):1011–1032

Schönbucher PJ (2003) Credit derivatives pricing models. Wiley Finance, Chichester

Summary and Future Directions

The future directions in development and applications of credit risk models are comprehensively laid out in the recent volume (Bielecki et al. 2011). One additional future direction is modeling of systemic risk.

Cross-References

- Financial Markets Modeling
- Option Games: The Interface Between Optimal Stopping and Game Theory

Bibliography

We do not give a long list of recommended reading here. That would be in any case incomplete. Up-to-date references can be found on www.defaultrisk.com.

Bielecki TR, Rutkowski M (2004) Credit risk: modeling, valuation and hedging. Springer, Berlin

Bielecki TR, Brigo D, Patras F (eds) (2011) Credit risk frontiers: subprime crisis, pricing and hedging, CVA, MBS, ratings and liquidity. Wiley, Hoboken

Cyber-Physical Maritime Robotic Systems

João Tasso de Figueiredo Borges de Sousa
Laboratório de Sistemas e Tecnologia
Subaquática (LSTS), Faculdade de Engenharia
da Universidade do Porto, Porto, Portugal

Abstract

Recent and exciting developments in robotic systems for maritime operations are presented along with projections of future applications. This will provide the background against which we will discuss new research challenges, lying at the intersection of control and computation. First, current practices and trends in robotic systems are reviewed. Second, future maritime operations are discussed with reference to a paradigm shift: from robotic systems to systems of robotic systems. Finally, the control and computation challenges posed by future robotic operations are discussed in the framework of cyber-

physical systems in which physical and computational entities co-evolve, interact, and communicate in an environment that can also be modified by the actions of these entities. This will be done with reference to modular design, systems of systems, models of systems with evolving structure, optimization and execution control for extended space-control spaces, and hierarchical control architectures.

Keywords

Autonomous underwater vehicles · Autonomous surface vehicles · Unmanned air systems · Ocean exploration · Networked vehicle systems · Robotic teams · Control hierarchies

Introduction

It took only a few decades for robotic underwater, surface, and air vehicles to revolutionize maritime operations (Bellingham and Rajan 2007; Curtin et al. 1993; Yang et al. 2018). This was driven by an application pull from science, industry, and the military, as well as by advances in computation, communication, energy storage, materials, and sensing technologies. But this is just the beginning.

Future maritime operations will involve large numbers of heterogeneous robots with significant functional diversity and endurance. Meanwhile, as robotic vehicles reach new levels of reliability and operability, operations are expected to move to remote and communications-challenged environments. Heterogeneity and functional diversity will enable new group capabilities and new concepts of operation. Robots will be organized into teams delivering capabilities that cannot be provided by isolated robots. Teams will be dynamically created, disbanded, and allocated to tasks and will cooperate with other teams, as well as with manned assets. But this will require a paradigm shift: from robotic systems to systems of robotic systems.

Here we discuss the state of the art, the current practice, trends, and operations before turning into future challenges in cyber-physical maritime robotic systems.

Background in Maritime Robotic Vehicles

Maritime robots come in several shapes, sizes, and categories. Autonomous underwater vehicles (AUVs) are unmanned and untethered submersibles. Autonomous surface vehicles (ASVs) and unmanned air vehicles (UAV) are, respectively, their surface and aerial counterparts (Fig. 1). Remotely operated vehicles (ROVs) are tethered submersibles remotely controlled by a human pilot. Finally, autonomous ships are a class on their own, mainly because of specific regulatory frameworks.

However, and despite the terminology, most of the commercially available maritime robots are not autonomous because they lack onboard decision-making capabilities. In fact, these are automated vehicles because “sensing and acting” are mediated by scripted control procedures.

Technological trends, capabilities, and future operations are better understood in terms of the product hierarchy of a maritime robot, typically comprising several subsystems: mechanical, propulsion, energy, communications, navigation, computers, payload, auxiliary equipment, and vehicle application and system software. We briefly discuss these subsystems from the cyber-physical systems viewpoint.

Energy storage and management still represent one of the main design constraints for maritime robotic systems. Cost and energy density will be positively impacted by advances in battery technology. Fuel cell technology is a potential game-changer, but there are still a few obstacles to operational deployments, especially for small vehicles. Mechanisms for harvesting energy from waves, wind, and the sun have been successfully deployed in ocean vehicles. Mechanisms for energy transfer from docking stations are under



Cyber-Physical Maritime Robotic Systems, Fig. 1 Autonomous air, underwater, and surface vehicles (from left to right: vertical take-off and landing UAV, light

autonomous underwater vehicle, Autonaut ASV (<https://www.autonautusv.com/>))

C

development and have already been deployed at sea to support, for example, resident AUVs. Energy management systems are also evolving to optimize energy consumption and power generation.

The aerodynamic and hydrodynamic performance of robotic vehicles has been improving due to advances in composite materials and mechanical design. The highly nonlinear dependence of drag on the velocity of a vehicle tends to favor the design of slow-moving vehicles, especially if mission duration is the key design parameter. For example, AUVs with buoyancy engines and active control of the center of mass, also called gliders, are now the mainstay for long-endurance data collection at sea.

Underwater and above water communications pose significant challenges to operations. Above water communications typically rely on radio technology. Future space-based Internet communications are expected to revolutionize communications at sea, mainly in remote locations. Underwater communications have relied mostly on slow acoustic communications (ranges of up to a few Km and data rates of up to a few kilobytes per second) despite recent advances in short-range optical communications (ranges of up to tens of meters and data rates in the order of hundreds of kilobytes per second). All these limitations pose significant challenges to the deployment of inter-operated communication networks (Zolich et al. 2019). This is also why delay- and disruptive-tolerant networking protocols (<https://www.nasa.gov/content/dtn>) are already enabling vehicles to transparently ferry data packets between communication nodes that are not connected.

Navigation and positioning are yet another main challenge in maritime robotics, especially for underwater vehicles. This is mainly because GPS is not available underwater. High-precision underwater navigation still relies on expensive inertial measurement units and current profilers (to measure velocity relative to the water column or the bottom of the ocean) typically aided by feature-based navigation algorithms or acoustic localization systems (that measure distances to acoustic beacons). Acoustic technologies are also a low-cost, yet range-constrained, approach to underwater localization. The combination of external navigation aids and high-precision navigation sensors is required for high-precision underwater mapping. This is not the case with ocean gliders, which typically use low-cost navigation solutions, initialized with the last GPS fix.

Current Practices and Trends

The motions of maritime robots are typically described by nonlinear models in 6 degrees of freedom (Fossen 2002). These models have a significant number of parameters to describe how forces and moments affect the motions of the robot. However, the estimation of some of these parameters is typically very expensive, thus making it prohibitive for low-cost vehicles. This also poses additional control challenges because of model uncertainty. Also, most of the ocean-going vehicles are underactuated (to reduce cost or maximize hydrodynamic performance), and control surfaces (used for steering in some vehicles) are not effective at low speeds. Moreover, state

estimation is quite difficult, not only because of nonlinear behavior and model approximations but also because accelerations and velocities are hard to measure, except with high-end navigation sensors. This is the background against which we briefly discuss selected current navigation and control practices and trends.

Acoustic navigation techniques have evolved from measuring roundtrip time of flight to multiple beacons (Whitcomb et al. 1998) to the measurement of one-way travel time, done with the help of high-precision synchronized clocks used by each beacon to send acoustic pings on a tight schedule (Eustice et al. 2011). This technique is scalable with the number of beacons and serves multiple vehicles. Another advance came with the advent of single-beacon navigation techniques and associated observability results (Batista et al. 2010). Cooperative navigation techniques improved the navigation performance of multiple vehicles by combining acoustic localization techniques and communications (Webster et al. 2013). Other developments in navigation have focused on the deterministic estimators (Bryne et al. 2017; Kinsey and Whitcomb 2007) designed with the help of Lyapunov-based stability theory, thus departing from stochastic estimators, such as the extended Kalman filter and the unscented Kalman filter (Julier and Uhlmann 2004), based on approximate minimum variance filtering.

Several variations of problems of path following and trajectory tracking for underactuated vehicles have been addressed over the years (Aguiar and Hespanha 2007; Paliotta et al. 2019). Cooperative control problems, typically expressed in terms of formation control constraints for multiple vehicles, have been formulated and solved in the framework of nonlinear control techniques (Fiorelli et al. 2006; Alessandretti and Aguiar 2020). Variations of these problems involve acoustic communication exchanges that are typically affected by time-varying delays, low data rates, and dropouts (Abreu et al. 2015; Hung and Pascoal 2018).

Multiple underwater and surface vehicles have already performed cooperative missions in world oceans (Fiorelli et al. 2006; Sato et al. 2019;

Ferreira et al. 2019; Hobson et al. 2018). These have been enabled by advances in autonomy, namely, in what regards autonomous feature tracking (Zhang and Chen 2018; Branch et al. 2019) and in statistical methods (Fossum et al. 2019). Meanwhile, several science cruises have demonstrated new methods and technologies for ocean observation. Examples include the use of ship-based unmanned air vehicles to study air-sea fluxes (Zappa et al. 2020) and the demonstration of a novel approach to study an ocean front based on the coordinated ship-robotic surveys involving autonomous underwater, surface, and air vehicles (Belkin et al. 2018).

Future Applications

The pull from future applications comes primarily from science, offshore industry, ship automation, security, and the military, as briefly discussed next.

Our oceans are in trouble. The United Nations has proclaimed a Decade of Ocean Science for Sustainable Development (2021–2030) to support efforts to reverse the cycle of decline in ocean health (<https://en.unesco.org/ocean-decade>). But this calls for a comprehensive evaluation of the health of the oceans and the development of informed ocean sustainability and sustenance policies. The problem is that the oceans are under-sampled, mainly because the traditional ship and remote-sensing methods are not cost-effective and do not provide the required spatial and temporal resolutions. This aim can only be achieved with the help of coordinated observations from space, aerial, surface, and underwater robots guided by artificial intelligence for adaptive spatial and temporal resolutions.

The offshore industry, as well as ports and coastal management, has already relied on autonomous underwater, surface, and air vehicles for routine operations. As offshore operations move to deeper waters, and complex machinery is deployed at the bottom of the ocean, the demand for new underwater inspection, monitoring, and intervention vehicles is increasing. Meanwhile,

future offshore platforms are expected to have high degrees of automation and to support interactions with robotic vehicles. For example, docking stations, delivering energy and communications, will support the operation of resident vehicles, and through-water wireless optical modems, delivering broadband communications, will enable, for example, closed-loop visual inspection with humans in the loop. In yet another concept, underwater vehicles will be autonomously launched and recovered from surface vehicles to minimize cost and expand the operational envelope.

Currently, the shipping industry is actively working on autonomous ships, with a special focus on high levels of automation and reliability. The trend is also towards systems of systems encompassing ship and harbor automation. In some future concepts, multiple robots, including autonomous ships, will support cargo transfer and other types of operations.

The military and security applications will also emphasize combined operations with large numbers of specialized assets and supporting the operation of manned assets.

Future Maritime Robotic Operations

The pull from applications and the accelerating pace of technological change will significantly shape the future landscape of maritime operations.

The number and diversity of manned and unmanned assets involved in maritime operations will increase significantly with respect to what is done today. One trend is towards systems of systems in which robotic and manned assets with complementary sensing, actuation, computation, communication, transportation, and refueling capabilities will form multi-functional teams delivering system-level capabilities that are a function of the assets, of communications networks, and of the interactions established among these assets. This will be enabled by the exchange of data and commands over interoperated underwater and above water communication networks, with support for disruptive- and

delay-tolerant networking protocols. Another significant trend is towards increased autonomy for robotic systems, as well as for systems of systems, with provisions for mixed-initiative interactions, to allow intervention in planning and control loops by experienced human operators. Yet, new opportunities also lie ahead for all-weather persistent operations in remote and communications-challenged environments (e.g., bottom of the ocean or the Arctic) (Fig. 2).

Future Directions

Here we discuss future directions, starting with robotic vehicles, as components of larger systems, before elaborating on what lies ahead for systems of robotic systems.

Robotic Vehicles

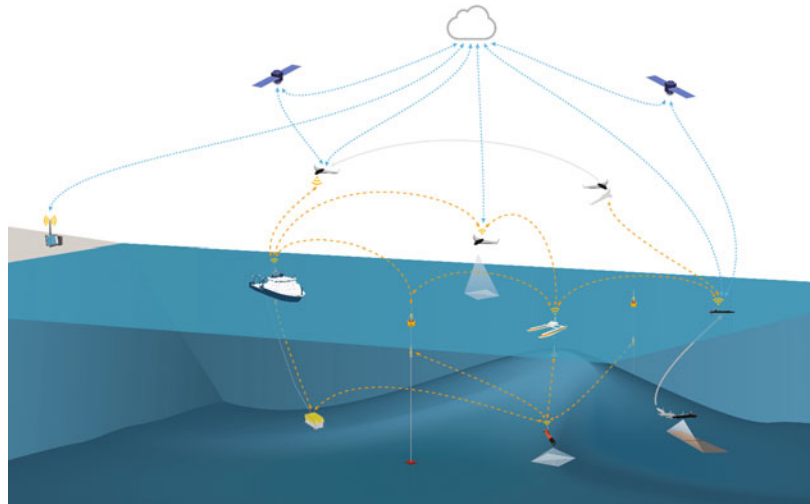
Some of the key navigation and control challenges will come from all-weather operations, autonomous launch and recovery, docking operations, interactions with infrastructures, *cloud*-based operations (Adaldo et al. 2017), and mapping and tracking of natural or man-made features.

Planning for long-endurance operations will require high-resolution ocean/weather models to minimize exposure to adverse environmental conditions or to maximize the energy harvested from waves, wind, or the sun (Aguilar et al. 2018; Lermusiaux et al. 2016). This will require a tradeoff between the need to minimize communication exchanges and the need to use cloud-based high-resolution models. A related challenge will come from active power management to balance motion, communication, and energy harvesting requirements.

Autonomous launch and recovery of robotic vehicles from unmanned or manned assets will present new relative positioning and control challenges, especially in what regards robustness and safety guarantees. These problems will become even more challenging in inspection and intervention operations because of uncontrolled interactions due to proximity to other infrastructures. Proximity will enable fast short-range light-based

Cyber-Physical Maritime Robotic Systems,

Fig. 2 Preview of future maritime robotic systems



communications to complement slower longer-range acoustic-based communications.

Cloud-based systems will support operations in remote and communications-challenged environments enabling interactions with remote vehicles (e.g., commands will be available at some cloud service for future retrieval by an autonomous underwater vehicle when it surfaces). Yet another future direction concerns the development of feature mapping and tracking capabilities (e.g., of isolines for temperature or salinity), driven by sensor readings, to study dynamic features of the ocean with adaptive spatial and temporal resolutions.

Fault identification and recovery techniques, as well as onboard risk management, will become increasingly sophisticated for robots to achieve the levels of operational reliability and safety guarantees required for operation in the world's oceans. This is especially important for autonomous ships.

Finally, robotic vehicles should be enabled for interactions to support integration into systems of systems. This poses interoperability challenges, not only for communications, command, and control but also for launch and recovery and docking mechanisms. Other key enablers for interactions are control primitives for spatial-temporal rendezvous, docking, motions in restricted spaces, path and trajectory following,

tracking of environmental features, target tracking, and formation control.

Cyber-physical Systems of Robotic Systems

The true nature of future maritime cyber-physical systems is better understood from the perspective of systems of systems. These will be cyber-physical systems in which physical entities, such as robots or control stations, and computational entities, such as computer programs, will evolve and interact enabled by mobile computation. Mobile computation, in turn, entails virtual mobility (mobile software) and physical mobility (mobile hardware), also called mobile computing (Cardelli and Gordon 1998). Physical entities are governed by the laws of physics, computational entities by the laws of computation. Physical entities evolve in the physical environment, may interact among themselves and with the environment, and can be “composed” to form other physical entities. Computational entities evolve in the computational environment formed by communicating physical entities, interact through communications, may be created and deleted on the fly, can be composed with other computational entities, and may also migrate between communicating physical entities. The mappings from computational entities to the physical entities in which they reside may change

with time, thus providing new mechanisms for reconfiguration and resilience to failures.

Models of Systems with Evolving Structure

Currently, we still lack formal modeling frameworks to support the analysis and synthesis of cyber-physical systems of systems. This is in part because research in control engineering has not yet incorporated concepts such as link, interaction, and dynamic structure, let alone concepts of mobile connectivity and mobile locality. Researchers in computer science were already developing these concepts in the early 1990s (Milner 96). The problem of mobile connectivity was addressed in the Pi-calculus (Milner 99), a calculus of communicating systems in which the components of a system may be arbitrarily linked, and communications over linked neighbors may carry information which changes that linkage. The problem of mobile locality was addressed almost one decade later, in part because of the advent of ubiquitous mobile computing, by the theory of Bi-graphical Reactive Systems (BRS's) (Milner 2009). The theory is based on the bi-graph model of mobile computation, with both mobile locality and connectivity. A bi-graph has a place graph, representing locations of computational nodes in a tree structure; a link graph, representing the interconnection of these nodes; and a set of reaction rules describing the connectivity and locality dynamics. However, BRS has several modeling limitations. This is because the reaction rules do not encompass the continuous-time mobility of physical entities and do not provide constructs for mobile locality and connectivity matching the dynamics of physical and computational entities. These limitations are not present in a model of dynamic networks of hybrid automata (DNHA) describing systems consisting of automata components that can be created, interconnected, and destroyed as the system evolves. In this model, there are two types of interactions for a hybrid automaton: (i) differential inclusions, guards, jump, and reset relations that are functions of variables from other automata and (ii) exchange of events among automata. The SHIFT language (Desh-

pande et al. 1998) implemented DNHA with formal syntax and semantics that incorporates first-order predicate logic used to dynamically reconfigure input/output connections and synchronous composition. However, synchronous composition lacks expressivity to model communications in general robotic networks, and the dynamic configuration mechanisms are just too complex for formal analysis and verification. Further research is needed into models having physical and computational entities as atomic entities, as well as coupled physical and computational dynamics.

Optimization and Execution Control in Extended Space-Control Spaces

Physical robots are abstracted by coupled energy, computation, motion, sensing, communication, and actuation capabilities. Computational entities present additional coupling constraints. Accordingly, optimization and control problems for maritime cyber-physical systems will have to be formulated in extended hybrid state-control spaces with continuous-time components, as well as discrete components, including set-valued variables (e.g., to model time-varying interactions). The concept of reach sets for these state spaces will allow the generalization of other control engineering concepts, such as invariance and optimization. In fact, several behavioral specifications can be translated into new invariance, reachability, or optimization problems in these hybrid extended state-control spaces. Conceptually, these problems can be solved in the framework of dynamic optimization, which has been extensively applied to solving optimal hybrid control problems (Branicky and Mitter 1995; Sethian and Vladimirovsky 2002).

Recent advances in the implementation of numerical methods to solve the Hamilton-Jacobi-Bellman variational inequalities (Branicky and Mitter 1995) resulting from the application of the dynamic programming principle motivated the development of modular solutions to problems in multi-vehicle planning and execution control (Aguilar et al. 2020). Further developments are also needed in this area.

Organizational Concepts

The nature of future robotic systems will go beyond analogies to human organizations. This is because physical and computational entities will evolve and interact in ways that have no parallel in human endeavors. The relevance of organizational concepts has already been demonstrated in automated highway systems – AHS (Horowitz and Varaiya 2000), mobile offshore bases (Borges de Sousa et al. 2000), mixed-initiative control of automata teams (Borges de Sousa et al. 2004), and coordinated robotics for ocean operations (Ferreira et al. 2019; Fossum et al. 2019; Belkin et al. 2018), all of which shared hierarchical organization principles (Varaiya 2000) drawing inspiration from the layered structure of software systems (e.g., the Internet protocol stack).

The AHS was a distributed and hierarchical control system that aimed to increase highway capacity, safety, and efficiency without building more roads. In this concept, a highway system is organized into a network in which sections, with limited flow capacities, connect one or more junctions, and traffic is organized into groups of tightly spaced vehicles, called platoons. The size of the platoons depends on the flow capacity of each section. The salient feature of the AHS architecture is that it breaks down AHS control into five self-contained hierarchically organized functional layers, thus providing us with a semantic framework in which we extract sub-problems from AHS and reason about them in self-contained theories.

New organization concepts involving teams, sub-teams, tasks, and subtasks were adopted in the hierarchical control of unmanned air vehicle teams for military operations (Borges de Sousa et al. 2004). Problems of task decomposition, task allocation to teams, team coordination, dynamic re-tasking, and reallocation were solved within a dynamic hierarchy of interacting team, sub-team, and maneuver controllers. Coordination among controllers was based on a few coordination variables describing the state of execution for each controller (e.g., for load balancing strategies among teams). Teams were also directly addressed by name to enable team-level command and control primitives. The structure

of controllers and the dynamics of interactions within this structure enabled this system of systems to have some degree of resiliency to the elimination of UAVs.

Further research is needed to study the relations between system-level properties, such as resilience and agility, and dynamic structure. This research would benefit from the study of the emergence of these properties in nature. For man-made systems, this would require the development of mechanisms to dynamically map teams to capabilities, to develop team controllers to deliver team capabilities, even in the presence of disturbances, and possibly through dynamic allocation of controllers to vehicles. This will require formal compositional frameworks with performance guarantees. However, the composition can only be achieved if systems are designed for interaction.

Connections between models of systems with evolving structure and control hierarchies were established in the development of a verified hierarchical control architecture for a robotic team (Borges de Sousa et al. 2007). By design, the architecture satisfies a formal specification for the behavior of the team in the sense of a bi-simulation relation between two hybrid automata, one encoding the control architecture and the other the specification.

Further investigations are needed to study behavioral equivalence for systems with evolving structure and coupled physical and computational dynamics. In this setting, the control design problem consists of deriving a structure of computational entities that, when “composed” with the system, satisfies a formal specification in some sense of behavioral equivalence.

Finally, further research in distributed artificial intelligence is required to achieve high levels of deliberation (Py et al. 2016; McGann et al. 2008) in the framework of systems with dynamic structure and functional composition.

Cross-References

- [Connected and Automated Vehicles](#)
- [Consensus of Complex Multi-agent Systems](#)

- **Control of Networks of Underwater Vehicles**
- **Information and Communication Complexity of Networked Control Systems**
- **Information-Based Multi-Agent Systems**
- **Motion Planning for Marine Control Systems**
- **Networked Systems**
- **Underactuated Marine Control Systems**
- **Underwater Robots**
- **Unmanned Aerial Vehicle (UAV)**

Recommended Reading

Allgöwer, F., Borges de Sousa, J., Kapinski, J., Mosterman, P., Oehlerking, J., Panciatici, P., Prandini, M., Rajhans, A., Tabuada, P., Wenzelburger, P. (2019) Position Paper on the Challenges Posed by Modern Applications to Cyber-physical Systems Theory. In *Nonlinear Analysis: Hybrid Systems*, Volume 34, 2019, Pages 147–165.

Bibliography

- Adaldo A, Liuzza D, Dimarogonas D, Johansson KH (2017) Coordination of multi-agent systems with intermittent access to a cloud repository. In: Fossen TI, Pettersen KY, Nijmeijer H (eds) *Sensing and control for autonomous vehicles*. Springer, pp 453–471
- Abreu PC, Bayat M, Pascoal AM, Botelho J, Góis P, Ribeiro J, Ribeiro, M, Rufino M, Sebastião L, Silva H (2015) Formation control in the scope of the MORPH project. Part II: implementation and results. In: *IFAC-PapersOnLine*, vol 48, issue 2, pp 250–255
- Aguiar AP, Hespanha JP (2007) Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. In: *IEEE transactions on automatic control*, vol 52, no 8, pp 1362–1379
- Alessandretti A, Aguiar AP (2020) An optimization-based cooperative path-following framework for multiple robotic vehicles. In: *IEEE transactions on control of network systems*, vol 7, no 2, pp 1002–1014
- Aguiar M, Borges de Sousa J, Dias JM, Silva JE, Mendes R, Ribeiro AS (2018) Trajectory optimization for underwater vehicles in time-varying ocean flows. In: *Proceedings of AUV 2018 – 2018 IEEE/OES autonomous underwater vehicle symposium*
- Aguiar M, Borges de Sousa J, Dias JM, Silva JE, Ribeiro AS, Mendes R (2020) The value function as a decision support tool in unmanned vehicle operations. In: *Proceedings of 2020 IFAC world congress*, accepted for publication
- Batista P, Silvestre C, Oliveira P (2010) Single beacon navigation: observability analysis and filter design. In: *Proceedings of the 2010 American control conference*, Baltimore, 2010, pp 6191–6196
- Bellingham JG, Rajan K (2007) Robotics in remote and hostile environments. *Science (New York)* 318: 1098–102
- Belkin IM, Borges de Sousa, J., Pinto J, Mendes R, & López-Castejón, F. (2018). Marine robotics exploration of a large-scale open-ocean front. In *Proceedings of the 2018 IEEE/OES autonomous underwater vehicle symposium*
- Borges de Sousa J, Girard AR, Hedrick K, Kretz F (2000) Real-time hybrid control of mobile offshore base scaled models. In: *Proceedings of the American control conference 2000*, Chicago, vol 1, pp 682–686
- Borges de Sousa J, Simsek T, Varaiya P (2004) Task planning and execution for UAV teams. In: *Proceedings of the IEEE conference on decision and control*, vol 4, pp 3804–3810
- Borges de Sousa J, Johansson KH, Silva J, Speranzon A (2007) A verified hierarchical control architecture for co-ordinated multivehicle operations. In: *International journal of adaptive control and signal processing*, 21(2–3), pp 159–188
- Branch A, Clark EB, Chien S et al (2019) Front delination and tracking with multiple underwater vehicles. *J Field Robot* 36:568–586
- Branicky MS, Mitter SK (1995) Algorithms for optimal hybrid control. In: *Proceedings of the IEEE conference on decision and control*, vol 3, pp 2661–2666
- Bryne TH, Hansen JM, Rogne RH, Sokolova N, Fossen TI, Johansen TA (2017) Nonlinear observers for integrated INS/GNSS navigation: implementation aspects. In: *IEEE control systems magazine*, vol 37, no 3, pp 59–86
- Cardelli L, Gordon AD (1998) Mobile ambients. In: *Lecture notes in computer science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol 1378, pp 140–155
- Curtin TB, Bellingham JG, Catipovic J, Webb D (1993) Autonomous oceanographic sampling networks. *Oceanography* 6(3):86–94
- Deshpande A, Göllü, A., Semenzato L (1998) The SHIFT programming language for dynamic networks of hybrid automata. *IEEE Trans Autom Control* 43(4): 584–587
- Eustice RM, Singh H, Whitcomb LL (2011) Synchronous-clock, one-way-travel-time acoustic navigation for underwater vehicles. *J Field Robot* 28(1):121–136
- Ferreira A, Costa M, Py F, Pinto J, Silva M, Nimmo-Smith A, Johansson T, Borges de Sousa J, Rajan K (2019) Advancing multi-vehicle deployments in oceanographic field experiments. In: *Autonomous robots*, vol 43, pp 1555–1574
- Fiorelli E, Leonard N, Bhatta P, Paley D, Bachmayer R, Fratantoni D (2006) Multi-AUV control and adaptive sampling in monterey bay. *Oceanic Eng IEEE J Oceanic Eng* 31:935–948
- Fossen TI (2002) Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles. *Marine Cybernetics*, Trondheim

- Fossum TO, Fragoso G, Davies E, Ullgren J, Mendes R, Johnsen G, Ellingsen I, Eidsvik J, Ludvigsen M, Rajan K (2019) Toward adaptive robotic sampling of phytoplankton in the coastal ocean. *Sci Robot* 4(27):eaav3041
- Hobson B, Kieft B, Raanan BY, Zhang Y, Birch JM, Ryan JP, Chavez FP (2018) An autonomous vehicle based open ocean lagrangian observatory. In: *Proceedings of the 2018 IEEE/OES autonomous underwater vehicle symposium*
- Horowitz R, Varaiya P (2000) Control design of an automated highway system. In: *Proceedings of the IEEE*, vol 88, no 7, pp 913–925
- Hung NT, Pascoal AM (2018) Cooperative path following of autonomous vehicles with model predictive control and event triggered communications. In: *IFAC-papers on line*, vol 51, no 20, 2018, pp 562–567
- Julier SJ, Uhlmann JK (2004) Unscented filtering and nonlinear estimation. *Proc IEEE* 92(3):401–422
- Kinsey JC, Whitcomb LL (2007) Model-based nonlinear observers for underwater vehicle navigation: theory and preliminary experiments. In: *Proceedings – IEEE international conference on robotics and automation*, pp 4251–4256
- Lermusiaux, P. F. J., Lolla T, Haley P, Yigit K, Ueckermann M, Sondergaard T, Leslie W (2016) Science of autonomy: Time-optimal path planning and adaptive sampling for swarms of ocean vehicles. In Tom Curtin, editor, *Springer Handbook of Ocean Engineering: Autonomous Ocean Vehicles, Subsystems and Control*, pp. 481–498.
- McGann C, Py F, Rajan K, Thomas H, Henthorn R, McEwen R (2008) A deliberative architecture for AUV control. In: *Proceedings of the 2008 IEEE international conference on robotics and automation*, Pasadena, 2008, pp 1049–1054
- Milner R (1996) Semantic ideas in computing. In: *Computing tomorrow*. Cambridge University Press, Cambridge, pp 246–283
- Milner R (1999) *Communicating and mobile systems: the π -calculus*. Cambridge University Press, Cambridge
- Milner R (2009) *The space and motion of communicating agents*. Cambridge University Press, New York
- Paliotta C, Lefeber E, Pettersen KY, Pinto J, Costa M, Borges de Sousa J (2019) Trajectory tracking and path following for underactuated marine vehicles. In: *IEEE transactions on control systems technology*, vol 27, no 4, pp 1423–1437
- Py F, Pinto J, Silva M, Johansen T, Borges de Sousa J, Rajan K (2016) EUROPlus: a mixed-initiative controller for multi-vehicle oceanographic field experiments. In: Kulić D, Nakamura Y, Khatib O, Venture G (eds) *2016 International symposium on experimental robotics*. ISER 2016. Springer proceedings in advanced robotics, vol 1. Springer, pp 323–340
- Sato T, Kim K, Inaba S, Matsuda T, Takashima S, Oono A, Takahashi D, Oota K, Takatsuki N (2019) Exploring hydrothermal deposits with multiple autonomous underwater vehicles. *IEEE Underwater Technology (UT)*, Kaohsiung, 2019, pp 1–5
- Sethian JA, Vladimirovsky A (2002) Ordered upwind methods for hybrid control. In: *Lecture notes in computer science*, vol 2289, pp 393–406
- Varaiya P (2000) A question about hierarchical systems. In: *System theory*. Springer, pp 313–324
- Yang G-Z, Bellingham J, Dupont PE, Fischer P, Floridi L, Full R, Jacobstein N, Kumar V, McNutt M, Merrifield R, Nelson BJ, Scassellati B, Taddeo M, Taylor R, Veloso M, Wang ZL, Wood R (2018) The grand challenges of science robotics. *Sci Robot* 3(14). <https://doi.org/10.1126/scirobotics.aar7650>. <https://robotics.sciencemag.org/content/3/14/ear7650>
- Zappa C, Brown SM, Laxague JM, Dhakal, T, Harris R, Farber M, Subramaniam A (2020) Using ship-deployed high-endurance unmanned aerial vehicles for the study of ocean surface and atmospheric boundary layer processes. In: *Frontiers in marine science*, vol 6
- Zhang S, Chen L (2018) Control and decision making for ocean feature tracking with multiple underwater vehicles. *J Coast Res* 83:552–564
- Zolich A, Palma D, Kansanen K, Fjortoft K, Borges de Sousa J, Johansson K, Jiang Y, Dong H, Johansen TA (2019) Survey on communication and networks for autonomous marine systems. *J Intell Robot Syst Theory Appl* 95:789–813
- Webster SE, Walls JM, Whitcomb LL, Eustice RM (2013) Decentralized extended information filter for single-beacon cooperative acoustic navigation: theory and experiments. *IEEE Trans Robot* 29(4):957–974
- Whitcomb L, Yoerger D, Singh H (1998) Towards precision robotic maneuvering, survey and manipulation in unstructured undersea environments. In: *Proceedings of the international symposium on robotics research*, pp 45–54

Cyber-Physical Security

Henrik Sandberg
Division of Decision and Control Systems,
School of Electrical Engineering and Computer
Science, KTH Royal Institute of Technology,
Stockholm, Sweden

Abstract

Recent cyberattacks targeting critical infrastructures, such as power systems and nuclear facilities, have shown that malicious, advanced attacks on control and monitoring systems are a serious concern. These systems rely on tight integration of cyber and physical

components, implying that cyberattacks may have serious physical consequences. This has led to an increased interest for Cyber-Physical Systems (CPS) security, and we here review some of the fundamental problems considered. We review basic modeling of the CPS components in networked control systems and how to model a cyberattack. We discuss different classes of attack strategies and the resources required. Particular attention is given to analysis of detectability and impact of attacks. For the system operator, attack mitigation is a particular concern, and we review some common strategies. Finally, we point out future directions for research.

Keywords

Attack mitigation · Attack modeling · CPS · Denial-of-service attack · Detectability · Eavesdropping attack · Impact · Replay attack · Stealth attack · Undetectable attack

Introduction

Control systems are essential elements in many critical infrastructures and govern the interaction of the cyber and physical components. A modern control system contains application programs running on a host computer or embedded devices and constitutes an important instance of CPS. Historically, cybersecurity has not been a concern for these systems. However, recent incidents have clearly illustrated that malicious corruption of signals in control and monitoring systems is a legitimate, serious attack vector. Several malwares dedicated to attacking control and monitoring systems in nuclear facilities and power systems, for example, have been discovered (Hemsley and Fisher 2018). Security has therefore in the last decade become an important topic on the research agenda of control system designers (Cardenas 2019). Unlike traditional security research, the development of security-enhancing mechanisms that leverage the control system itself is key. These mechanisms do not eliminate the need for traditional security

technologies, such as encryption or authentication, but should rather be seen as complements that are able to mitigate attack vectors that are hard, or even impossible, to detect on the lower protocol layers.

Three fundamental properties of information and services in IT systems are analyzed in the cybersecurity literature (Bishop 2002) using the acronym CIA: *confidentiality* (C), *integrity* (I), and *availability* (A). Confidentiality concerns the concealment of data, ensuring it remains known to the authorized parties alone. Integrity relates to the trustworthiness of data, meaning there is no unauthorized change to the information between the source and destination. Availability considers the timely access to information or system functionalities. These properties are relevant in CPS security as well, and our review will use them for classification of attack strategies.

Networked Control System Under Attack

CPS Components and Signals

In Fig. 1, a block diagram of a networked control system is shown. Networked control systems are special instances of CPS, containing strongly interacting components in both the cyber and physical domains. The signals $\tilde{u}$ (input to plant P) and y (output from plant P) can be thought of as physical quantities, where the physical effects of CPS attacks will be realized and sensed. These signals are sampled and communicated over a computer network, possibly under (partial) control of a malicious attacker; see section “CPS Attack Space.” The sampled signals u and $\tilde{y}$ are outputs and inputs to the feedback controller C . Note that in the presence of an attacker in the network, the signals may be different from their plant-side counterparts $\tilde{u}$ and y . It is commonly assumed that an anomaly detector D is co-allocated with the controller, although the detector may run on a different platform. The role of the detector is to generate alarms to alert the operator of unusual system behavior, caused by faults or attacks, for instance. Often the detectors are model-based fault-detection-type filters (see

Blanke et al. 2015, for instance), which sometimes can detect also malicious cyberattacks.

CPS Attack Space

In CPS security, it is commonly assumed cyberattacks are carried out through the network, as illustrated in Fig. 1, with the overall intention of disrupting the CPS. The attacker has three general modes of operation: By listening to the network traffic ($c_u = u$ and/or $c_y = y$), the attacker violates the confidentiality (property **C**) of the system. By modifying the content of the network traffic ($i_u \neq 0$ and/or $i_y \neq 0$), the attacker violates the integrity (property **I**) of the system. Finally, by dropping communication packets ($a_u = \text{OPEN}$ and/or $a_y = \text{OPEN}$), the attacker violates the availability (property **A**) of the system. Other modes of cyberattack, for instance, assuming that the attacker is able to modify controller or sensor software directly, may also be of interest. From a practical IT perspective, such attacks may be very different from the network-based attacks illustrated in Fig. 1. On the other hand, from a high-level control, physical impact, and detectability perspective, one can often model also such attacks by an appropriate choice of the signals $c_u, c_y, i_u, i_y, a_u, a_y$.

To compare different attack strategies, a CPS attack space was used in Teixeira et al. (2015), see Fig. 2. The three axes quantify the level of resources the attacker possesses.

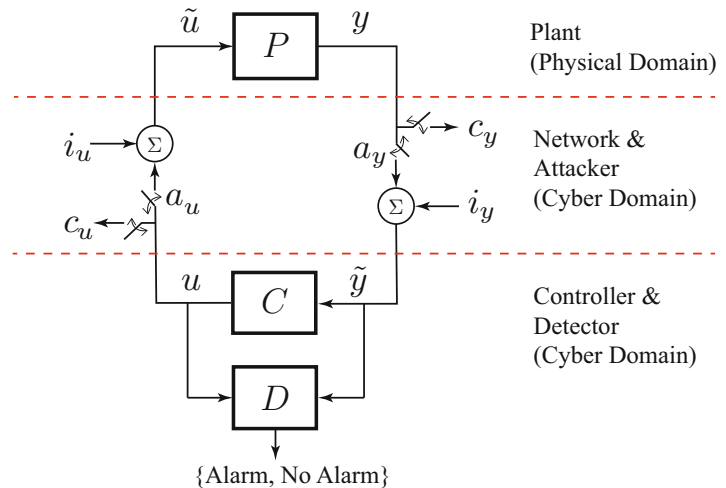
Disclosure resources measure how many network communication signals the attacker can listen to. That is, an attacker that can read all elements of signals y and u is more powerful than an attacker with only occasional access to some elements. *Disruption resources* measure the number of network signals the attacker can alter through integrity or availability attacks. Finally, *CPS model knowledge* quantifies the accuracy of the models of system components the attacker has access to. Some commonly considered attack strategies are mapped in Fig. 2 and are discussed in more detail next.

Eavesdropping attacks (see Leong et al. 2018, for instance) involve an attacker who is not actively altering data, but rather listens and records network traffic ($c_u = u$ and/or $c_y = y$). Such attacks are of high importance in privacy-sensitive applications, but could also be a precursor to physically damaging attacks. The attacker could carry out system identification or machine learning on recorded data (u, y) to obtain the CPS model knowledge required for other attack strategies.

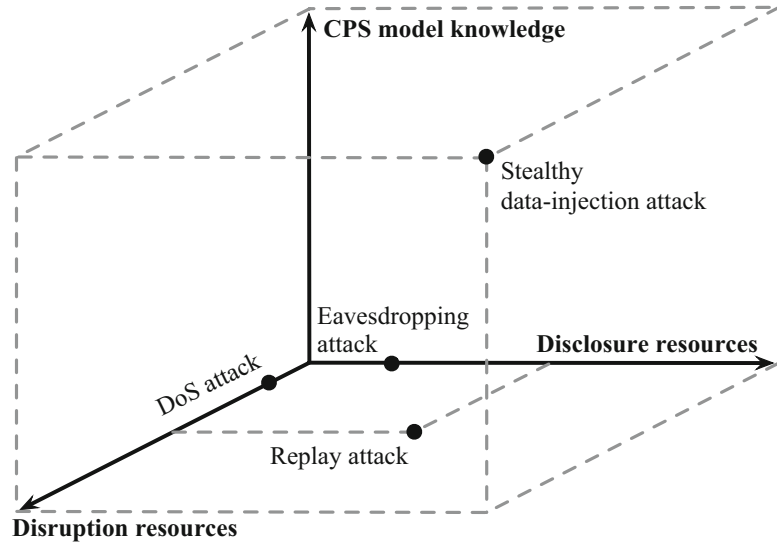
Replay attacks (see Mo and Sinopoli 2009, for instance) start with the recording of data, similar to an eavesdropping attack. In the second stage of the attack, the attacker replays captured outputs ($a_y = \text{OPEN}$ and $i_y = y_{\text{old}}$) while he is applying harmful control commands on the plant ($i_u \neq 0$).

Cyber-Physical Security,

Fig. 1 Block diagram of networked control system. Through the network communication system, the attacker may be able to intercept and corrupt the communication between the controller (C) and the physical plant (P). It is common in the CPS security area to assume that the operator has an anomaly detector (D) installed



Cyber-Physical Security, Fig. 2 The CPS attack space (Teixeira et al. 2015) maps out different attack strategies along three axes, each measuring the relative amount of attack resources required



Stealthy data-injection attacks (see Teixeira et al. 2015, for instance) involve an attacker with significant CPS model knowledge. The attacker exploits this knowledge to create integrity attacks ($i_u \neq 0$ and/or $i_y \neq 0$) so that the disrupted signals ($\tilde{u}, \tilde{y}$) appear similar to regular signals and do not trigger alarms in the anomaly detector. Note that the attacker here does not simply repeat old captured signals but actually generates new carefully crafted signals that may be more challenging to deem anomalous.

Denial-of-service (DoS) attacks (see Amin et al. 2009, for instance) involve an attacker that disrupts the communication between the controller and the plant ($a_u = \text{OPEN}$ and/or $a_y = \text{OPEN}$) by dropping packets in the communication network. There are often natural losses of packets in communication networks depending on the implementation, and so the detectability and physical impact depend on the exact attack strategy.

Properties of Attack Strategies

To determine the cyberattack vulnerability of a networked control system, it is important to understand the possibility for a CPS operator to *detect* an ongoing cyberattack, as well as the attack's possible physical *impact*.

Detectability

Undetectable attacks (see Pasqualetti et al. 2013, for instance) are attacks whose impact on the detector signals ($u, \tilde{y}$) are such that they could have resulted from an unknown initial state in the plant P (without the attacker present). An integrity attack signal (i_u, i_y) is undetectable if the measurement satisfies

$$\tilde{y}_t(x_0, i_u, i_y) = \tilde{y}_t(\bar{x}_0, 0, 0), \quad (1)$$

for all times t and some possibly different plant initial states $x_0, \bar{x}_0$. These attacks may be very hard, or even impossible, for the detector to catch since the measurement exactly corresponds to unperturbed physical behavior of the plant. The possibility of undetectable attacks is equivalent to the closed-loop system having non-trivial zero dynamics, and sometimes these attacks are referred to as *zero-dynamics attacks*. If the plant is linear, dangerous destabilizing undetectable attacks exist when the plant has unstable invariant zeros (by injecting exponentially growing i_u) or unstable poles (by injecting exponentially growing i_y).

A statistically based anomaly detector could check if the received signals ($u, \tilde{y}$) are likely or not, and not only if they exactly correspond to possible unperturbed plant behavior (1). Such a detector could raise an alarm also when “unde-

teable” attacks are staged. An attacker who would like to remain hidden may thus want to include a model of the anomaly detector during the attack design.

Stealth attacks (see Teixeira et al. 2015, for instance) are attacks that are designed such that a particular detector does not trigger an alarm. The attacker needs to have a model of the detector and its thresholds to design these attacks. For detectors that only check whether the signals $(u, \tilde{y})$ satisfy the unperturbed plant dynamics, undetectable attacks constitute examples of stealth attacks. But generally it is possible for stealth attacks to deviate from the plant dynamics quite a bit without generating alarms, since otherwise the detectors would generate too many false alarms in practice.

For linear stochastic plants with realization (A, B, C, D) and state x , a common anomaly-detector strategy is to monitor the innovation signal $r = \tilde{y} - C\hat{x}$, where $\hat{x}$ is the optimal state estimate. Under the null hypothesis that there is no attacker present, the innovation signal is an independent stochastic signal with known statistics. The detector may implement various change detection tests to generate alerts (Basseville 1988). A *stateless* test (Giraldo et al. 2018) raises an alarm every time t that

$$\|r_t\| \geq \tau, \quad (2)$$

where τ is a tunable threshold. A *stateful* non-parametric CUMulative SUM (CUSUM) test (Giraldo et al. 2018) raises an alarm at time t when

$$S_t \geq \tau, \quad (3)$$

where $S_{t+1} = \max(0, S_t + \|r_t\| - \delta)$, $S_0 = 0$. The forgetting factor δ is selected such that the expected value of $\|r_t\| - \delta$ is negative under the null hypothesis. In these tests, the operator should choose the parameters τ and δ such that a reasonable trade-off between the true and false alarm rates is achieved. Because of the possibility of stealth attacks, the true positive alarm rate may very well be zero, however, and tuning could be difficult (see Giraldo et al. (2018) and section “[Impact](#)”). The survey (Basseville 1988)

provides an overview of alternative (parametric) change detection tests. These are often optimal but require more detailed alternative (fault or attack) hypotheses than needed in (2)–(3).

To obtain fundamental limits on detectors, Bai et al. (2017) proposed to study the Kullback-Leibler divergence between the integrity-attacked measurement sequence $\tilde{y}_0^{t-1}(x_0, i_u, i_y)$ and the un-attacked (null hypothesis) sequence $\tilde{y}_0^{t-1}(x_0, 0, 0)$:

$$\lim_{t \rightarrow \infty} \frac{1}{t} D_{KL}(\tilde{y}_0^{t-1}(x_0, i_u, i_y) \parallel \tilde{y}_0^{t-1}(x_0, 0, 0)) \leq \epsilon. \quad (4)$$

The idea is that if ϵ is small, there is no detector test that can simultaneously achieve a high true alarm rate and a low false alarm rate.

If the attacker is aware of the applied detector test statistic, such as (2)–(3) above, he can generate stealth attacks more powerful than the more conservative undetectable attacks. The replay attacks, the stealthy data-injection attacks, and sometimes DoS attacks, discussed in section “[CPS Attack Space](#),” are often designed in this manner.

Impact

In section “[Detectability](#),” we discussed when cyberattacks are hard to detect by the operator. This is an important problem since if the operator is not alerted an attack is in progress, the CPS may sustain severe damage. Indeed, an attacker generally has a goal of disrupting some service in the CPS, and not only to remain undetected. In this vein, it becomes important for both operator and attacker to quantify the maximum possible impact of an attack, without the attacker being detected. For example, if one plots the maximum possible (physical) impact of a stealth attack versus the mean time between false alarms, one obtains a trade-off metric that can be used for tuning detector tests (Giraldo et al. 2018). To compute the maximum possible attack impact on the plant state, subject to not triggering an alarm (2)–(3), tools from optimal control theory can be applied, see, for instance, Mo et al. (2010), Cárdenas et al. (2011), and Teixeira et al. (2015).

Attack Mitigation Strategies

Attack mitigation strategies for the CPS operator are a very active area of research. We will here mention a few general approaches related to the attack strategies discussed in section “[CPS Attack Space](#).”

To prevent *eavesdropping attacks*, encryption and coding methods may be used to protect the information sent over the network. Homomorphic encryption has been proposed as suitable for control systems, (see Kim et al. (2016) and Farokhi et al. (2017)), since it allows for direct computation on encrypted data. Encryption relies on secret keys, whereas coding schemes are generally publicly known but transmit information in a manner that makes it difficult for an eavesdropper to recover the complete data; see Miao et al. (2017) and Wiese et al. (2019), for instance. Another approach is to intentionally corrupt the transmitted data before communication over the network (Leong et al. 2018). A popular privacy notion, deriving from the database literature, is differential privacy; see Cortés et al. (2016). Differential privacy is often achieved by intentionally injecting Laplacian or Gaussian noise.

To mitigate *replay attacks*, the operator may superimpose a watermark signal on the control input and check that its expected effect is present in the received measurement; see Mo et al. (2015), for instance. In a related manner, a multiplicative watermarking signal can be introduced on the sensor output and then removed on the controller side; see Ferrari and Teixeira (2017).

Several approaches have been proposed to counter *stealthy data-injection attacks*. Using the trade-off metric mentioned in section “[Impact](#),” it may be possible to tune detectors such that possible undetected attack impact is at an acceptable level (Giraldo et al. 2018). If this is not possible, additional sensors and signal authentication may be installed to obtain more favorable detector trade-offs. A more reactive form of mitigation is to install so-called secure state estimators (Fawzi et al. 2014). If the operator has enough redundant sensors to ensure undetectable attacks are not possible (see section “[Detectability](#)”), the secure

state estimator can always reconstruct a data-injection attack and compute a correct state estimate. Yet another form of defense is so-called moving target defense (Weerakkody and Sinopoli 2015). The idea is to decrease the attacker’s CPS model knowledge (see Fig. 2) by varying the system’s parameters over time to counter possible system identification by the attacker.

To mitigate the effect of *DoS attacks*, game-theoretic (Ugrinovskii and Langbort 2017), optimal control (Amin et al. 2009), and event-triggered control (Cetinkaya et al. 2017; De Persis and Tesi 2015) approaches have been proposed, for instance.

Summary and Future Directions

We have given a brief introduction to the area of cyber-physical security. We have discussed modeling of cyberattacks targeting networked control systems and how their confidentiality, integrity, and availability may be violated using different attack strategies. Some common attack strategies, namely, eavesdropping attacks, replay attacks, stealthy data-injection attacks, and DoS attacks, were covered in more detail. In allocating defense resources, it is important for the system operator to assess the detectability and impact of attacks, and we discussed some available tools for this. Finally, we gave a brief overview of available attack mitigation strategies.

CPS security is a relatively new research area in the systems and control community. Hence, there are many promising future research directions. It is noteworthy that there are few research papers strongly combining ideas from IT security and control engineering. The reason is most likely that IT security tends to focus on cyber aspects and the use of discrete mathematics, whereas control engineering tends to focus on physical aspects and continuous-time and continuous-state models. However, work that seriously combines ideas from both domains would likely have large impact, also outside of their traditional communities. A promising example along this line is the use of homomorphic encryption in feedback systems.

Another promising area for interdisciplinary work is machine learning and control engineering. Machine learning may be used for data-driven detection and diagnosis of anomalies and attacks. Machine learning is becoming increasingly societally relevant, but there are also privacy issues worth investigation, as well as concerns about vulnerability to integrity attacks. A definite concern is also that many machine learning methods come with few formal guarantees, which are highly desirable in safety-critical CPS applications. In general, investigation of fundamental trade-offs between performance criteria is an interesting problem in CPS security. One example is the trade-off between privacy, security, and control performance when the operator as defense is deliberately injecting noise (“watermarking”) or actively changing the system (“moving target defense”).

Finally, the areas of fault detection, fault-tolerant control, and change detection are well established in control engineering, and many tools developed there have inspired work in CPS security. Nevertheless, there are certainly more connections to be made, for instance, with regard to the diagnosis of root causes of faults and attacks. A key difference between faults and attacks is that the latter are caused by an intelligent agent with a usually malicious incentive. In contrast, faults are generally random and uncoordinated, which potentially make them easier to detect and mitigate.

Cross-References

- [Fault Detection and Diagnosis](#)
- [Fault-Tolerant Control](#)
- [Networked Control Systems: Architecture and Stability Issues](#)
- [Privacy in Network Systems](#)

Recommended Reading

The CPS security knowledge area report (Cardenas 2019) provides a broad and readable introduction to the area, with many references also

to work outside of control engineering. A more control-theoretically focused contribution is the special issue (Sandberg et al. 2015), which contains six articles covering basic topics in CPS security. An early experimental case study on CPS security focusing on control system aspects is Amin et al. (2013).

One of the first papers taking a control-theory approach to analyzing cyberattacks on CPS, namely, replay attacks, is Mo and Sinopoli (2009). The paper Pasqualetti et al. (2013) has been influential and establishes basic concepts such as undetectable attacks (section “[Detectability](#)”) and attack detection mechanism inspired by fault diagnosis tools. The paper Sundaram and Hadjicostis (2011) is another early influential paper, which focuses on distributed attack detection for networks of interconnected nodes. The paper Teixeira et al. (2015) provides a general modeling framework for attacks and defenses in networked control systems (Figs. 1 and 2). The article Fawzi et al. (2014) introduces the secure state estimator as a mitigation strategy, which has been extended in several follow-up works. Finally, a risk-based tutorial introduction to CPS security, with many references to recent work, is Chong et al. (2019).

Bibliography

- Amin S, Cárdenas AA, Sastry SS (2009) Safe and secure networked control systems under denial-of-service attacks. In: International workshop on hybrid systems: computation and control. Springer, pp 31–45
- Amin S, Litrico X, Sastry S, Bayen AM (2013) Cyber security of water SCADA systems-Part I: analysis and experimentation of stealthy deception attacks. *IEEE Trans Control Syst Technol* 21(5):1963–1970
- Bai CZ, Pasqualetti F, Gupta V (2017) Data-injection attacks in stochastic control systems: detectability and performance tradeoffs. *Automatica* 82:251–260
- Basseville M (1988) Detecting changes in signals and systems: a survey. *Automatica* 24(3):309–326
- Bishop M (2002) Computer security: art and science. Addison-Wesley Professional, USA
- Blanke M, Kinnaert M, Lunze J, Staroswiecki M (2015) Diagnosis and fault-tolerant control, 3rd edn. Springer
- Cardenas AA (2019) Cyber-physical systems security. Knowledge area report, CyBOK
- Cárdenas AA, Amin S, Lin ZS, Huang YL, Huang CY, Sastry S (2011) Attacks against process control systems: risk assessment, detection, and response. In: Pro-

- ceedings of the 6th ACM symposium on information, computer and communications security, ASIACCS'11. ACM, New York, pp 355–366
- Cetinkaya A, Ishii H, Hayakawa T (2017) Networked control under random and malicious packet losses. *IEEE Trans Autom Control* 62(5):2434–2449
- Chong MS, Sandberg H, Teixeira AMH (2019) A tutorial introduction to security and privacy for cyber-physical systems. In: 2019 18th European control conference (ECC), pp 968–978
- Cortés J, Dullerud GE, Han S, Le Ny J, Mitra S, Pappas GJ (2016) Differential privacy in control and network systems. In: 2016 IEEE 55th conference on decision and control (CDC). IEEE, pp 4252–4272
- De Persis C, Tesi P (2015) Input-to-state stabilizing control under denial-of-service. *IEEE Trans Autom Control* 60(11):2930–2944
- Farokhi F, Shames I, Batterham N (2017) Secure and private control using semi-homomorphic encryption. *Control Eng Pract* 67:13–20
- Fawzi H, Tabuada P, Diggavi S (2014) Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Trans Autom Control* 59(6):1454–1467
- Ferrari RM, Teixeira AM (2017) Detection and isolation of replay attacks through sensor watermarking. *IFAC-PapersOnLine* 50(1):7363–7368
- Giraldo J, Urbina D, Cardenas A, Valente J, Faisal M, Ruths J, Tuppenhauer NO, Sandberg H, Candell R (2018) A survey of physics-based attack detection in cyber-physical systems. *ACM Comput Surv* 51(4):76:1–76:36
- Hemsley KE, Fisher RE (2018) History of industrial control system cyber incidents. Technical Report INL/CON-18-44411-Rev002, Idaho National Lab. (INL), Idaho Falls
- Kim J, Lee C, Shim H, Cheon JH, Kim A, Kim M, Song Y (2016) Encrypting controller using fully homomorphic encryption for security of cyber-physical systems. *IFAC-PapersOnLine* 49(22):175–180
- Leong AS, Redder A, Quevedo DE, Dey S (2018) On the use of artificial noise for secure state estimation in the presence of eavesdroppers. In: 2018 European control conference (ECC), pp 325–330
- Miao F, Zhu Q, Pajic M, Pappas GJ (2017) Coding schemes for securing cyber-physical systems against stealthy data injection attacks. *IEEE Trans Control Netw Syst* 4(1):106–117
- Mo Y, Sinopoli B (2009) Secure control against replay attacks. In: 2009 47th annual Allerton conference on communication, control, and computing, Allerton, pp 911–918
- Mo Y, Garone E, Casavola A, Sinopoli B (2010) False data injection attacks against state estimation in wireless sensor networks. In: 49th IEEE conference on decision and control (CDC), pp 5967–5972
- Mo Y, Weerakkody S, Sinopoli B (2015) Physical authentication of control systems: designing watermarked control inputs to detect counterfeit sensor outputs. *IEEE Control Syst* 35(1):93–109
- Pasqualetti F, Dörfler F, Bullo F (2013) Attack detection and identification in cyber-physical systems. *IEEE Trans Autom Control* 58(11):2715–2729
- Sandberg H, Amin S, Johansson KH (2015) Cyberphysical security in networked control systems: an introduction to the issue. *IEEE Control Syst Mag* 35(1):20–23
- Sundaram S, Hadjicostis CN (2011) Distributed function calculation via linear iterative strategies in the presence of malicious agents. *IEEE Trans Autom Control* 56(7):1495–1508
- Teixeira A, Shames I, Sandberg H, Johansson KH (2015) A secure control framework for resource-limited adversaries. *Automatica* 51(1):135–148
- Ugrinovskii V, Langbort C (2017) Controller–jammer game models of denial of service in control systems operating over packet-dropping links. *Automatica* 84:128–141
- Weerakkody S, Sinopoli B (2015) Detecting integrity attacks on control systems using a moving target approach. In: 2015 54th IEEE conference on decision and control (CDC). IEEE, pp 5820–5826
- Wiese M, Oechtering TJ, Johansson KH, Papadimitratos P, Sandberg H, Skoglund M (2019) Secure estimation and zero-error secrecy capacity. *IEEE Trans Autom Control*. 64(3):1047–1062

Cyber-Physical Systems Security: A Control-Theoretic Approach

Yuqing Ni, Ziyang Guo, and Ling Shi
Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong, China

Abstract

This entry introduces security issues in cyber-physical systems mainly from two perspectives: one is to analyze the attack effectiveness from the malicious attacker's perspective and the other is to design the attack detection mechanism and control scheme from the system designer's perspective. For the first research aspect, existing works focused on the worst-case performance degradation analysis when the system is under a certain type of attack are introduced, among which *DoS attack* and *integrity attack* are of the

most interest. For the second research aspect, existing works providing countermeasures to detect the existence of attacks and proposing resilient estimation and control methodologies against hostile attacks are introduced. The main framework of cyber-physical systems is provided, and solid results about security issues are reviewed from selected high-quality journal papers in this entry.

Keywords

Cyber-physical system security · DoS attack · Integrity attack · Countermeasure mechanism design

Introduction

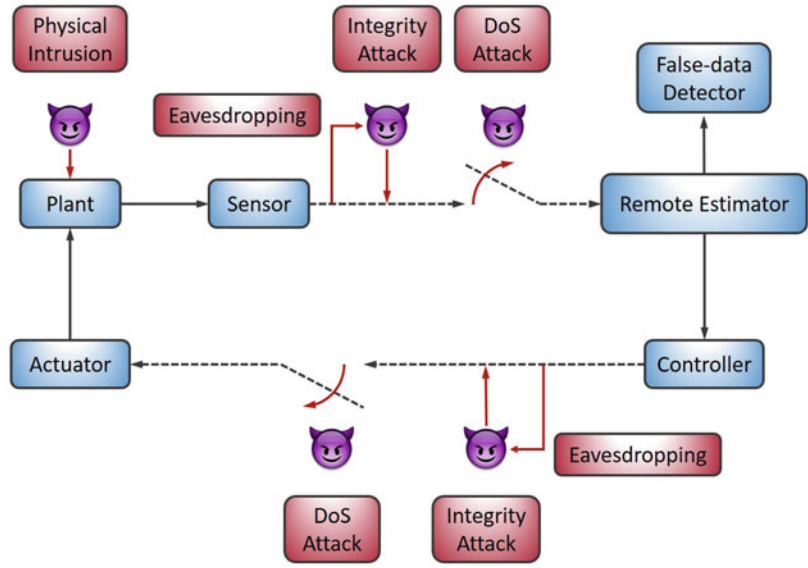
Cyber-physical systems (CPSs), which coordinate sensing, computing, communication, and control methods with physical plants, have drawn an enormous amount of attention in the most recent couple of years. A wide scope of applications include power grids, wireless sensor networks, autonomous vehicles, intelligent transportation systems, healthcare and medical systems, environment monitoring systems, etc. (Kim and Kumar 2012). In most CPSs, with the recent advances in wireless technologies, tiny wireless sensors play an indispensable role, well-known for their low cost, easy installation, self-power, and inherent intelligent processing capability (Gungor and Hancke 2009). The wireless communication has been replacing the traditional wired communication between system components, which fundamentally upgrades the system's mobility, scalability, and feasibility.

However, along with the advantages, CPSs are confronted with critical security challenges. The nature of wireless communication and the complex interconnection between the physical layer and the cyber layer make CPSs vulnerable to potential threats. An incident caused by the attack on one system in a single infrastructure at Maroochy Water Services in Queensland is described in Slay and Miller (2007). The terrible Stuxnet virus infected around 100,000

computers mostly in Iran, India, Indonesia, and Pakistan and even showed up in an Iranian nuclear plant (Farwell and Rohozinski 2011). Such security-related incidents suggest that any successful attack may result in disastrous consequences on the national economy, social security, or even human lives. Therefore, the security is of significant importance to guarantee the reliable operation of CPSs.

To evaluate the information security, the established CIA (confidentiality, integrity, and availability) triad is utilized as a benchmark model (Andress 2014). Confidentiality means that the private information is not allowed to be accessed by unauthorized users. Integrity is the property that the data should be accurate and complete. Availability indicates that information should always be available when it is required for the service purpose. According to the system model knowledge, disclosure resources, and disruption resources of the adversary (Teixeira et al. 2015), *eavesdropping*, *denial-of-service (DoS) attack*, *integrity attack*, and physical intrusion which compromise data confidentiality, integrity, or availability are main security threats to the CIA triad in CPSs as shown in Fig. 1. There have been many approaches developed to security based on the features of the CPSs mainly from two different perspectives: one is to analyze the attack effectiveness from the malicious attacker's point of view and the other is to design the attack detection mechanism and the corresponding secure estimation and control scheme from the system designer's perspective. For the first research aspect, existing works mostly concentrate on the worst-case performance degradation analysis when a system is under a certain type of attack, among which *DoS attack* and *integrity attack* are of the most interest. For the second research aspect, most existing works provide countermeasures to detect the existence of attacks and propose robust estimation and control methods against hostile attacks. The detection schemes can be divided into the attack-dependent schemes and the attack-independent schemes. The countermeasure designed specifically for one type of attack is called attack-dependent. Otherwise, it is attack-

Cyber-Physical Systems Security: A Control-Theoretic Approach, Fig. 1 Attack types in cyber-physical systems



independent since it is applicable to different attack scenarios. The complicated situations make it extremely difficult for a system designer to predict the specific attack type beforehand. Consequently, different methodologies should be adopted to design the attack-independent detection mechanism, e.g., the information-theoretic, the control-theoretic, and the optimization-based approaches.

Future research directions of CPSs security are abundant including but not limited to proposing more accurate and practical attack/defend models and analyzing attacker/defender's behavior subject to limited or partial knowledge.

The remainder of the entry is divided into two main sections discussing the attack strategies and defense strategies, respectively.

Notations: $\mathbb{N}$ denotes the set of nonnegative integers. $\mathbb{R}$ denotes the set of real numbers. $\mathbb{R}^n$ is the n -dimensional Euclidean space. For $x \in \mathbb{R}^n$, $\|x\|$ denotes its ℓ_2 norm. $X^\top$, $\text{Tr}(X)$, $\rho(X)$, and X^{-1} stand for the transpose, trace, spectral radius, and inverse of matrix X . $\text{Diag}\{\cdot\}$ denotes the block diagonal matrix. $\mathbb{S}_+^n$ ($\mathbb{S}_{++}^n$) is the set of $n \times n$ positive semi-definite (definite) matrices. When $X \in \mathbb{S}_+^n$ ($\mathbb{S}_{++}^n$), it can be denoted by $X \succeq 0$ ($X \succ 0$). δ_{ij} is the Dirac delta function: $\delta_{ij} = 1$ if $i = j$; $\delta_{ij} = 0$ otherwise. For functions f , f_1 , f_2 , $f_1 \circ f_2$ is defined as $f_1 \circ f_2(X) \triangleq f_1(f_2(X))$, and

f^k is defined as $f^k(X) \triangleq \underbrace{f \circ f \cdots f}_{k \text{ times}}(X)$ with $f^0(X) = X$.

In CPSs, the physical plant is described by a standard discrete-time linear time-invariant (LTI) process:

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad (1)$$

$$y_k = Cx_k + v_k, \quad (2)$$

where $k \in \mathbb{N}$ is the time index, $x_k \in \mathbb{R}^n$ is the process state vector, $u_k \in \mathbb{R}^q$ is the control input, $y_k \in \mathbb{R}^m$ is the measurement collected by the sensor, $w_k \in \mathbb{R}^n$, and $v_k \in \mathbb{R}^m$ are zero-mean i.i.d. Gaussian noises with $\mathbb{E}[w_i w_j^\top] = \delta_{ij} Q$ ($Q \succeq 0$), $\mathbb{E}[v_i v_j^\top] = \delta_{ij} R$ ($R \succ 0$), $\mathbb{E}[w_i v_j^\top] = 0, \forall i, j \in \mathbb{N}$. The initial state x_0 is zero-mean Gaussian with covariance matrix $\Pi_0 \succeq 0$, which is independent of w_k and v_k for all $k \in \mathbb{N}$. Assume that the pair (A, C) is observable and $(A, \sqrt{Q})$ is controllable.

Attack Effectiveness Analysis

DoS Attack

The *DoS attack* aims at blocking the communication channel and preventing legitimate access

between system components. The adversary can either emit a high-strength radio signal to flood the targeted channel or violate network protocols to cause packet collisions (Xu et al. 2006). Since jamming is a power-intensive activity and the available energy or attack capability of a jammer might be limited, *DoS attacks* on estimates for the resource-constrained attacker are investigated in Zhang et al. (2015), Qin et al. (2018), Li et al. (2015a, b), and Ding et al. (2017).

With the development of manufacturing techniques, many modern sensors are equipped with extra functions, e.g., signal processing, decision making, and alarm monitoring, beyond those necessary for generating the measured quantity (Lewis 2004; Hovareshti et al. 2007; Frank 2013). A smart sensor with computational abilities is deployed to pre-estimate the state x_k based on the gathered raw measurement y_k as shown in Fig. 2. The minimum mean squared error (MMSE) estimate $\hat{x}_k^s \triangleq \mathbb{E}[x_k | y_0, y_1, \dots, y_k]$ and the corresponding error covariance

$$\hat{P}_k^s \triangleq \mathbb{E}[(x_k - \hat{x}_k^s)(x_k - \hat{x}_k^s)^\top | y_0, y_1, \dots, y_k]$$

can be calculated by a standard Kalman filter. The Kalman filtering analysis shows that the error covariance converges to a unique positive semi-definite matrix $\bar{P}^s$ from any initial condition (Bertsekas 1995). In practice, the Kalman gain usually converges in a few steps. For brevity, it is assumed that $\hat{P}_0^s = \bar{P}^s$. Define the Lyapunov operator $h : \mathbb{S}_+^n \rightarrow \mathbb{S}_+^n$ as $h(X) \triangleq AXA^\top + Q$. It is shown that the optimal state estimate $\hat{x}_k$ and the corresponding error covariance $\hat{P}_k$ at the remote estimator have the following recursive form according to the estimate $\hat{x}_k^s$ received from the smart sensor (Schenato 2008):

$$\hat{x}_k = \begin{cases} \hat{x}_k^s, & \text{if } \hat{x}_k^s \text{ is received at the remote} \\ & \text{estimator,} \\ A\hat{x}_{k-1}, & \text{otherwise,} \end{cases} \quad (3)$$

$$\hat{P}_k = \begin{cases} \bar{P}^s, & \text{if } \hat{x}_k^s \text{ is received at the} \\ & \text{remote estimator,} \\ h(\hat{P}_{k-1}), & \text{otherwise.} \end{cases} \quad (4)$$

In Zhang et al. (2015), the attacker's decision variable $\gamma_k = 1$ or 0 means launching a *DoS attack* or not at time k . If a *DoS attack* is launched, the data packet transmitted from the smart sensor to the remote estimator will be dropped with probability $0 < \alpha < 1$. The problem of interest for the attacker is to determine an attack strategy $\{\gamma_k\}$ under some energy constraint, such that the average expected estimation error covariance matrix at the remote estimator is maximized during a finite time horizon T . The energy constraint only enables the attacker to launch T_0 attacks from $k = 1$ to $k = T$ where $T_0 < T$. The optimization problem is given by:

Problem 1

$$\begin{aligned} \max_{\{\gamma_k\}} J &= \frac{1}{T} \sum_{k=1}^T \text{Tr}(\mathbb{E}[\hat{P}_k]), \\ \text{s.t. } \sum_{k=1}^T \gamma_k &= T_0. \end{aligned}$$

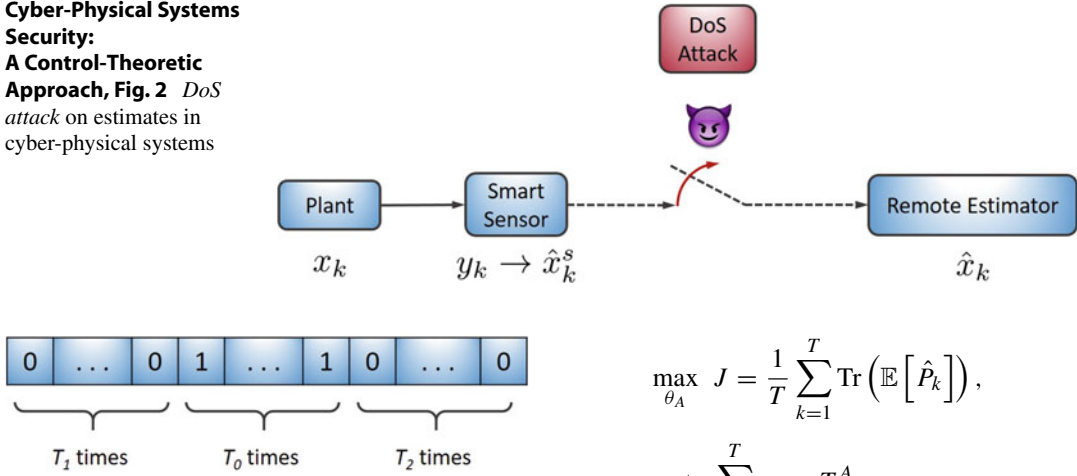
It is proved in Zhang et al. (2015) that the optimal attack strategy is to launch *DoS attacks* consecutively for T_0 times which can start at any time in the interval $[1, T - T_0 + 1]$, and the corresponding cost is as follows:

$$\begin{aligned} J_{\max} &= \frac{1}{T} \sum_{i=1}^{T_0} \left[(T_0 - i) (\alpha^i - \alpha^{i+1}) + \alpha^i \right] \\ &\quad \text{Tr}(h^i(\bar{P}^s)) + \frac{T - \alpha T_0}{T} \text{Tr}(\bar{P}^s). \end{aligned} \quad (5)$$

Furthermore, a packet-dropping network is considered in Qin et al. (2018). When there is not any *DoS attack*, the data packet transmitted from the smart sensor to the remote estimator will be dropped with probability β , where $0 < \beta < \alpha < 1$. The optimal attack solution to Problem 1 is characterized by Fig. 3, where $T_1 + T_2 = T - T_0$, and

Cyber-Physical Systems**Security:****A Control-Theoretic****Approach, Fig. 2** *DoS*

attack on estimates in
cyber-physical systems



Cyber-Physical Systems Security: A Control-Theoretic Approach, Fig. 3 Optimal *DoS* attack with dropping network

$$\begin{aligned} \max_{\theta_A} J &= \frac{1}{T} \sum_{k=1}^T \text{Tr} \left(\mathbb{E} \left[\hat{P}_k \right] \right), \\ \text{s.t. } \sum_{k=1}^T \gamma_k &= T_0^A. \end{aligned}$$

$|T_1 - T_2| \leq 1$, which is a little different from the optimal solution in Zhang et al. (2015).

Instead of focusing only on the attacker, game-theoretic approaches are adopted to investigate the interactive decision-making process between the DoS attacker and the system in Li et al. (2015b). The DoS attacker's jamming strategy is denoted as $\theta_A \triangleq \{\gamma_1, \gamma_2, \dots, \gamma_T\}$ and $\sum_{k=1}^T \gamma_k \leq T_0^A$. Similarly, the sensor's data-sending strategy is denoted as $\theta_S \triangleq \{\lambda_1, \lambda_2, \dots, \lambda_T\}$ and $\sum_{k=1}^T \lambda_k \leq T_0^S$. It is assumed that the dropout probability of data packet from the smart sensor to the remote estimator is β in the absence of attack and is α ($> \beta$) under the *DoS* attack. The problem of interest in a game-theoretic framework is given by:

Problem 2 For the sensor,

$$\begin{aligned} \min_{\theta_S} J &= \frac{1}{T} \sum_{k=1}^T \text{Tr} \left(\mathbb{E} \left[\hat{P}_k \right] \right), \\ \text{s.t. } \sum_{k=1}^T \lambda_k &= T_0^S. \end{aligned}$$

For the attacker,

It is proved in Li et al. (2015b) that there exists a Nash equilibrium for the considered two-player zero-sum game between the sensor and the attacker. For the case where the two players have online information about the previous transmission outcomes and the occurrence of *DoS* attacks, an algorithm is provided to perform the game dynamically.

A Markov game between the sensor and the attacker with multiple channels under a signal-to-interference-and-noise-ratio (SINR)-based network model is studied in Ding et al. (2017), and a modified Nash Q-learning algorithm is applied to obtain the Nash equilibrium. In Ding et al. (2019), an asymmetric-information-structured game framework is constructed to study the interaction between a sensor and a DoS attacker. It is assumed that the acknowledgment information from the remote estimator to the sensor is hidden from the DoS attacker. The game is converted into a continuous-state one and solved by the multi-agent reinforcement learning method.

Integrity Attack

Integrity attack is also called deception attack. The studies of *integrity attack* can be divided into the control attack analysis and the measurement attack analysis as shown in Fig. 4. Three main *integrity attack* types will be introduced: *replay attack*, *false-data injection*

attack, and *innovation-based integrity attack*. Besides, there are many other *integrity attack* types, e.g., GPS spoofing, time synchronization attack, reset attack, and so on.

For the system, a linear quadratic Gaussian (LQG) controller minimizing the following objective function is considered:

$$J_{LQG} = \min_{\{u_k\}} \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[\sum_{k=0}^{T-1} \left(x_k^\top W x_k + u_k^\top U u_k \right) \right], \quad (6)$$

where $W \succeq 0$, $U \succeq 0$, and u_k are measurable with respect to $y_0, \dots, y_k$. The optimal solution is a fixed gain controller: $u_k = L\hat{x}_k$. It is assumed that the pair (A, B) is controllable and $(A, \sqrt{W})$ is observable. Besides, a false-data detector is deployed at the estimator side to monitor system behavior and detect the existence of potential malicious attacks. Note that innovation $z_k = y_k - C\hat{x}_k^-$, where $\hat{x}_k^- \triangleq A\hat{x}_{k-1}^s$, has a steady-state Gaussian distribution $\mathcal{N}(0, \Sigma)$. A χ^2 detector (Mehra and Peschon 1971) makes a decision based on the sum of the normalized innovation sequence, i.e., at time k , the detection criterion follows a hypothesis testing:

$$g_k = \sum_{i=k-D+1}^k z_i^\top \Sigma^{-1} z_i \underset{H_1}{\overset{H_0}{\leq}} \sigma, \quad (7)$$

where D is the detection window size, σ is the threshold, and the null hypothesis H_0 stands for the normal operation of the system, while the alternative hypothesis H_1 means that the system is under attack.

Replay Attack

As one of the *integrity attacks*, the *replay attack* does not require any system knowledge. It attempts to inject exogenous control inputs while repeating historical measurements to maintain stealthiness to the false-data detector. In Mo and Sinopoli (2009), it is assumed that the attacker can inject a control input into the system anytime, and it knows all the measurements, with the ability to modify them at the same time. After recording a sufficient number of measurements without giving any input to the system, the

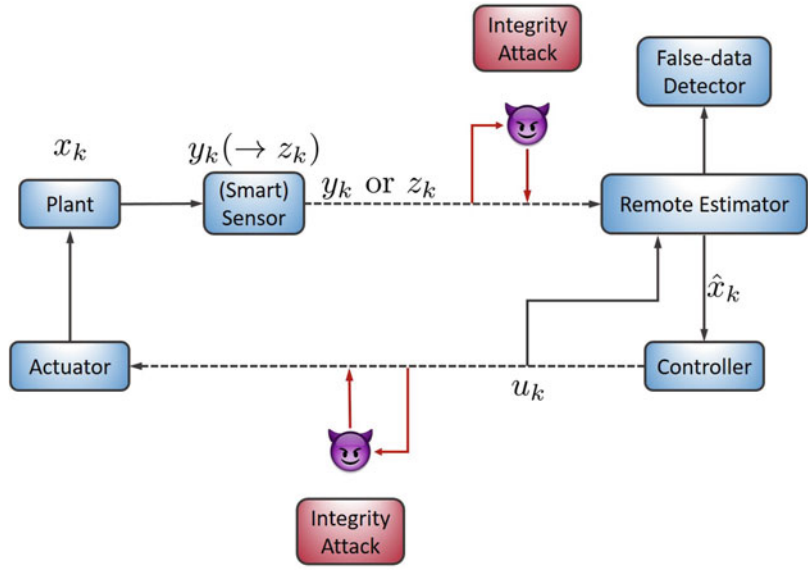
attacker can launch a *replay attack* by injecting a sequence of desired control input while replaying the previously recorded measurements. The χ^2 detector turns out to be useless for the system if $(A + BL)(I - KC)$ is stable, where K is the steady Kalman filter gain, which means that the detection rate of the system converges to the false alarm rate.

False-Data Injection Attack

The *false-data injection attack* needs to be launched with full system knowledge. In Mo and Sinopoli (2010), a control system with a Kalman filter, an LQG controller, and a χ^2 detector with the detection window size $D = 1$ is considered. It is assumed that the attacker knows system matrices A, B, C, Q , and R and the Kalman gain K and the LQG gain L , where $A - KCA$ and $A + BL$ are both stable. As its name implies, the *false-data injection attack* is launched by injecting a malicious input y_k^a from time $k = 0$:

$$y_k = Cx_k + v_k + \Theta y_k^a, \quad (8)$$

where $\Theta \triangleq \text{Diag}\{\theta_1, \dots, \theta_m\}$ is the sensor selection matrix and θ_i s are binary variables. Define Δx_k , Δz_k , and Δe_k as the state, the innovation, and the estimation error differences between the partially compromised system and the healthy system, respectively. A control system is *perfectly attackable* if there exists an attack sequence $\{y_k^a\}$ such that $\limsup_{k \rightarrow \infty} \|\Delta x_k\| = \infty$, $\|\Delta z_k\| \leq 1$ for all $k \in \mathbb{N}$. Intuitively, it implies that the *false-data injection attack* could destabilize the system while successfully bypassing a large set of possible false-data detectors. A necessary and sufficient condition

Cyber-Physical Systems**Security:****A Control-Theoretic****Approach,****Fig. 4** Integrity attack in cyber physical systems

for a system to be *perfectly attackable* is derived as follows:

Theorem 1 A control system (1) is *perfectly attackable* if and only if A has an unstable eigenvalue and the corresponding eigenvector v satisfies:

1. $Cv \in \text{span}(\Theta)$, where $\text{span}(\Theta)$ is the column space of Θ ;
2. v is a reachable state of the dynamic system $\Delta e_{k+1} = (A - KCA)\Delta e_k - K\Theta y_{k+1}^a$.

Similarly in Hu et al. (2018), a system is called *insecure* if there exists an attack sequence $\{y_k^a\}$ such that $\lim_{k \rightarrow \infty} \|\Delta \hat{x}_k\| \rightarrow \infty$ and $\|\Delta z_k\|$ is upper bounded by a prescribed small positive constant scalar for all $k \in \mathbb{N}$, where $\Delta \hat{x}_k$ is the estimate difference between the partially compromised system and the healthy system. It is proved that the system is *insecure* if and only if the spectral radius of A is not less than 1, under the assumption that the attacker can inject malicious input to all the sensors, i.e., $\Theta = I$.

In Mo and Sinopoli (2016), an adversary attempts to induce perturbation in CPSs by compromising a subset of the sensors and injecting an exogenous control input while remaining undetected by the false-data detector. To analyze the control system's estimation

performance degradation under the *false-data injection attack*, a constrained control problem is formulated. The characterization of the maximum perturbation induced is posed as a reachable set computation problem, which can be solved by ellipsoidal calculus.

A *false-data injection attack* with control objectives in analyzed in Chen et al. (2018). The CPS model under the attack is as follows:

$$x_{k+1} = Ax_k + Bu_k + w_k + \Gamma d_k, \quad (9)$$

$$y_k = Cx_k + v_k + \Psi d_k, \quad (10)$$

where $d_k \in \mathbb{R}^s$ is the attack injection, and matrices Γ and Ψ model the attacker. The problem of interest is to drive the system state to a target value x^* while remaining undetected from a χ^2 detector to the best of the attacker's ability. The attacker's quadratic cost function is a linear combination of components penalizing the distance from the target state and the energy of the system's detection statistic. A closed-form optimal attack solution $\{d_k\}$ is obtained by dynamic programming.

Besides the χ^2 detector, the Kullback-Leibler (KL) divergence, a nonnegative measure of the distance between two probability distributions, is also well-known in detection

theory. *False-data injection attack* against the KL divergence detector is studied in Bai et al. (2015, 2017a, b) and Kung et al. (2017). Specifically, an ϵ -stealthy *false-data injection attack* on the control input u_k for first-order control systems is studied in Bai et al. (2015), where the KL divergence between the compromised innovation and the true innovation, i.e., $\mathcal{D}(\tilde{z}_1^k \| z_1^k)$, is adopted as a stealthiness metric. For high-dimensional dynamic control systems, more advanced results are reported in Bai et al. (2017a) and Kung et al. (2017). In Bai et al. (2017a), the optimal ϵ -stealthy attack on the control input which induces the largest estimation performance degradation is obtained for right invertible systems, and a suboptimal attack solution is obtained for right non-invertible systems. In Kung et al. (2017), the achievable upper bound of the average estimation error covariance by an ϵ -stealthy attack is derived when the observation matrix $C \in \mathbb{R}^{m \times n}$ is full rank and $m \geq n$. When C is full rank and $m < n$, there exists an ϵ -stealthy attack such that the induced average estimation error covariance can be arbitrarily large. A *false-data injection attack* on measurement y_k for first-order systems under the ϵ -stealthy metric is investigated in Bai et al. (2017b).

Innovation-Based Integrity Attack

The *integrity attacks* mentioned above are all focused on control input u_k and measurement y_k . Integrity attacks on innovation z_k are introduced as follows. An optimal linear *innovation-based integrity attack* is proposed in Guo et al. (2017). It considers a remote estimation scenario where a sensor measures a linear dynamic system and transmits its local innovation z_k to a remote estimator through a wireless network. There exists a malicious attacker who is able to intercept and modify the transmitted z_k . A χ^2 false-data detector is deployed at the remote estimator side to check data anomalies. A specific affine attack strategy is $\tilde{z}_k = T_k z_k + b_k$ where $T_k \in \mathbb{R}^{m \times m}$ is an arbitrary matrix and $b_k \in \mathbb{R}^m$ is a zero mean i.i.d. Gaussian random variable with covariance L_k and independent of z_k . Since the original innovation z_k has a steady-state Gaussian distribution $\mathcal{N}(0, \Sigma)$, $\tilde{z}_k$ is supposed to satisfy the

same Gaussian distribution $\mathcal{N}(0, \Sigma)$ to bypass the false-data detector, i.e.,

$$T_k \Sigma T_k^\top + L_k = \Sigma, \quad (11)$$

which is equivalent to $L_k = \Sigma - T_k \Sigma T_k^\top \geq 0$. It is proved in Guo et al. (2017) that the optimal stealthy attack strategy which yields the largest remote estimation error covariance is $T_k^* = -I$ and $b_k^* = 0$. Intuitively, the estimation error covariance is maximized when the attacker just inverts the innovation sequence. Besides, the ϵ -stealthy innovation-based affine *integrity attack* against the KL divergence is analyzed in Guo et al. (2018). Denote the estimation error covariance at the remote estimator under such affine *innovation-based attack* as $\tilde{P}_k$ and the problem of interest for the attacker is as follows:

Problem 3

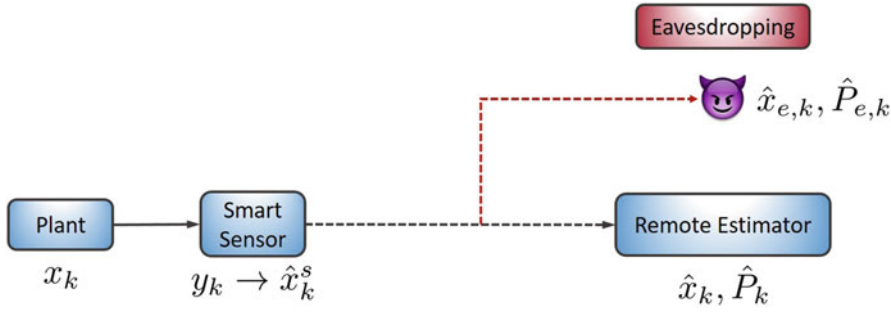
$$\begin{aligned} \max_{T_k, b_k} \quad & \text{Tr}(\tilde{P}_k) \\ \text{s.t.} \quad & \mathcal{D}(\tilde{z}_k \| z_k) \leq \epsilon, \quad \forall k. \end{aligned}$$

By solving a two-stage optimization problem, the worst-case linear attack strategy is proved to be zero-mean Gaussian distributed.

Countermeasure Mechanism Design

As mentioned above, one way to investigate the security in CPSs is to focus on specific types of attacks, and the other way is to design detection schemes and build attack-resilient systems.

A transmission scheduling for remote state estimation in the presence of an eavesdropper is proposed in Leong et al. (2019). As shown in Fig. 5, let $\lambda_k \in \{0, 1\}$ be system's decision variables such that $\lambda_k = 1$ if and only if $\hat{x}_k^s$ is to be transmitted at time k . If the local estimate $\hat{x}_k^s$ is transmitted, it will be successfully received by the remote estimator with probability ζ and overheard by the eavesdropper with probability ζ_e , where the stochastic transmission processes for both the remote estimator and the eavesdropper are mutually independent. The problem of



Cyber-Physical Systems Security: A Control-Theoretic Approach, Fig. 5 Remote estimation with an eavesdropper in cyber-physical systems

interest for the system designer is to minimize the expected error covariance $\hat{P}_k$ at the remote estimator while trying to keep the expected error covariance $\hat{P}_{e,k}$ at the eavesdropper above a certain level:

Problem 4

$$\min_{\{\lambda_k\}} \sum_{k=1}^T \mathbb{E} \left[\varepsilon \text{Tr}(\hat{P}_k) - (1 - \varepsilon) \text{Tr}(\hat{P}_{e,k}) \right],$$

for some $\varepsilon \in (0, 1)$. In the case where the transmission decision λ_k depends on the error covariances of both the remote estimator $\hat{P}_{k-1}$ and the eavesdropper $\hat{P}_{e,k-1}$, the optimal solution to Problem 4 is a nondecreasing threshold policy on $\hat{P}_{k-1}$ for fixed $\hat{P}_{e,k-1}$. Suppose that A is unstable and that $\zeta > 1 - \frac{1}{\rho^2(A)}$. Then it is also proved in Leong et al. (2019) that for any $0 < \zeta_e < 1$, there exist transmission policies in the infinite horizon situation such that $\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=1}^T \text{Tr}(\mathbb{E}[\hat{P}_k])$ is bounded and $\liminf_{T \rightarrow \infty} \frac{1}{T} \sum_{k=1}^T \text{Tr}(\mathbb{E}[\hat{P}_{e,k}])$ is unbounded. Intuitively, the system designer can make the estimation error covariance at the eavesdropper diverge while keeping its own estimation error covariance bounded, which shows the defense ability of the system designer to the *eavesdropping*. Besides preserving information from the malicious eavesdropper, sometimes each sensor should also prevent undesirable disclosure of information to the other sensors in the network, which is called the privacy. In Mo and Murray (2017), a privacy preserving average consensus

algorithm is proposed to guarantee the privacy of the initial state and asymptotic consensus on the exact average of the initial values, by introducing random noises to the consensus process.

For the *replay attack* proposed in Mo and Sinopoli (2009), a physical watermarking strategy is introduced as a countermeasure, and the optimal watermark authentication signal is derived in Mo et al. (2015), and the trade-off between estimation quality and detection rate is analyzed under a stochastic game-theoretic framework in Miao et al. (2013). For noiseless dynamic systems under *false-data injection attacks* on control inputs or measurements, Fawzi et al. (2014) characterizes the maximum number of attacks that can be detected and corrected. It is shown that if more than half the sensors are corrupted, it is impossible to accurately reconstruct the system's state. The results are further extended to systems with noise and modeling errors in Pajic et al. (2014) and Mishra et al. (2017). Since locating the sensors under *false-data injection attacks* is essentially a combinatorial problem, a satisfiability modulo theory-based method for secure state estimation is proposed in Shoukry et al. (2017), where the scalability, soundness, and completeness of the proposed algorithm are analyzed. In Shi et al. (2018), transfer entropy-based causality countermeasures are investigated for both sensor measurements and innovation sequences. The countermeasures are shown to be related to the system parameters, based on which the convergence condition is analyzed and the effectiveness is evaluated. In Mo and Garone

(2016), for the scenario where a subset of the sensors are under *false-data injection attacks*, a convex optimization-based data fusion approach is adopted to combine the local estimates into a more secure state estimation. A more general convex optimization-based estimator with an ℓ_1 penalty is considered in Han et al. (2016), and necessary and sufficient conditions for resilience are analyzed with a trivial gap. A stochastic modeling framework for secure state estimation is introduced in Shi et al. (2017), based on which the attacked system is modeled as a finite-state hidden Markov model. Under this framework, a joint state and attack estimation problem are formulated and solved. In Guo et al. (2019), a modified Gaussian mixture model (GMM)-based detection algorithm is proposed to locate the compromised sensors and obtain an accurate state estimate in a situation where a subset of sensors are under *integrity attacks* from a malicious attacker. The algorithm can autonomously cluster the sensors and fuse the measurements based on different beliefs. For the *innovation-based integrity attack* introduced in Guo et al. (2017) with multiple sensors, three sequential data verification and fusion procedures for different detection information scenarios based on the information extracted from the trusted sensors and correlations among the sensors are proposed in Li et al. (2019).

Cross-References

- [Cyber-Physical Security](#)
- [Game Theory for Security](#)

Bibliography

- Andress J (2014) The basics of information security: understanding the fundamentals of InfoSec in theory and practice, 2nd edition. Syngress Media, Waltham, MA
- Bai C-Z, Pasqualetti F, Gupta V (2015) Security in stochastic control systems: fundamental limitations and performance bounds. In: Proceedings of the American control conference, pp 195–200
- Bai C-Z, Pasqualetti F, Gupta V (2017a) Data-injection attacks in stochastic control systems: detectability and performance tradeoffs. *Automatica* 82:251–260
- Bai C-Z, Gupta V, Pasqualetti F (2017b) On Kalman filtering with compromised sensors: attack stealthiness and performance bounds. *IEEE Trans Autom Control* 62(12):6641–6648
- Bertsekas DP (1995) Dynamic programming and optimal control, vol 1. Athena Scientific, Belmont
- Chen Y, Kar S, Moura JMF. Cyber-physical attacks with control objectives. *IEEE Trans Autom Control* 63(5):1418–1425
- Ding K, Li Y, Quevedo DE, Dey S, Shi L (2017) A multi-channel transmission schedule for remote state estimation under DoS attacks. *Automatica* 78:194–201
- Ding K, Ren X, Quevedo DE, Dey S, Shi L (2019) DoS attacks on remote state estimation with asymmetric information. *IEEE Trans Control Netw Syst* 6(2):653–666
- Farwell JP, Rohozinski R (2011) Stuxnet and the future of cyber war. *Survival* 53(1):23–40
- Fawzi H, Tabuada P, Diggavi S (2014) Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Trans Autom Control* 59(6):1454–1467
- Frank R (2013) Understanding smart sensors. Artech House, Norwood, MA
- Gungor VC, Hancke GP (2009) Industrial wireless sensor networks: challenges, design principles, and technical approaches. *IEEE Trans Ind Electron* 56(10):4258–4265
- Guo Z, Shi D, Johansson KH, Shi L (2017) Optimal linear cyber-attack on remote state estimation. *IEEE Trans Control Netw Syst* 4(1):4–13
- Guo Z, Shi D, Johansson KH, Shi L (2018) Worst-case stealthy innovation-based linear attack on remote state estimation. *Automatica* 89:117–124
- Guo Z, Shi D, Quevedo DE, Shi L (2019) Secure state estimation against integrity attacks: a Gaussian mixture model approach. *IEEE Trans Signal Process* 67(1):194–207
- Han D, Mo Y, Xie L (2016) Resilience and performance analysis for state estimation against integrity attacks. *IFAC-PapersOnLine* 49(22):55–60
- Hovareshti P, Gupta V, Baras JS (2007) Sensor scheduling using smart sensors. In: Proceedings of IEEE conference on decision and control, pp 494–499
- Hu L, Wang Z, Han Q, Liu X (2018) State estimation under false data injection attacks: security analysis and system protection. *Automatica* 87:176–183
- Kim KD, Kumar PR (2012) Cyber-physical systems: a perspective at the centennial. *Proc IEEE* 100:1287–1308
- Kung E, Dey S, Shi L (2017) The performance and limitations of ϵ -stealthy attacks on higher order systems. *IEEE Trans Autom Control* 62(2):941–947
- Leong AS, Quevedo DE, Dolz D, Dey S (2019) Transmission scheduling for remote state estimation over packet dropping links in the presence of an eavesdropper. *IEEE Trans Autom Control* 64(9):3732–3739
- Lewis FL (2004) Wireless sensor networks. In: Smart environments: technologies, protocols, and applications, pp 11–46

- Li Y, Quevedo DE, Dey S, Shi L (2015a) Fake-acknowledgment attack on ACK-based sensor power schedule for remote state estimation. In: Proceedings of IEEE conference on decision and control, pp 5795–5800
- Li Y, Shi L, Cheng P, Chen J, Quevedo DE (2015b) Jamming attacks on remote state estimation in cyber-physical systems: a game-theoretic approach. *IEEE Trans Autom Control* 60(10):2831–2836
- Li Y, Shi L, Chen T (2019) Detection against linear deception attacks on multi-sensor remote state estimation. *IEEE Trans Control Netw Syst* 5(3):846–856
- Mehra RK, Peschon J (1971) An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica* 7(5):637–640
- Miao F, Pajic M, Pappas GJ (2013) Stochastic game approach for replay attack detection. In: Proceedings of IEEE conference on decision and control, pp 1854–1859
- Mishra S, Shoukry Y, Karamchandani N, Diggavi SN, Tabuada P (2017) Secure state estimation against sensor attacks in the presence of noise. *IEEE Trans Control Netw Syst* 4(1):49–59
- Mo Y, Garone E (2016) Secure dynamic state estimation via local estimators. In: Proceedings of IEEE conference on decision and control, pp 5073–5078
- Mo Y, Murray RM (2017) Privacy preserving average consensus. *IEEE Trans Autom Control* 62(2):753–765
- Mo Y, Sinopoli B (2009) Secure control against replay attacks. In: Proceedings of annual Allerton conference on communication, control, and computing (Allerton), pp 911–918
- Mo Y, Sinopoli B (2010) False data injection attacks in control systems. In: Proceedings of 1st workshop on secure control systems, pp 1–6
- Mo Y, Sinopoli B (2016) On the performance degradation of cyber-physical systems under stealthy integrity attacks. *IEEE Trans Autom Control* 61(9):2618–2624
- Mo Y, Weerakkody S, Sinopoli B (2015) Physical authentication of control systems: designing watermarked control inputs to detect counterfeit sensor outputs. *IEEE Control Syst Mag* 35(1):93–109
- Pajic M, Weimer J, Bezzi N, Tabuada P, Sokolsky O, Lee I, Pappas GJ (2014) Robustness of attack-resilient state estimators. In: Proceedings of international conference on cyber-physical systems, pp 1163–1174
- Qin J, Li M, Shi L, Yu X (2018) Optimal denial-of-service attack scheduling with energy constraint over packet-dropping networks. *IEEE Trans Autom Control* 63(6):1648–1663
- Schenato L (2008) Optimal estimation in networked control systems subject to random delay and packet drop. *IEEE Trans Autom Control* 53(5):1311–1317
- Shi D, Elliott RJ, Chen T (2017) On finite-state stochastic modeling and secure estimation of cyber-physical systems. *IEEE Trans Autom Control* 62(1):65–80
- Shi D, Guo Z, Johansson KH, Shi L (2018) Causality countermeasures for anomaly detection in cyber-physical systems. *IEEE Trans Autom Control* 63(2):386–401
- Shoukry Y, Nuzzo P, Puggelli A, Sangiovanni-Vincentelli AL, Seshia SA, Tabuada P (2017) Secure state estimation for cyber-physical systems under sensor attacks: a satisfiability modulo theory approach. *IEEE Trans Autom Control* 62(10):4917–4932
- Slay J, Miller M (2007) Lessons learned from the Maroochy Water breach. In: Proceedings of the international conference on critical infrastructure protection, pp 73–82
- Teixeira A, Shames I, Sandberg H, Johansson KH (2015) A secure control framework for resource-limited adversaries. *Automatica* 51:135–148
- Xu W, Ma K, Trappe W, Zhang Y (2006) Jamming sensor networks: attack and defense strategies. *IEEE Netw* 20(3):41–47
- Zhang H, Cheng P, Shi L, Chen J (2015) Optimal denial-of-service attack scheduling with energy constraint. *IEEE Trans Autom Control* 60(11):3023–3028

Cyber-Physical-Human Systems

Anuradha M. Annaswamy¹ and Yildiray Yildiz²

¹Active-adaptive Control Laboratory,
Department of Mechanical Engineering,
Massachusetts Institute of Technology,
Cambridge, MA, USA

²Department of Mechanical Engineering,
Bilkent University, Ankara, Turkey

Abstract

Cyber-physical-human systems (CPHS) is an emerging field where the human, the physical system, and enabling cyber technologies are interconnected through complex interactions to accomplish a certain goal. This sharply contrasts with a conventional perspective where the human is treated as an isolated element who operates or uses the system. In this entry, we provide a general introduction to CPHS and present representative examples from the recent literature. CPHS is a growing field, and currently there are no agreed upon definitions about what constitutes a CPHS or not. Therefore, the entry mainly reflects the authors' current understanding of the topic.

Keywords

Cyber-physical-human systems ·
Decision-making · Human-machine
interaction

Introduction

Systems and control has always been a broad and abstract discipline. It has progressed, thrived, and flourished due to its ability to reinvent, reconfigure, and reorient its vistas of theoretical inquiries and technological advances. As several surveys, reports, and position papers (Murray 2003; Samad and Annaswamy 2011, 2014; Lamnabhi-Lagarrigue et al. 2017) have clearly articulated, the need for control systems has been reinforced through societal grand challenges in health and medicine, energy and climate, sustainability and development, and productivity and economic growth, to name a few. New frontiers are continuing to emerge, in application-centric directions such as Internet of Things and Industry 4.0, theoretical directions such as machine learning and data science, and conceptual directions such as cyber-physical-human systems (CPHS). The focus of this entry is on the last item, CPHS.

Significant advances in automation, communications, and computing have formed the foundation of the field of cyber-physical systems (CPS). As societal challenges increase, there is an increasing need for human elements to form closer ties with CPS. The underlying interactions and integrations of CPS and human systems are varied, complex, and highly challenging and merit a systematic investigation, forming the underpinning of CPHS.

The common thread between automation and human systems has a long and rich history and can perhaps be traced to the notion of cybernetics, if not even earlier. As Wiener postulates in his magnum opus (Wiener 1948), it is the scientific study of how humans, animals, and machines control and communicate with each other. The fundamental tenets of automation – steer, navigate, and govern – pervade intelligence, resilience, and several other high levels of

functioning in an uncertain environment. These same tenets are attributes of a human system as well. One can perhaps view CPHS as a new incarnation of cybernetics, with more fleshed-out roles for humans to play in specific application areas such as ground transportation, aerial platforms, and robotics, to name a few.

Let's start with an apt example of a CPHS, an automobile with varying levels of autonomy. Per the SAE J3016 standards, levels 0 through 5 are defined (ASME 2018), based on increasing levels of automation, with the roles of the human and automation, driver support features, and automated driving features defined. While much of the focus in this taxonomy is on the machine side, the human modeling component or the requisite interface between human and automation is only now beginning to be investigated. As we move from Level 2 to 3 and higher, the CPHS aspects become increasingly important. How the combined human-automation system performance, robustness, and safety can be addressed so that the resulting system exceeds what would be achievable by either a human or an automation acting alone is an important research direction that should be addressed by our research community. A second example is also a transportation one, in air, where the decision-makers involve a skilled human (i.e., a pilot) and automation (i.e., an autopilot) and the focus is on their combined interaction in the context of emergencies. Here too, the goal is to design the underlying CPHS to realize a performance that would exceed what may be achievable by the human or automation alone. Several other examples can be drawn, from energy (e.g., participation of consumers allowing power consumption to be flexible and on-demand), robotics (e.g., assistance to elders and to differentially abled humans), and manufacturing (congruence between skilled operators and automation).

The examples above illustrate the changing and nuanced nature of interactions between humans and CPS. Humans are no longer passive consumers or actors in these examples; they are empowered decision-makers and drive the evolution of the technology. While notions such as humans-in-the-loop have been around for a few decades, they are transforming with technological

advances. With a parallel development occurring in data analytics and AI research in human decision-making, human-technology coevolution is likely to lead to a very high impact research. Whether in-the-loop, on-the-loop, for-the-loop, or any combinations thereof, these combined interactions between humans and CPS need to be collectively studied, under the rubric of CPHS. The introduction of humans as a key component in the underlying system necessitates new tools, which are grounded in social sciences. Notions of human resilience have been studied in the area of cognitive systems, and concepts such as capacity for maneuver (CfM) and graceful command degradation (GCD) (Woods 2015) appear to have important roles to play in the design of CPHS. Human decision-making under uncertainty has been addressed very elegantly using the notion of prospect theory in the Nobel Prize-winning work of Kahneman and Tversky, which too is beginning to be increasingly employed in the energy and transportation systems modeling and control. Intrinsic to several scenarios in CPHS are interactions between multiple decision-makers that need to be strategic, necessitating the deployment of tools from game theory. In addition to strategic decisions, long-term or time-extended decisions may be called for and may have to be made in the presence of uncertainties. This implies that some form of learning may be useful, which leads to yet another intensely popular tool used of late – reinforcement learning. In this entry, we first elaborate on how prospect theory, game theory, reinforcement learning, and capacity for maneuver concept are used to model humans. Then, we provide specific examples where these tools are actively employed to model the underlying CPHS.

The Human Element in CPHS

The first challenge in CPHS is modeling. Apart from the intricacies of cyber-physical system (CPS) modeling, the human element is even more difficult to model in terms of forecasting their behavior. In this section, we introduce three different tools that are used to predict human actions within CPHS setting.

Prospect Theory

Many of the roles that humans play in CPS may be due to discrete actions rather than continuous ones. Typical examples concern a decision-making that consists of choosing one option out of many. In many infrastructures such as in transportation and energy, such decision-making is quite common and essential to ensure resource allocation and balance of supply and demand. The underlying theory that is often used to model such decision-making is the expected utility theory (EUT) (Ben-Akiva et al. 1985). Briefly, EUT can be explained as follows: Consider a problem where consumers have several travel modes. Utility theory postulates that they choose a travel mode based on optimization of the corresponding utility U of that travel mode. Suppose in a specific travel mode, U takes on discrete values $u_i \in \mathbb{R}, \forall i \in \{1, \dots, n\}$ where n is the number of possible outcomes, with u_i and p_i denoting the utility and probability of outcome i , respectively, with $\sum_{i=1}^n p_i = 1$. Then, one can determine the overall utility function as $U^o = E[U]$, where $E[\cdot]$ denotes the expectation operator, and compute U^o as (Von Neumann and Morgenstern 2007)

$$U^o = \sum_{i=1}^n p_i u_i \quad (1)$$

An interesting method that has been garnering increased attention of late for decision-making in the presence of multiple choices is cumulative prospect theory (CPT) (Tversky and Kahneman 1992; Kahneman and Tversky 2013). This method is appropriate when the underlying problem involves subjective human decision-making in the presence of uncertainty and risk. The behavioral model is described by a value function $V(\cdot)$ and a probability distortion function $\pi(\cdot)$, defined as follows (Tversky and Kahneman 1992; Prelec 1998):

$$V(u) = \begin{cases} (u - R)^{\beta^+} & \text{if } u \geq R \\ -\lambda(R - u)^{\beta^-} & \text{if } u < R \end{cases} \quad (2)$$

$$\pi(p) = e^{-(-\ln(p))^\alpha} \quad (3)$$

In the above, $u(x)$ denotes the utility function for an outcome x , and the nonlinearity $V(\cdot)$

introduces an asymmetry depending on whether or not the utility exceeds a certain reference R , which can be equated with a gain (if $u \geq R$) and a loss (if $u < R$). The parameter λ captures loss aversion and is typically greater than unity, $0 < \beta^+ < \beta^- < 1$ denotes diminishing sensitivity in gains compared to losses, and the parameter α indicates a distorted perception of probability, with low probabilities amplified and high probabilities attenuated. With these two non-linearities, the subjective utility of the outcome x , as perceived by the consumers, is given by

$$U_{PT} = \pi(p)V(u) \quad (4)$$

according to prospect theory, rather than the objective utility pu .

When multiple outcomes are present, a cumulative approach has to be taken in order to evaluate the distorted probability. Suppose that k out of the n outcomes are losses and the rest are gains. That is, $u_i < R$ if $1 \leq i \leq k$ and $u_i \geq R$ if $k < i \leq n$, and $F(U)$ is the cumulative distribution function of U . Then the perceived probability of outcome u_i with the probability p_i is given by (Von Neumann and Morgenstern 2007)

$$w_i = \begin{cases} \pi[F(u_i)] - \pi[F(u_{i-1})], & \text{if } i \in [1, k] \\ \pi[1 - F(u_{i-1})] - \pi[1 - F(u_i)], & \text{otherwise} \end{cases} \quad (5)$$

The subjective utility U^s perceived by the passenger within the CPT framework is given by

$$U^s = \sum_{i=1}^n w_i V(u_i). \quad (6)$$

Capacity for Maneuver Concept

A recent method in modeling of humans under anomalous events utilizes the capacity for maneuver (CfM) concept (Farjadian et al. 2016, 2017, 2020). CfM refers to the remaining range of the actuators before saturation, which quantifies the available maneuvering capacity of the vehicle. One example of the CfM is of the form $CfM = u_{\max} - \text{rms}(u(t))$, where $u_{\max}$ and $u(t)$ refer to the maximum available actuator deflection and actuator deflection at time t , respectively (Eraslan et al. 2020). It is hypothesized that surveillance and regulation of a system's available capacity to respond to all events help maintain the resiliency of a system, which is a necessary merit to recover from unexpected and abrupt failures or disturbances (Woods 2018). Below, two different types of human pilot models, both of which use CfM, are explained.

Perception Trigger

Here, the pilot model is assumed to assess the CfM and implicitly compute a perception gain,

K_t , based on the CfM. The perception algorithm for the pilot is

$$K_t = \begin{cases} 0, & |F_0| < 1 \\ 1, & |F_0| \geq 1 \end{cases} \quad (7)$$

where

$$F_0 = G_1(s)[F(t)], \quad (8)$$

and

$$F(t) = \frac{\frac{d}{dt}(\text{CfM}) - \mu_p}{3\sigma_p}. \quad (9)$$

$G_1(s)$ is a filter introduced as a smoothing and lagging operator into human perception algorithm and $F(t)$ is the perception variable. μ_p and σ_p are the average and the standard deviation of $\frac{d}{dt}(\text{CfM})$. The hypothesis is that the human pilot has such a perception trigger, K_t , and when $K_t = 1$ the pilot takes over control from the autopilot (Farjadian et al. 2016).

CfM-GCD Trade-off

Here, the pilot is assumed to implicitly assess the available CfM when an anomaly occurs and decide on the amount of graceful command degradation (GCD) that is allowable so as to let the CfM become comparable to a certain desired value. Degradation of the command, or the control input, can be implemented by setting a certain autopilot parameter by the pilot. In other

words, it is assumed that the pilot is capable of assessing the suitable value of a certain autopilot control parameter and inputs this value to the autopilot following the occurrence of an anomaly (Farjadian et al. 2020).

Game Theory and Reinforcement Learning

The difficulty of predicting human actions intensifies when the system contains more than one human, which requires factoring in multiple human-human and human-CPS interactions. For example, human interactions are generally not deterministic in nature, which can be projected into CPHS models by utilizing a probabilistic modeling framework. Furthermore, before taking an action, a human generally contemplates other intelligent agents' possible actions and then tries to choose a move that will increase the chances of obtaining the best outcome, which is called "strategic behavior" (Camerer 2011). Finally, humans do not always act in an optimal fashion. This final point is important for distinguishing CPHS models from autonomy models, where, based on available information, an algorithm can possibly be designed to react in the most appropriate way.

Game Theory

Game theory studies the interactions between strategic agents. "Players" in a game theoretical setting refer to the entities who can affect the game by their "moves." "Strategy" of a player defines the procedure based on which a player chooses his or her actions. A "solution concept" is a well-established set of rules that are used to predict how a game will unfold. A "Nash Equilibrium" is a solution concept, defined similarly to the equilibrium in system dynamics: When players have no incentive to deviate from their selected actions, the game is said to be in Nash Equilibrium. This means that in Nash Equilibrium, players choose their best actions against each other's actions. There are other equilibrium concepts such as "quantal response equilibrium" where instead of giving the best response to other players' actions, the players choose a probability distribution over their action space where actions

with higher expected payoffs have higher probability of being played.

Not all solution concepts predict an equilibrium. For example, "level-k thinking" is a non-equilibrium model, which assigns different levels of "reasoning" for players (Stahl and Wilson 1995; Costa-Gomes et al. 2009). In this model, the lowest level of reasoning is level-0, which represents non-strategic thinking, simply meaning that the players who reason at this level have a strategy that does not take into account other players' possible actions. A level-1 player, on the other hand, takes the best action assuming that his or her opponents are level-0 players. Similarly, a level-k player responds best to his *belief* that the other players are reasoning at level-(k-1). Therefore, this model assumes an *iterated* best response (Crawford 2008).

Reinforcement Learning

In RL, there exists an "agent" capable of exerting "actions" that can change the "state" of the environment where the agent operates. The RL problem can be defined as finding the optimal set of action sequence for an agent to achieve a given goal defined as a function of the environment states, through interaction with the said environment (Sutton and Barto 2018).

RL uses the idea of a "reward function" to describe the preferences of the agent while its learning to achieve a predetermined goal. A "policy" is defined as a map from states to actions. The task of the RL algorithm is to find a policy that will make the agent maximize a cumulative discounted reward expressed as

$$C = \sum_{t=0}^{\infty} \gamma^t r_t, \quad (10)$$

where γ is the discount factor and r is the reward obtained in every step t . There are various RL techniques proposed to discover action sequences that will attain the goal of maximizing (10). Almost all of these different methods are based on estimating the "value function," which is the value of being in a certain state, s , based on the policy, π . A value function is given as

$$V^\pi(s) = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s \right\}, \quad (11)$$

where E refers to expected value. The “action value function” is defined as the value of taking a certain action a , in a given state s , and is defined as

$$Q^\pi(s, a) = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a \right\}. \quad (12)$$

The aim of RL is to find the optimum policy π^* that will maximize the value function. The optimal value function is written as V^{π^*} , and all its state values are larger than or equal to that of all other value functions that are created by policies different than the optimal policy. This can be represented as $V^{\pi^*}(s) \geq V^\pi(s), \forall \pi, \forall s$. Similarly, we can write $Q^{\pi^*}(s, a) \geq Q^\pi(s, a), \forall \pi, \forall (s, a)$, for the optimal action value function. Once the optimal action value function is found, the following method can be used to select the best action in each state.

$$\pi^*(s) = \arg \max_a Q^{\pi^*}(s, a). \quad (13)$$

The process of finding the optimal policy is called “training.” During training, the agent observes the states and produces an action based on the observed states. This action influences the environment and results in a new set of states. These new states are evaluated by the reward function, and a reward signal is formed. The agent uses this signal to update the action value function, and the next cycle starts with the new action.

One of the RL methods that played a significant role for the success of RL is “Q-learning” (Watkins 1989). In Q-learning, an incremental estimate of the optimal action value function is realized using the following update rule.

$$\begin{aligned} Q_{k+1}(s_t, a_t) = & Q_k(s_t, a_t) \\ & + \alpha \left(r_t + \gamma \max_a Q_k(s_{t+1}, a) \right. \\ & \left. - Q_k(s_t, a_t) \right). \end{aligned} \quad (14)$$

Merging Game Theory and Reinforcement Learning to Model Human Interactions

For human interactions that last long periods of time, using equilibrium-based game theoretical solutions may be computationally infeasible. One way around this problem is using the non-equilibrium game theoretical solution concept, level- k thinking, explained in section “[Game Theory](#).” This means that once the “anchoring” level, level-0, is selected, a level-1 human agent’s behavior can be identified as the best response to all the other human actions that are determined by the level-0 policy. Similarly, once level-1 behavior is identified, all the agents in the system are assigned the level-1 policy except the one whose level-2 behavior is to be found. Therefore, to predict the policy of a level- k agent, all the rest of the agents’ policies are set to level- $(k-1)$, which effectively make them a part of the environment whose dynamics are known, and level- k policy is determined as the best response to the rest of the level- $(k-1)$ policy actions. This isolates the level- k policy as the single policy that needs to be computed. Once the difficulty of high computational cost due to multiple decision-makers is solved using level- k thinking, the problem reduces to estimating the optimal action sequence of an intelligent agent in a given environment. This problem can be stated as a reinforcement learning (RL) problem by properly defining the states, actions, the environment, and the reward function. The algorithm used in obtaining the agent policies, demonstrating the interplay between the game theory and RL, is provided in Algorithm 1 (Albaba and Yildiz 2019), where k is the maximum desired level.

Interconnecting Humans and Cyber-Physical Systems: Examples of CPHS

Prospect Theory Examples

We describe two examples below where prospect theory has been explored, the first of which is in power grids and the second is in ride-sharing services.

Algorithm 1 Merging RL and game theory

```

1: Set  $i = 0$ 
2: while  $i < k$  do
3:   Load the level- $i$  policy
4:   Set cognition levels of all players in the environ-
     ment other than the learning agent to level- $i$ , i.e.,
     set policies of players to level- $i$  policy
5:   Place the learning agent in the initialized environ-
     ment, in which all players are level- $i$ 
6:   Start the training of the learning agent using a rein-
     forcement learning method, through which agent
     learns how to best respond to level- $i$  players
7:   Once the training is completed, learning agent
     becomes a level- $(i+1)$  player
8:   Save the policy of the learning agent as level- $(i+1)$ 
     policy
9:    $i += 1$ 
10: end while

```

- (1) Suppose there are N consumption units capable of adjusting their loads located at a node in a power grid. This capability may be either due to the fact that the load itself is adjustable (e.g., temperature set-point in a HVAC system) or they include a storage unit that has a flexible charging/discharging schedule. The corresponding utility function of a unit k consists of $U(D_k, S_k)$, where D_k is the quantity of energy that is charged and S_k is the amount of energy discharged. In general U_k depends on the amount of energy imbalance, the price of electricity, the actions of other assets at that node and their flexibility, and the penalty that the power company may impose for unmet commitment, among others. The argument for using a PT-based model rather than EUT is because of the uncertainty introduced by other players in the underlying trading game. The effect of the overall risk-aversion may be modeled through the use of the (V, π) tuple described in (2) and (3) using (4) (Wang et al. 2014).
- (2) The second example has been explored in the context of ride-sharing services. Here, the utility function depends on both travel times and the price of the service and is of the form

$$U = X + b\gamma \quad (15)$$

where γ is the tariff from ride offer and is deterministic and b represents the weighting of the economic cost of the ride in comparison to the travel cost. The time-based utility is of the form

$$X = a_1 T_{\text{walk}} + a_2 T_{\text{wait}} + a_3 T_{\text{ride}} + c \quad (16)$$

and is a stochastic quantity, as each of the travel times can vary randomly. As the amount of stochasticity is significant, a prospect theory-based approach may be more appropriate. It should also be noted that the utility function is a continuous function, with the corresponding function $U(t)$ varying over an interval $[t_{\min}, t_{\max}]$ which is determined by the bounds of the overall travel. Each $U(t)$ has an associated cumulative distribution function $F(U)$. Since there is a significant subjective component in this problem, the passengers who are willing to accept the ride service may perceive the utility U as $V(U)$ and perceive the corresponding probability of $U(t)$ in a distorted manner as well. As the underlying utility function is continuous, one needs to adopt a CPT approach here (Tversky and Kahneman 1992).

We first present the CPT model for a discrete case, with utility U_i for outcome $i, i = 1, \dots, n$. Each U_i represents the utility of the ride-sharing service for a time cost t_i . For example, the wait time could be 10 min, the travel time 30 min, and the walk time 5 min, leading to a total time of 45 min, which has a certain utility to the passenger. The objective utility therefore can be calculated using (1). Using the CPT discussions above, the subjective utility can be calculated using (6). The extension from (6) to the continuous case of U_R^s then becomes

$$\begin{aligned}
U^s = & \int_{-\infty}^R V(u) \frac{d}{du} \left\{ \pi[F_U(u)] \right\} du \\
& + \int_R^{\infty} V(u) \frac{d}{du} \left\{ -\pi[1 - F_U(u)] \right\} du
\end{aligned} \quad (17)$$

One can similarly calculate the distorted probabilities. Suppose we consider a simple case each passenger is assumed to choose between two options, the ride-sharing service and another alternative such as public transportation, which may have a utility A^o . The objective probability of acceptance is given by

$$p^o = \frac{e^{U^o}}{e^{U^o} + e^{A^o}} \quad (18)$$

The subjective probability of acceptance is given by

$$p^s = \frac{e^{U^s}}{e^{U^s} + e^{A^s}} \quad (19)$$

where A^s denotes the subjective utility of the alternative perceived by the passenger. The overall prospect theory-based passenger model is then quantified using the pair (U^s, p^s) . These in turn could be used to determine the dynamic tariff of the ride-sharing service. Such an approach directly accommodates the behavioral model of humans. The reader is referred to Guan et al. (2019) for further details.

Shared Control Architectures

Shared control architectures (SCAs) exemplified in this section describe the responsibility sharing between pilots and autopilots in terms of controlling the aircraft. In section “Capacity for Maneuver Concept” two different models of human decision-making were discussed, both based on the monitoring of the CfM. In this section, we introduce two different SCAs based on these models (Eraslan et al. 2020).

SCA 1: A Pilot with a CfM-Based Perception and a Fixed-Gain Autopilot

The first shared control architecture can be summarized as a sequence {autopilot runs, anomaly occurs, pilot takes over}. That is, it is assumed that an autopilot based on PD control is in place, ensuring a satisfactory command tracking under nominal conditions. The human pilot is assumed to consist of a perception component and an adaptation component. The perception component consists of monitoring CfM, through which

a perception trigger F_0 is calculated using (8). The adaptation component consists of monitoring the control gain in (7) and taking over control of the aircraft when $K_t = 1$. The details of this shared controller and its evaluation using simulations and experiments are given by Farjadian et al. (2016) and Eraslan et al. (2020). Figure 1 illustrates the schematic of SCA1.

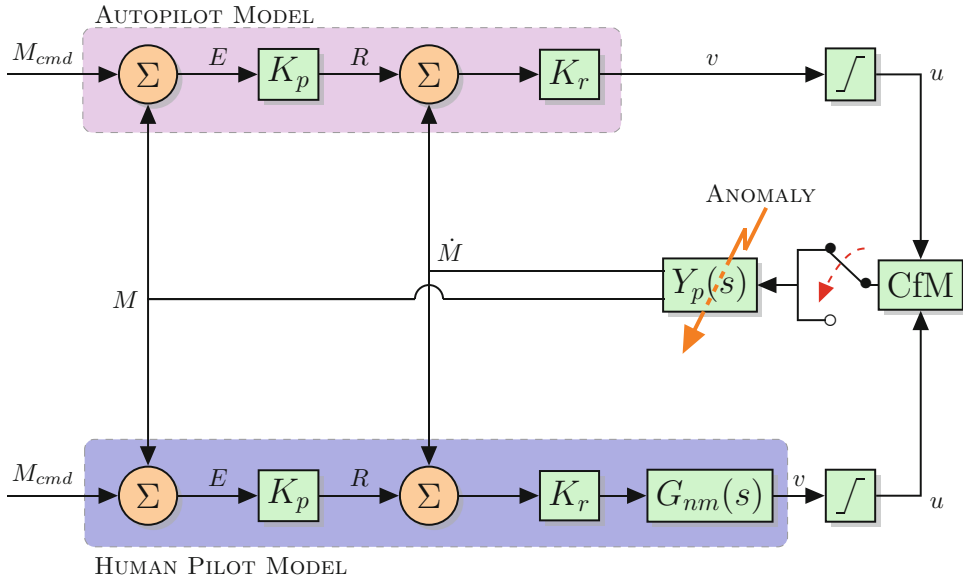
SCA 2: A Pilot with a CfM-Based Decision-Making and an Advanced Adaptive Autopilot

In this shared control architecture, the role of the pilot is a supervisory one, while the autopilot takes on an increased and complex role. The pilot is assumed to monitor the CfM of the actuators in the aircraft following an anomaly. In an effort to allow the CfM to stay close to a desired value, the command is allowed to be degraded; the pilot then determines a parameter μ which directly scales the control effort. Once μ is specified by the pilot, then the adaptive autopilot continues to supply the control input. If the pilot has high situational awareness, he/she provides an estimate of the severity of the anomaly. The details of this shared controller and its evaluation using simulations and experiments are given by Farjadian et al. (2017, 2020) and Eraslan et al. (2020). Figure 2 shows the schematic of SCA2.

Airspace in the Presence of Both Manned and Unmanned Aircraft

Obtaining airspace models where both manned and unmanned aircraft are present is a necessity for successful integration of unmanned aircraft systems (UAS) into the National Airspace System (NAS). Sense and avoid (SAA) algorithms, human-human and human-automation interactions through aircraft dynamics, aircraft, and traffic collision avoidance algorithms (TCAS), together with the communication links, make this system a typical example of a CPHS.

Figure 3 shows such an airspace scenario where a UAS (magenta) is assigned to follow certain waypoints (yellow) in a crowded airspace filled with manned aircraft (red). We are interested in how the overall system will evolve in time.



Cyber-Physical-Human Systems, Fig. 1 Block diagram of SCA1. (Adapted from Farjadian et al. 2016). The autopilot model consists of a fixed gain controller, whereas the human pilot model comprehends a perception part based on the CfM concept and an adaptation part governed by empirical adaptive laws (Hess 2016). The neuromuscular transfer function (Hess 2006), $G_{nm}(s)$,

corresponds to control input formed by an arm or leg. When an anomaly occurs, the plant dynamics, $Y_p(s)$, undergoes an abrupt change by rendering the autopilot insufficient for the rest of the control. At this stage, the occurrence of an anomaly is captured by the CfM such that the control is handed over to the human pilot model for a resilient flight control

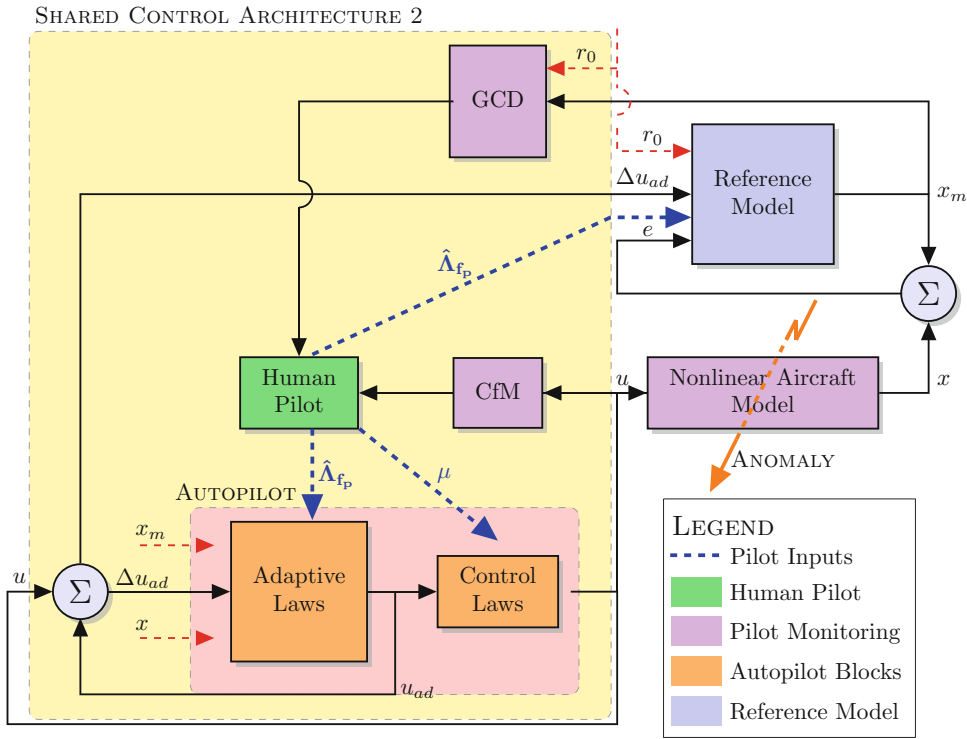
To model this scenario, we need to include aircraft dynamics, TCAS and SAA algorithms, and communication dynamics which may include signal delays. These components of the model are already well understood and documented in the literature. The human interaction dynamics, on the other hand, is the main challenge.

As discussed in section “[Game Theory and Reinforcement Learning](#),” the GT- and RL-based modeling approach produces policies, which are probabilistic maps from observations to actions, to represent human interactions. Therefore, to obtain these policies, we first need to clearly define the *observation and action spaces* and represent them in a way that is meaningful for the RL algorithm. Once these spaces are explicitly defined, pilot goals and preferences need to be expressed in form of a *reward function*. Obtaining these policies and integrating them with the automation and the physical manned and unmanned aircraft dynamics enables a simulation capability to investigate and study various UAS integration into the NAS scenarios. A detailed

description of how to achieve this together with several simulation studies and validations with a data-based model is given by Musavi et al. (2016).

CPHS Beyond the Engineering Domain

The role of CPHS is significantly larger. Much of the focus in the topics outlined above was on engineering systems and how we come up with better tools for their analysis and synthesis. The focus needs to be diverted towards social systems as well and how the emergence of CPHS and the varied roles of humans therein introduce new challenges therein, in domains related to ethics, economics, justice, and the like. While the topic of ethics has been studied for millennia by societies and civilizations, the creation of new socio-technical systems introduces new paradigms for societal interactions and thereby new challenges.



Cyber-Physical-Human Systems, Fig. 2 Block diagram of SCA 2. The human pilot undertakes a supervisory role by providing the key parameters μ and $\hat{\lambda}_{fp}$ to the

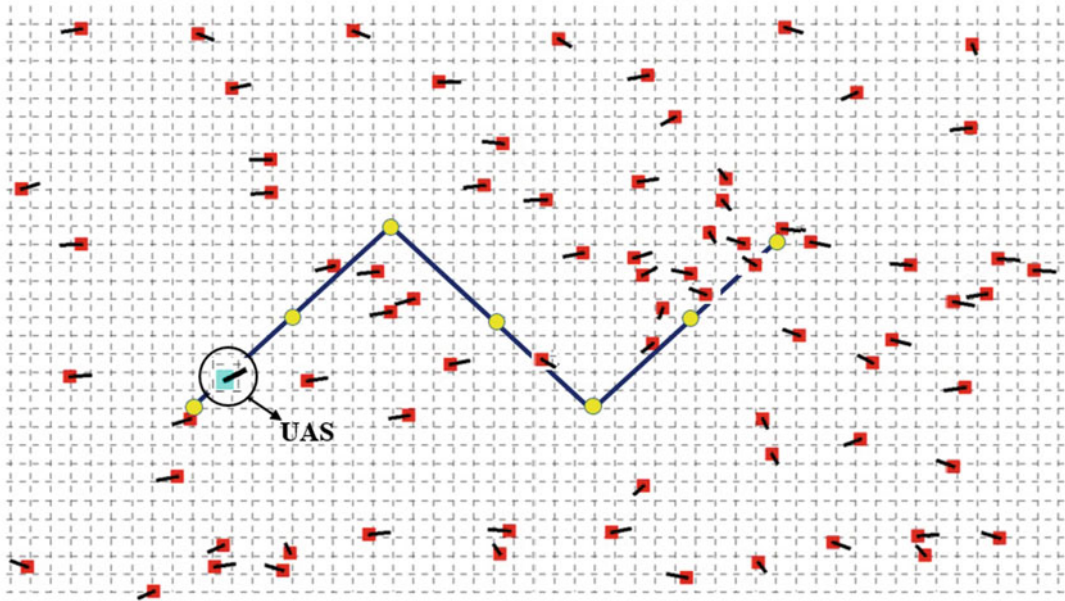
adaptive autopilot. The blocks are expressed in different colors based on their functions in the proposed SCA2

This entry would not be complete without bringing in yet another important connection between systems and control and social sciences, which can be grouped under the rubric of dynamics over techno-social networks, DTSN (see Section 5.20 on Social Networks in Lamnabhi-Lagarrigue et al. (2017)). DTSN has its origins in social network analysis which is grounded in social philosophy and psychology and focuses on the examination of social relations, influences, group actions, and interactions. By leveraging the intersection between SNA and graph theory, several computationally efficient algorithms have been derived for the analysis of large-scale systems. With rapid changes in technology, these large-scale systems are now beginning to have a strong temporal signature, which implies the need for a dynamical system point of view, leading to DTSN. The focus of this emerging area has a huge range of applicability, with its emphasis on the study of both topological and dynamical

properties of the network, properties of resilience and robustness in the presence of natural disasters and cyberattacks that may cause failures in nodes and links in the network. A notable and topical application area is the current COVID-19 spread over the world population.

Summary

As societal challenges increase, there is an increasing need for human systems to interact with cyber-physical systems in various ways to provide innovative solutions that overcome these challenges. The rubric that addresses the foundations of such emerging interactions and integrations between CPS and human systems forms CPHS. This entry provides an introduction to CPHS and representative examples from energy and transportation sectors where such CPHS have begun to emerge.



Cyber-Physical-Human Systems, Fig. 3 A hybrid airspace scenario where red squares represent manned aircraft and the magenta square is used to indicate an unmanned aircraft (UA). The yellow circles in the 600 ×

300 km airspace are the waypoints assigned to the UA. (Musavi et al. (2016), reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc)

One of the main challenges in CPHS is modeling of the human element. Given the enormously complex human system that functions over a huge range of timescales to perform a wide range of tasks, there are perhaps as many models of humans needed for as many problems. The models presented in this entry therefore are in no way meant to be comprehensive but rather illustrations. We have focused on a few instances where the underlying models attempt to capture the perceptive, strategic, and decision-making abilities of humans in various contexts. In addition, we have tried to illustrate varied concepts and methodologies which include prospect theory, game theory and reinforcement learning, and capacity for maneuver that may be appropriate for deriving human models in these contexts.

While the applications that we have outlined in this entry correspond to the energy and transportation sectors, by no means are they meant to be the only applications of interest in CPHS. Several examples abound in the area of robotics,

healthcare, and manufacturing, which have not been covered here (see the ongoing IFAC workshop series on CPHS for ongoing and emerging activities in this area!).

Cross-References

- ▶ [Adaptive Human Pilot Models for Aircraft Flight Control](#)
- ▶ [Human-Building Interaction \(HBI\)](#)
- ▶ [Interaction Patterns as Units of Analysis in Hierarchical Human-in-the-Loop Control](#)
- ▶ [Medical Cyber-Physical Systems: Challenges and Future Directions](#)
- ▶ [Physical Human-Robot Interaction](#)
- ▶ [Pilot-Vehicle System Modeling](#)
- ▶ [Trust Models for Human-Robot Interaction and Control](#)

Acknowledgments This effort was sponsored by Boeing Strategic University Initiative and the Turkish Academy of Sciences.

Bibliography

- Albaba BM, Yildiz Y (2019) Modeling cyber-physical human systems via an interplay between reinforcement learning and game theory. *Ann Rev Control* 48:1–21
- ASME (2018) SAE international releases updated visual chart for its “levels of driving automation” standard for self-driving vehicles. Available at <https://www.sae.org/news/press-room/2018/12/sae-international-releases-updated-visual-chart-for-its-“levels-of-driving-automation”-standard-for-self-driving-vehicles>. Accessed at 05 Apr 2020
- Ben-Akiva ME, Lerman SR, Lerman SR (1985) Discrete choice analysis: theory and application to travel demand, vol 9. MIT Press
- Camerer CF (2011) Behavioral game theory: experiments in strategic interaction. Princeton University Press
- Costa-Gomes MA, Crawford VP, Iriberri N (2009) Comparing models of strategic thinking in Van Huyck, Battalio, and Beil’s coordination games. *J Eur Econ Assoc* 7(2–3):365–376
- Crawford VP (2008) Modeling behavior in novel strategic situations via level-k thinking. In: North American Economic Science Association Meetings
- Eraslan E, Yildiz Y, Annaswamy AA (2020) Shared control between pilots and autopilots: illustration of a cyber-physical human system. *IEEE control systems magazine*, in press. Preprint available at <https://arxiv.org/abs/1909.07834> Accessed on 05 June 2020
- Farjadian A, Annaswamy A, Woods D (2016) A resilient shared control architecture for flight control. In: Proceedings of the international symposium on sustainable systems and technologies. <http://aacslab.mit.edu/publications.php>
- Farjadian AB, Annaswamy AM, Woods D (2017) Bumpless reengagement using shared control between human pilot and adaptive autopilot. *IFAC-PapersOnLine* 50(1):5343–5348
- Farjadian AB, Thomsen B, Annaswamy AM, Woods DD (2020) Resilient flight control: An architecture for human supervision of automation. *IEEE Trans Control Syst Technol* (in press)
- Guan Y, Annaswamy AM, Tseng HE (2019) Cumulative prospect theory based dynamic pricing for shared mobility on demand services. In: IEEE 58th annual conference on decision and control (CDC)
- Hess RA (2006) Simplified approach for modelling pilot pursuit control behaviour in multi-loop flight control tasks. *Proc Inst Mech Eng Part G J Aerosp Eng* 220(2):85–102
- Hess RA (2016) Modeling human pilot adaptation to flight control anomalies and changing task demands. *J Guid Control Dyn* 39(3):655–666
- Kahneman D, Tversky A (2013) Prospect theory: an analysis of decision under risk. In: *Handbook of the fundamentals of financial decision making: Part I*. World Scientific, pp 99–127
- Lamnabhi-Lagarrigue F, Annaswamy A, Engell S, Isaksson A, Khargonekar P, Murray RM, Nijmeijer H, Samad T, Tilbury D, den Hof PV (2017) Systems & control for the future of humanity, research agenda: current and future roles, impact and grand challenges. *Ann Rev Control* 43:1–64. <http://www.sciencedirect.com/science/article/pii/S1367578817300573>
- Murray RM (2003) Control in an information rich world: report of the panel on future directions in control, dynamics, and systems. SIAM
- Musavi N, Onural D, Gunes K, Yildiz Y (2016) Unmanned aircraft systems airspace integration: a game theoretical framework for concept evaluations. *J Guid Control Dyn* 40(1):96–109
- Prelec D (1998) The probability weighting function. *Econometrica* 66(3):497–527
- Samad T, Annaswamy A (2011, 2014) The impact of control technology: overview, success stories, and research challenges. IEEE Control Systems Society. Available at <http://ieeecss.org/sites/ieeecss/files/2019-07/IoCT-FullReport-v2.pdf> (1st edn) <http://ieeecss.org/general/IoCT2-report> (2nd edn)
- Stahl D, Wilson P (1995) On players’ models of other players: theory and experimental evidence. *Games Econ Behav* 10(1):218–254
- Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT Press
- Tversky A, Kahneman D (1992) Advances in prospect theory: cumulative representation of uncertainty. *J Risk Uncertain* 5(4):297–323
- Von Neumann J, Morgenstern O (2007) Theory of games and economic behavior (commemorative edition). Princeton University Press
- Wang Y, Saad W, Mandayam NB, Poor HV (2014) Integrating energy storage into the smart grid: a prospect theoretic approach. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 7779–7783
- Watkins CJCH (1989) Learning from delayed rewards. Ph.D. thesis, King’s College, Cambridge
- Wiener N (1948) Cybernetics or control and communication in the animal and the machine. Technology Press
- Woods DD (2015) Four concepts for resilience and the implications for the future of resilience engineering. *Reliab Eng Syst Saf* 141:5–9
- Woods DD (2018) The theory of graceful extensibility: basic rules that govern adaptive systems. *Environ Syst Decis* 38(4):433–457

Data Association

Yaakov Bar-Shalom and Richard W. Osborne
University of Connecticut, Storrs, CT, USA

Abstract

In tracking applications, following the signal detection process that yields measurements, there is a procedure that *selects* the measurement(s) to be incorporated into the state estimator – this is called **data association (DA)**. In multitarget-multisensor tracking systems, there are generally three classes of data association: specifically, **measurement-to-track association (M2TA)**, **track-to-track association (T2TA)**, and **measurement-to-measurement association (M2MA)**. M2TA is the process of associating each measurement from a list (originating from one or more sensors) to a new or existing track. T2TA is the process of associating multiple existing tracks (from multiple sensors or from different periods in time), generally with the intent of fusing them afterward. M2MA is the process of associating measurements from different sensors in order to form “composite measurements” and/or do track initialization. The processes of M2TA and T2TA will be discussed in more detail here, while details on M2MA can be found in Bar-Shalom et al. (2011).

Keywords

Clutter · Measurement origin uncertainty · Measurement validation · Persistent interference · Tracking

Introduction

In a radar the “return” from the target of interest is sought within a time interval determined by the anticipated range of the target when it reflects the energy transmitted by the radar: a “range gate” is set up and the detection(s) within this gate can be *associated* with the target of interest.

In general the measurements have a higher dimension:

- Range, azimuth (bearing), elevation, or direction sines for radar, possibly also range rate
- Bearing and frequency (when the signal is narrow band) or time difference of arrival and frequency difference in passive sonar
- Two line-of-sight angles or direction sines for optical or passive electromagnetic sensors

Then a *multidimensional gate* is set up for detecting the signal from the target. This is done to avoid searching for the signal from the target of interest in the entire measurement space. A measurement in the gate, while not guaranteed to have originated from the target the gate pertains to, is a *valid association candidate* – thus, the name **validation region** or **association region**. If there

is more than one detection (measurement) in the gate, this leads to an **association uncertainty**.

In the discussion to follow, it will be assumed that one has **point measurements** rather than distributed over several **resolution cells** of the sensor as in the case of an **extended target**.

Similar validation has to be carried out in T2TA.

Validation Region

In view of the variety of variables that can be measured, a generic gating (or validation or association) procedure for continuous-valued measurements is discussed.

Consider a target that is in track, i.e., its filter has been initialized. Then, according to Sect. 5.2.3 of Bar-Shalom et al. (2001), one has the predicted value (mean) of the measurement $\hat{z}(k+1|k)$ and the associated covariance $S(k+1)$.

Assumption. The true measurement conditioned on the past is **normally (Gaussian) distributed** (The notation $\mathcal{N}(x; \mu, S)$ stands for the normal (Gaussian) pdf with the argument (vector) random variable x , mean μ , and covariance matrix S . The reason for the use of the designation “normal” is to distinguish this omnipresent pdf from all the others (abnormal).) with its **probability density function (pdf)** given by

$$p[z(k+1)|Z^k] = \mathcal{N}[z(k+1); \hat{z}(k+1|k), S(k+1)] \quad (1)$$

where $S(k+1)$ is the **innovation (residual) covariance** matrix and z is the true measurement.

Then the true measurement will be in the following region:

$$\mathcal{V}(k+1, \gamma) = \{z : d^2 \leq \gamma\} \quad (2)$$

with probability determined by the **gate threshold** γ and

$$d^2 \triangleq [z - \hat{z}(k+1|k)]' S(k+1)^{-1} [z - \hat{z}(k+1|k)] \quad (3)$$

This distance metric, d^2 , is referred to in the literature as the **normalized innovation squared (NIS)**, **statistical distance squared**, **Mahalanobis distance**, or **chi-square distance**.

The region defined by (2) is called the **gate** or **validation region** (hence, the notation $\mathcal{V}$) or **association region**. It is also known as the **ellipsoid (or ellipsoid) of probability concentration** – the region of *minimum volume* that contains a given probability mass under the Gaussian assumption. The semiaxes of the ellipsoid (2) are the square roots of the eigenvalues of γS . The threshold γ is obtained from tables of the chi-square distribution since the quadratic form (3) that defines the validation region in (2) is chi-square distributed with number of degrees of freedom equal to the dimension n_z of the measurement.

Table 1 gives the **gate probability** (The notation $P\{\cdot\}$ is used to denote the probability of event $\{\cdot\}$.)

$$P_G \triangleq P\{z(k+1) \in \mathcal{V}(k+1, \gamma)\} \quad (4)$$

or the “probability that the (true) measurement will fall in the gate” for various values γ and dimensions n_z of the measurement. The square root $g = \sqrt{\gamma}$ is sometimes referred to as the “number of sigmas” (standard deviations) of the gate. This, however, does not fully define the probability mass in the gate as can be seen from Table 1.

Remark 1 It should be pointed out that thresholding in a detector is also a form of gating – only a signal above a certain intensity level (at the end of the signal processing chain) is accepted as a detection and then one has a measurement. In this case the “gate” is the interval $[\tau, \infty]$ in the signal intensity space, where τ is the detection threshold.

A Single Target in Clutter

The validation procedure limits the region in the measurement space where the information processor will “look” to find the measurement from the target of interest. In spite of this, it can happen that *more than one detection*, i.e., several

Data Association, Table 1 Gate thresholds and the probability mass P_G in the gate

γ	1	4	6.6	9	9.2	11.4	16	25
g	1	2	2.57	3	3.03	3.38	4	5
n_z								
1	0.683	0.954	0.99	0.997			0.99994	1
2	0.393	0.865		0.989	0.99		0.9997	1
3	0.199	0.739		0.971		0.99	0.9989	0.99998

measurements, will be found in the validation region.

Measurements *outside the validation region* can be ignored: they are “too far” and thus very unlikely to have originated from the target of interest. This holds if the gate probability is close to unity and *the model used to obtain the gate is correct*.

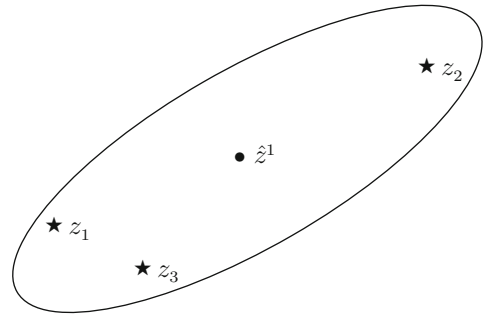
The problem of tracking a single target in clutter considers the situation where there are possibly several measurements in the validation region (gate) of a target. The set of **validated measurements** consists of:

- The correct measurement (if detected and it fell in the gate)
- The undesirable measurements: clutter or false alarm originated

In practice detections are obtained by thresholding the signal received by the sensor after processing it. This is the simplest (binary) way of using a target **feature** – its intensity. More sophisticated ways of using such feature information can be found in Bar-Shalom et al. (2011).

It is assumed that the measurement contains all the information that could be used to discard the undesirable measurements. Therefore, any measurement that has been validated could have originated from the target of interest.

A situation with a single-target track and several validated measurements is depicted in Fig. 1. The (two-dimensional) validation region is an ellipse centered at the predicted measurement $\hat{z}^1$. The parameters of the ellipse are determined by the covariance matrix S of the innovation, which is assumed to be Gaussian.

**Data Association, Fig. 1** Several measurements in the validation region of a single track

All the measurements in the validation region can be said to be *not too unlikely* to have originated from the target of interest, even though only one is assumed to be the true one.

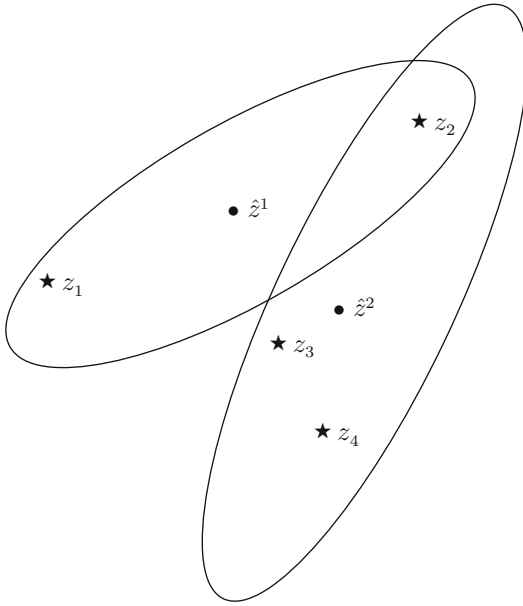
The implication of the assumption that there is a single target is that the *undesirable measurements constitute a random interference*. The common mathematical model for such **false measurements** is that they are:

- Uniformly spatially distributed
- Independent across time

This corresponds to what is known as **residual clutter** – the constant clutter, if any, is assumed to have been removed.

Multiple Targets in Clutter

The situation where there are several target tracks in the same neighborhood as well as clutter (or false alarms) is more complicated. Figure 2 illustrates such a case for a given time, with the predicted measurements for the two targets considered denoted as $\hat{z}^1$ and $\hat{z}^2$. In this figure the following measurement origins are possible:



Data Association, Fig. 2 Two tracks with a measurement in the intersection of their validation regions

- z_1 from target 1 or clutter
- z_2 from either target 1 or target 2 or clutter
- z_3 and z_4 from target 2 or clutter

However, if z_2 originated from target 2, then it is quite likely that z_1 originated from target 1. This illustrates the *interdependence of the associations* in a situation where a **persistent interference** (neighboring target) is present in addition to random interference (clutter).

Up to this point, it was assumed that a measurement could have originated from *one of the targets* or from *clutter*. However, in view of the fact that any signal processing system has an inherent **finite resolution** capability, an additional possibility has to be considered:

z_2 could be the result of the **merging** of the detections from the two targets – it is an **unresolved measurement**.

This constitutes a fourth origin hypothesis for a measurement that lies in the intersection of two validation regions. Most tracking algorithms ignore the possibility that a measurement is an unresolved one.

This illustrates only the difficulty of association of measurements to tracks *at one point in time*. The full problem, as will be discussed later, consists of associating measurements *across time*.

Approaches to Tracking and Data Association

The problem of **tracking and data association** is a **hybrid** problem because it is characterized by:

- (1) Continuous uncertainties – state estimation in the presence of continuous noises
- (2) Discrete uncertainties – which measurement(s) should be used in the estimation process

Assuming the goal is to obtain the MMSE estimate of the target state – its conditional mean – one can distinguish the following approaches.

Pure MMSE Approach

The **Pure MMSE Approach** to tracking and data association is obtained using the smoothing property of expectations (see, e.g., Bar-Shalom et al. 2001, Sect. 1.4.12), as follows:

$$\begin{aligned}\hat{x}^{\text{MMSE}} &= E[x|Z] = E\{E[x|A, Z]|Z\} \\ &= \sum_{A_i \in \mathcal{A}} E[x|A_i, Z] P\{A_i|Z\}\end{aligned}\quad (5)$$

where A is an association event (assuming a Bayesian model, with prior probabilities from which one can calculate posterior probabilities), and the summation is over all events A_i in the set $\mathcal{A}$ of mutually exclusive and exhaustive association events.

The above, which requires the evaluation of all the conditional (posterior) probabilities $P\{A_i|Z\}$, is a direct consequence of the **total probability theorem** (see, e.g., Bar-Shalom et al. 2001, Sect. 1.4.10), which yields the conditional pdf of the state as the following mixture

$$p(x|Z) = \sum_{A_i \in \mathcal{A}} p(x|A_i, Z) P\{A_i|Z\} \quad (6)$$

In the linear-Gaussian case the above becomes a **Gaussian mixture**. Algorithms that fall in this category are PDAF and JPDAF, see Bar-Shalom et al. (2011).

MMSE-MAP Approach

The **MMSE-MAP Approach**, instead of enumerating and summing over all the association events, selects the one with highest posterior probability, namely,

$$A^{\text{MAP}} = \arg \max_i P\{A_i|Z\} \quad (7)$$

and then

$$\hat{x}^{\text{MMSE-MAP}} = E[x|A^{\text{MAP}}, Z] \quad (8)$$

The HOMHT as proposed by Reid (1979) falls in this category, see Bar-Shalom et al. (2011).

MMSE-ML Approach

The **MMSE-ML Approach** does not assume priors for the association events and relies on the maximum likelihood approach to select the event, that is,

$$A^{\text{ML}} = \arg \max_i p\{Z|A_i\} \quad (9)$$

and then

$$\hat{x}^{\text{MMSE-ML}} = E[x|A^{\text{ML}}, Z] \quad (10)$$

The TOMHT falls into this category and S-D assignment (or MDA) is an implementation of this, see Bar-Shalom et al. (2011).

Heuristic Approaches

There are numerous simpler/heuristic approaches. The most common one relies on the distance metric (3) and makes the selection of which measurement is associated with which track based on the “nearest neighbor” rule. The same criterion can be used in a global cost function.

Remarks

It should be noted that the MMSE-MAP estimate (8) and the MMSE-ML estimate (10) are obtained assuming that the selected association is *correct* – a **hard decision**. This hard decision is sometimes correct, sometimes wrong. On the other hand, the pure MMSE estimate (5) yields a **soft decision** – it averages over all the possibilities. This soft decision is never totally correct, never totally wrong.

The uncertainties (covariances) associated with the MMSE-MAP and MMSE-ML estimates might be optimistic in view of the above observation. The uncertainty associated with the pure MMSE estimate will be *increased* (realistically) in view of the fact that it includes the data association uncertainty.

Estimation and Data Association in Nonlinear Stochastic Systems

The Model

Consider the discrete time stochastic system

$$\mathbf{x}(k+1) = f[k, \mathbf{x}(k), \mathbf{u}(k), \mathbf{v}(k)] \quad (11)$$

where $\mathbf{x} \in \mathcal{R}^n$ is the stacked state vector of the targets under consideration, $\mathbf{u}(k)$ is a known input (included here for the sake of generality), and $\mathbf{v}(k)$ is the process noise with a known pdf. The measurements at time $k+1$ are described by the stacked vector

$$\mathbf{z}(k+1) = h[k+1, \mathbf{x}(k+1), A(k+1), \mathbf{w}(k+1)] \quad (12)$$

where $A(k+1)$ is the data association event at $k+1$ that specifies (i) which measurement component originated from which components of $\mathbf{x}(k+1)$, namely, from which target, and (ii) which measurements are false, that is, originated from the clutter process. The vector $\mathbf{w}(k)$ is the observation noise, consisting of the error in the true measurement and the false measurements. The pdf of the false measurements and the probability mass function (pmf) of their number are also assumed to be known.

The noise sequences and false measurements are assumed to be white with known pdf and mutually independent. The initial state is assumed to have a known pdf and to be independent of the noises. Additional assumptions are given below for the optimal estimator, which evaluates the pdf of the state conditioned on the observations.

The optimal state estimator in the presence of data association uncertainty consists of the computation of the conditional pdf of the state $\mathbf{x}(k)$ given all the information available at time k , namely, the prior information about the initial state, the intervening inputs, and the sets of measurements through time k . The conditions under which the optimal state estimator consists of the computation of this pdf are presented in detail.

The Optimal Estimator for the Pure MMSE Approach

The **information set** available at k is

$$I^k = \{Z^k, U^{k-1}\} \quad (13)$$

where

$$Z^k \triangleq \{\mathbf{z}(j)\}_{j=1}^k \quad (14)$$

is the cumulative set of observations through time k , which subsumes the initial information Z^0 , and U^{k-1} is the set of known inputs prior to time k .

For a stochastic system, an **information state** (Striebel 1965) is a function of the available information set that summarizes the past of the system in a probabilistic sense.

It can be shown that the conditional pdf of the state

$$p_k \triangleq p[\mathbf{x}(k)|I^k] \quad (15)$$

is an information state if (i) the two noise sequences (process and measurement) are white and mutually independent and (ii) the target detection and clutter/false measurement processes are white. Once the conditional pdf (15) is available, the **pure MMSE estimator**, i.e., the conditional mean, as well as the conditional variance, or covariance matrix, can be obtained.

The optimal estimator, which consists of the recursive functional relationship between the

information states p_{k+1} and p_k , is given by

$$p_{k+1} = \psi[k+1, p_k, \mathbf{z}(k+1), \mathbf{u}(k)] \quad (16)$$

where

$$\begin{aligned} & \psi[k+1, p_k, \mathbf{z}(k+1), \mathbf{u}(k)] \\ &= \frac{1}{c} \sum_{i=1}^{M(k+1)} p[\mathbf{z}(k+1)|\mathbf{x}(k+1), A_i(k+1)] \\ & \cdot \int p[\mathbf{x}(k+1)|\mathbf{x}(k), \mathbf{u}(k)] p_k d\mathbf{x}(k) \\ & P\{A_i(k+1)\} \end{aligned} \quad (17)$$

is the transformation that maps p_k into p_{k+1} ; the integration in (17) is over the range of $\mathbf{x}(k)$ and c is the normalization constant.

The recursion (17) shows that the optimal MMSE estimator in the presence of data association uncertainty has the following properties:

- P1. The pdf p_{k+1} is a weighted sum of pdfs, conditioned on the current time association events $A_i(k+1)$, $i = 1, \dots, M(k+1)$, where $M(k+1)$ is the number of mutually exclusive and exhaustive association events at time $k+1$.
- P2. If the exact previous pdf, which is the sufficient statistic, is available, then only the most recent association event probabilities are needed at each time.

However, the number of terms of the mixture in the right-hand side of (17) is, by time $k+1$, given by the product

$$M^{k+1} = \prod_{i=1}^{k+1} M(i) \quad (18)$$

which amounts to an exponential increase in time. This increase is similar to the increase in the number of the branches of the MHT hypothesis tree.

A detailed derivation of the recursion for the optimal estimator can be found in Bar-Shalom et al. (2011).

Track-to-Track Association

In addition to **measurement-to-track association (M2TA)**, an additional class of data association is **track-to-track association (T2TA)**. Following T2TA, **track-to-track fusion (T2TF)** may be performed to (hopefully) improve the overall tracking accuracy. For more details on track fusion, see Bar-Shalom et al. (2011).

It is desired first to test the hypothesis that two tracks pertain to the same target. The optimal test would require using the entire data base (the sequences of measurements that form the tracks) through the present time k and is not practical. In view of this, the test to be presented is based only on the most recent estimates from the tracks. The test based on the state estimates within a time window is discussed in Tian and Bar-Shalom (2009).

Association of Tracks with Independent Errors

Let $\hat{x}^i(k)$ be the estimated state of a target by sensor i with its own information processor. Assume that one has an estimate $\hat{x}^j(k)$ from sensor j , corresponding to the same time. Both can be current estimates or one can be a prediction as long as they pertain to the same time (the second time argument has been omitted for simplicity).

The corresponding covariances are denoted as $P^m(k)$, $m = i, j$. The state estimation errors at different sensors (local trackers),

$$\tilde{x}^i(k) = x^i(k) - \hat{x}^i(k) \quad (19)$$

$$\tilde{x}^j(k) = x^j(k) - \hat{x}^j(k) \quad (20)$$

where x^i and x^j are the corresponding true states, are assumed to be *independent*. This is the **state estimation error independence assumption**.

Remark 2 As shown in the sequel, for *independent sensors*, the state estimation errors for the same target are *dependent* in the *presence of process noise*.

Denote the difference of the two estimates as

$$\hat{\Delta}^{ij}(k) = \hat{x}^i(k) - \hat{x}^j(k) \quad (21)$$

This is the estimate of the difference of the true states

$$\Delta^{ij}(k) = x^i(k) - x^j(k) \quad (22)$$

The **same target hypothesis** is that the true states are equal,

$$H_0 : \quad \Delta^{ij}(k) = 0 \quad (23)$$

while the *different target* alternative is

$$H_1 : \quad \Delta^{ij}(k) \neq 0 \quad (24)$$

While (21) is the appropriate statistic to test whether (22) is zero or not, the rigorous proof of this fact is presented in Bar-Shalom et al. (2011).

The error in the difference between the state estimates

$$\tilde{\Delta}^{ij}(k) = \Delta^{ij}(k) - \hat{\Delta}^{ij}(k) \quad (25)$$

is zero mean and has covariance

$$\begin{aligned} T^{ij}(k) &\triangleq E\{\tilde{\Delta}^{ij}(k)\tilde{\Delta}^{ij}(k)'\} \\ &= E\{[\tilde{x}^i(k) - \tilde{x}^j(k)][\tilde{x}^i(k) - \tilde{x}^j(k)]'\} \end{aligned} \quad (26)$$

given, under the **error independence assumption**, by

$$T^{ij}(k) = P^i(k) + P^j(k) \quad (27)$$

Assuming the estimation errors to be Gaussian, the test of H_0 vs. H_1 – the **T2TA** test – is

Accept H_0 if

$$D \triangleq \hat{\Delta}^{ij}(k)'[T^{ij}(k)]^{-1}\hat{\Delta}^{ij}(k) \leq D_\alpha \quad (28)$$

The threshold D_α is such that

$$P\{D > D_\alpha | H_0\} = \alpha \quad (29)$$

where, e.g., $\alpha = 0.01$. From the Gaussian assumption, the threshold is the $1 - \alpha$ point of the **chi-square distribution** with n_z degrees of freedom (Bar-Shalom et al. 2001)

$$D_\alpha = \chi_{n_z}^2(1 - \alpha) \quad (30)$$

Association of Tracks with Dependent Errors

In the previous section, the association testing was done under the assumption that the estimation errors in these tracks are independent. However, as shown in Bar-Shalom et al. (2011), *whenever there is process noise* (or, in general, motion uncertainty), the track errors based on data from *independent sensors are dependent*.

The *dependence* between the estimation errors $\tilde{x}^i(k|k)$ and $\tilde{x}^j(k|k)$ from the two tracks arises from the *common process noise* which contributes to both errors. This is due to the fact that there is a common motion equation for both trackers.

The testing of the hypothesis that the two tracks under consideration originated from the same target is done in the same manner as before, except for the following modification to account for the *dependence* of their state estimation errors.

The covariance associated with the difference of the estimates

$$\hat{\Delta}^{ij}(k) = \hat{x}^i(k|k) - \hat{x}^j(k|k) \quad (31)$$

is, accounting for the dependence,

$$\begin{aligned} T^{ij}(k) &\triangleq E\{\tilde{\Delta}^{ij}(k)\tilde{\Delta}^{ij}(k)'\} \\ &= E\{[\tilde{x}^i(k|k) - \tilde{x}^j(k|k)][\tilde{x}^i(k|k) \\ &\quad - \tilde{x}^j(k|k)]'\} \end{aligned} \quad (32)$$

and, with the known cross-covariance P^{ij} , is given by the expression

$$\begin{aligned} T^{ij}(k) &= P^i(k|k) + P^j(k|k) \\ &\quad - P^{ij}(k|k) - P^{ji}(k|k) \end{aligned} \quad (33)$$

Note the difference between the above and (27).

Effect of the Dependence

The effect of the dependence between the estimation errors is to *reduce* the covariance of the difference (31) of the estimates. This is due to

the fact that the cross-covariance term reflects a positive correlation between the estimation errors (this is always the case for linear systems).

The Test

The hypothesis testing for the **track-to-track association** with the dependence accounted for is done in the same manner as before in (28), except that the “smaller” covariance from (33) is used in the test statistic, which is, as before, the **normalized distance squared** between the estimates

$$D = \hat{\Delta}^{ij}(k)'[T^{ij}(k)]^{-1}\hat{\Delta}^{ij}(k) \quad (34)$$

The Cross-Covariance of the Estimation Errors

The **cross-covariance recursion** for *synchronized sensors* can be shown to be (see Bar-Shalom et al. 2011)

$$\begin{aligned} P^{ij}(k|k) &\triangleq E[\tilde{x}^i(k|k)\tilde{x}^j(k|k)'] \\ &= [I - W^i(k)H^i(k)] \\ &\quad \cdot \left[F(k-1)P^{ij}(k-1|k-1)F(k-1)' \right. \\ &\quad \left. + Q(k-1) \right] [I - W^j(k)H^j(k)]' \end{aligned} \quad (35)$$

This is a *linear recursion* – a Lyapunov-type equation – and its initial condition is, assuming the initial errors to be uncorrelated,

$$P^{ij}(0|0) = 0 \quad (36)$$

This is a reasonable assumption in view of the fact that the initial estimates are usually based on the initial measurements, which were assumed to have independent errors.

The cross-covariance for the case of asynchronous sensors can be found in Bar-Shalom et al. (2011).

Summary and Future Directions

This entry surveyed the issues involved in data association (specifically M2TA and T2TA) with regard to multitarget-multisensor tracking systems.

The future developments in this topic will be in regard to the use of new feature variables and classification in data association (some preliminary results are in Bar-Shalom et al. (2011)).

Cross-References

- [Estimation for Random Sets](#)
- [Estimation, Survey on](#)

Bibliography

- Bar-Shalom Y, Li XR, Kirubarajan T (2001) Estimation with applications to tracking and navigation: theory, algorithms and software. Wiley, New York
- Bar-Shalom Y, Willett PK, Tian X (2011) Tracking and data fusion. YBS, Storrs
- Blackman SS, Popoli R (1999) Design and analysis of modern tracking systems. Artech House, Norwood
- Mallick M, Krishnamurthy V, Vo BN (eds) (2013) Integrated tracking, classification, and sensor management. Wiley, Hoboken
- Reid DB (1979) An algorithm for tracking multiple targets. IEEE Trans Autom Control 24:843–854
- Striebel C (1965) Sufficient statistics in the optimum control of stochastic systems. J Math Anal Appl 12: 576–592
- Tian X, Bar-Shalom Y (2009) Track-to-track fusion configurations and association in a sliding window. J Adv Inf Fusion 4(2):146–164

Data Rate of Nonlinear Control Systems and Feedback Entropy

Christoph Kawan
Courant Institute of Mathematical Sciences,
New York University, New York, USA

Abstract

Topological feedback entropy is a measure for the smallest information rate in a digital

communication channel between the coder and the controller of a control system, above which the control task of rendering a subset of the state space invariant can be solved. It is defined purely in terms of the open-loop system without making reference to a particular coding and control scheme and can also be regarded as a measure for the inherent rate at which the system generates “invariance information.”

Keywords

Communication constraints · Controlled invariance · Invariance entropy · Minimal data rates · Stabilization

Introduction

In the theory of networked control systems, the assumption of classical control theory that information can be transmitted within control loops instantaneously, lossless, and with arbitrary precision is no longer satisfied. Realistic mathematical models of many important real-world communication and control networks have to take into account general data rate constraints in the communication channels, time delays, partial loss of information, and variable network topologies. This raises the question about the smallest possible information rate above which a given control task can be solved. Though networked control systems can have a complicated topology, consisting of multiple sensors, controllers, and actuators, a first step towards understanding the problem of minimal data rates is to analyze the simplest possible network topology, consisting of one controller and one dynamical system connected by a digital channel with a certain rate in bits per unit time. There is a wealth of literature concerned with the problem of stabilizing a system under different assumptions about the specific coding and control scheme, in this context. However, with few exceptions, mainly linear systems (both deterministic and stochastic) have been considered. A comprehensive and detailed overview of this literature until 2007 can be found in the survey Nair et al. (2007). The first

systematic approach to the problem of minimal data rates for set invariance and stabilization of (deterministic, nonlinear) control systems was presented in Nair et al. (2004), where the notion of *topological feedback entropy* was introduced. This quantity, defined in terms of the open-loop control system, is a measure for the smallest data rate a communication channel may have if the system is supposed to solve the control task of rendering a subset of the state space invariant. Other challenges that digital communication channels come along with are not yet taken into account here.

Feedback entropy was first introduced in Nair et al. (2004), using a similar approach via open covers as in the definition of topological entropy of for classical dynamical systems in Adler et al. (1965). In Colonius and Kawan (2009), a quantity named *invariance entropy* was defined which later turned out to be equivalent to the feedback entropy of Nair et al. (cf. Colonius et al. 2013). The notion of invariance entropy has been further studied in the papers Kawan (2011a), Kawan (2011b), and Kawan (2011c). Several variations and generalizations have been introduced in Colonius (2010), Colonius (2012), Colonius and Kawan (2011), Da Silva (2013), and Hagihara and Nair (2013). The research monograph Kawan (2013) provides a comprehensive presentation of the results obtained so far in the deterministic case.

Definition

Topological feedback entropy is a nonnegative real-valued quantity which serves as a measure for the smallest possible data rate in a digital channel, connecting a coder to a controller, above which the controller is able to generate inputs which guarantee invariance of a given subset of the state space. In the literature, one finds several slightly differing versions. The original definition given in Nair et al. (2004) is (with minor modifications) as follows. Consider a discrete-time control system

$$x_{k+1} = F(x_k, u_k) = F_{u_k}(x_k), \quad k \geq 0,$$

with $F : X \times U \rightarrow X$, where X is a topological space and U a nonempty set such that $F_u : X \rightarrow X$ is continuous for every $u \in U$. The transition map associated to this system is

$$\begin{aligned} \varphi : \mathbb{N}_0 \times X \times U^{\mathbb{N}_0} &\rightarrow X, \\ \varphi(k, x, (u_n)) &:= F_{u_{k-1}} \circ \cdots \circ F_{u_1} \circ F_{u_0}(x). \end{aligned}$$

A compact subset $K \subset X$ with nonempty interior is (*strongly*) *controlled invariant* if for every $x \in K$ there is an input $u \in U$ such that $F_u(x) \in \text{int}K$. A triple $(\mathcal{A}, \tau, G)$ is called an *invariant open cover* of K if $\mathcal{A}$ is an open cover of K , τ is a positive integer, and $G : \mathcal{A} \rightarrow U^\tau$ is a map with components $G_0, G_1, \dots, G_{\tau-1}$ which assign control values to all sets in $\mathcal{A}$ such that for every $A \in \mathcal{A}$ the finite sequence of controls $G(A)$ yields $\varphi(k, A, G(A)) \subset \text{int}K$ for $k = 1, 2, \dots, \tau$. The *entropy* of $(\mathcal{A}, \tau, G)$ is defined as follows. For every sequence $\alpha = (A_i)_{i \geq 0}$ of sets in $\mathcal{A}$ one defines an associated sequence of controls by

$$\begin{aligned} \underline{u}(\alpha) &= (u_0, u_1, u_2, \dots) \quad \text{with } (u_l)_{l=(i-1)\tau}^{i\tau-1} \\ &= G(A_{i-1}) \quad \text{for all } i \geq 0, \end{aligned}$$

and for every $j \geq 1$ a set

$$\begin{aligned} B_j(\alpha) &:= \{x \in X : \varphi(i\tau, x, \underline{u}(\alpha)) \in A_i \\ &\quad \text{for } i = 0, 1, \dots, j-1\}. \end{aligned}$$

The family $\mathcal{B}_j := \{B_j(\alpha) : \alpha \in \mathcal{A}^{\mathbb{N}_0}\}$ is an open cover of K . Letting $N(\mathcal{B}_j|K)$ denote the minimal cardinality of a finite subcover, the entropy of $(\mathcal{A}, \tau, G)$ is

$$\begin{aligned} h(\mathcal{A}, \tau, G) &:= \lim_{j \rightarrow \infty} \frac{1}{j\tau} \log_2 N(\mathcal{B}_j|K) \\ &= \inf_{j \geq 1} \frac{1}{j\tau} \log_2 N(\mathcal{B}_j|K). \end{aligned}$$

Finally, the topological feedback entropy (TFE) of K is given by

$$h_{\text{fb}}(K) := \inf_{(\mathcal{A}, \tau, G)} h(\mathcal{A}, \tau, G),$$

where the infimum is taken over all invariant open covers of K .

A conceptually simpler but equivalent definition, introduced in Colonius and Kawan (2009), is the following. A subset $\mathcal{S} \subset U^\tau$ is called (τ, K) -spanning if for every $x \in K$ there is $\underline{u} \in \mathcal{S}$ with $\varphi(k, x, \underline{u}) \in \text{int}K$ for $k = 1, \dots, \tau$. Writing $r_{\text{inv}}(\tau, K)$ for the minimal cardinality of such a set, it can be shown that

$$\begin{aligned} h_{\text{fb}}(K) &= \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \log_2 r_{\text{inv}}(\tau, K) \\ &= \inf_{\tau \geq 1} \frac{1}{\tau} \log_2 r_{\text{inv}}(\tau, K). \end{aligned}$$

This definition and several variations of it are mostly referred to by the name *invariance entropy* instead of feedback entropy. The intuition behind this definition is that a controller which receives a certain amount of information about the state, say n bits, can generate at most 2^n different control sequences to steer the system on a finite time interval and hence, the number of control sequences needed to accomplish the control task on this interval is a measure for the necessary amount of information.

The different variations of feedback or invariance entropy which can be found in the literature are briefly summarized as follows. For simplicity, in this entry we refer to several of these variations by the name *(topological) feedback entropy*:

- (i) Instead of requiring that trajectories enter the interior of K after one step of time, one can allow for a waiting time τ_0 before entering $\text{int}K$.
- (ii) One can require that trajectories stay in K instead of $\text{int}K$ or that they stay in an arbitrarily small neighborhood of K , respectively.
- (iii) One can restrict the set of initial states to a subset of $K' \subset K$. In this case, a set $\mathcal{S}$ of control sequences is (τ, K', K) -spanning if for every $x \in K'$ there is $\underline{u} \in \mathcal{S}$ with $\varphi(k, x, \underline{u}) \in \text{int}K$ for $k = 1, \dots, \tau$.
- (iv) Feedback entropy can be defined for other classes of systems, e.g., continuous-time

deterministic systems, random control systems, or systems with piecewise continuous right-hand sides.

There is also a local version of topological feedback entropy (LTFE) which measures the smallest data rate for local uniform asymptotic stabilization at an equilibrium. Also for other control tasks there have been attempts to define corresponding versions of feedback or invariance entropy.

Comparison to Topological Entropy

Though there are similarities in the definitions of TFE and topological entropy of dynamical systems, which also reflect in similar properties, there is no direct relation between these two quantities. Topological entropy detects exponential complexity in the orbit structure of a dynamical system. In contrast, TFE measures the complexity of the control task to keep a system in a subset of the state space by applying appropriate inputs. If no escape from this subset is possible, the TFE is zero, no matter how complicated the orbit structure is. Hence, topological entropy is sensitive to the local behavior of the system, while TFE in general is not. Interpreted in terms of information rates, topological entropy is a measure for the largest average rate of information about the initial state a dynamical system can generate. TFE measures the smallest rate of information about the state of the system above which a controller is able to render the set invariant. It should also be mentioned that topological entropy was first introduced as a topological counterpart of the measure-theoretic entropy defined by Kolmogorov and Sinai, and that the two notions are related by the variational principle, which asserts that the topological entropy is the supremum of the measure-theoretic entropies with respect to all invariant probability measures of the given system. For TFE, so far no convincing measure-theoretic approach exists. An excellent survey on the entropy theory of dynamical systems can be found in Katok (2007).

The Data Rate Theorem

The *data rate theorem* for the TFE confirms that the infimal data rate in a coding and control loop which guarantees strong controlled invariance of a set K is equal to $h_{\text{fb}}(K)$. More precisely, suppose that a sensor which measures the state of the system at discrete times $\tau_k = k\tau$, $k = 0, 1, 2, \dots$, is connected to a coder which at time τ_k has a finite alphabet S_k of symbols available. The measured state is coded by use of this alphabet and the corresponding symbol is sent via a noiseless digital channel to a controller which generates an input sequence of length τ . This sequence is used to steer the system until the next symbol arrives at time τ_{k+1} . The associated *asymptotic average bit rate*, which depends on the sequence $S = (S_k)_{k \geq 0}$ of coding alphabets, is given by

$$R(S) = \lim_{k \rightarrow \infty} \frac{1}{k\tau} \sum_{i=0}^{k-1} \log_2 |S_i|.$$

If the limit does not exist, one may replace it with $\liminf$ or $\limsup$. The data rate theorem establishes the equality

$$h_{\text{fb}}(K) = \inf_S R(S),$$

where the infimum is taken over all coding and control loops which guarantee strong controlled invariance of K , i.e., for initial states in K they generate trajectories $(x_k)_{k \geq 0}$ with $x_k \in \text{int}K$ for $k = 1, 2, \dots$. Similar data rate theorems can be proved for other variants of feedback entropy. In particular, the data rate theorem for the LTFE asserts that the infimal bit rate for local uniform asymptotic stabilization at an equilibrium is given by the LTFE. Proofs of different data rate theorems can be found in Nair et al. (2004), Hagihara and Nair (2013), and Kawan (2013).

Estimates and Formulas

Linear Systems

For linear systems, under reasonable assumptions, the feedback entropy is given by the sum of

the unstable eigenvalues of the dynamical matrix, i.e., if the system is given by $x_{k+1} = Ax_k + Bu_k$, then

$$h_{\text{fb}}(K) = \sum_{\lambda \in \text{Sp}(A)} \max \{0, n_\lambda \log_2 |\lambda|\}, \quad (1)$$

where $\text{Sp}(A)$ denotes the spectrum of A and n_λ is the algebraic multiplicity of the eigenvalue λ (cf. Colonius and Kawan 2009). It is worth to mention that the TFE therefore coincides with the topological entropy of the uncontrolled system $x_{k+1} = Ax_k$, as defined by Bowen for maps on non-compact metric spaces (cf. Bowen 1971). However, this is a special property of linear systems and is related to the facts that (i) there is no difference between the local and the global dynamical behavior of uncontrolled linear systems and that (ii) the control sequence does not affect the exponential complexity of the dynamics, since it only appears as an additive term in the transition map of the system. Formula (1) is in correspondence with several former results on minimal data rates for stabilization of linear systems. Thinking of the definition of feedback entropy via spanning sets of control sequences, its interpretation is that in order to guarantee invariance of a bounded set, the only reason for exponential growth of the number of necessary inputs as time increases is the volume expansion of the open-loop system in the unstable subspace.

Upper Bounds Under Controllability Assumptions

If the state space of the control system is a differentiable manifold and the right-hand side is continuously differentiable, under certain controllability assumptions upper bounds for the feedback entropy can be formulated in terms of the Lyapunov exponents of periodic trajectories (for the concept of Lyapunov exponents, see, e.g., Barreira and Valls 2008) (cf. Nair et al. 2004; Kawan 2011b, 2013). More precisely, if there is a periodic trajectory in the interior of the given set K such that the linearization along this trajectory is controllable, and if complete approximate controllability holds on the interior of K (cf. Colonius and Kliemann 2000), then

$$h_{\text{fb}}(K) \leq \sum_{\lambda} \max\{0, n_{\lambda} \lambda\}, \quad (2)$$

where the sum is taken over all Lyapunov exponents λ of the periodic trajectory and n_{λ} denotes the multiplicity of λ . Using the definition of feedback entropy in terms of (τ, K) -spanning sets, one can prove this by constructing spanning sets of control functions which first steer all initial states in K into a small neighborhood of a point on the given periodic orbit and then, by use of local controllability, keep the corresponding trajectories in a neighborhood of the periodic trajectory for arbitrary future times. Similar ideas first have been used in Nair et al. (2004) to prove that the LTFE at an equilibrium is given by the sum of the unstable eigenvalues of the linearization about this equilibrium. For systems given by differential equations the upper estimate (2) can be improved under additional regularity assumptions. Assuming that the system is smooth and satisfies the strong jet accessibility rank condition (cf. Coron 1994), one can show that both assumptions, controllability of the linearization and periodicity, can be omitted. The only restriction that remains is that the trajectory must not leave a compact subset of the interior of K . However, in the case of nonperiodic trajectories, the sum of the positive Lyapunov exponents has to be replaced by the maximal Lyapunov exponent of the induced linear flow on the exterior bundle of the manifold. For control-affine systems, strong jet accessibility can be weakened to local accessibility. In general, it is unlikely that such upper bounds are tight, since they are related to very specific control strategies for making the given set invariant.

Lower Bounds, Volume Growth Rates, and Escape Rates

A general approach to obtain lower bounds of feedback entropy is via a volume growth argument, which in its simplest form works as follows. Every (τ, K) -spanning set $\mathcal{S}$ defines a cover of K , consisting of the sets (cf. Kawan 2011a, c)

$$K_{\tau, \underline{u}} = \{x \in K : \varphi(k, x, \underline{u}) \in \text{int} K,$$

$$1 \leq k \leq \tau\}, \quad \underline{u} \in \mathcal{S}.$$

It follows that $\varphi_{\tau, \underline{u}}(K_{\tau, \underline{u}}) \subset K$ and hence, since K is bounded, the volume expansion under $\varphi_{\tau, \underline{u}} = \varphi(\tau, \cdot, \underline{u})$ gives upper bounds for the volumes of the sets $K_{\tau, \underline{u}}$, which result in a lower bound for the number of these sets. For instance, the lower estimate in (1) can be established by applying this argument to the system which arises by projection of the given linear system to the unstable subspace of the uncontrolled part $x_{k+1} = Ax_k$. A refinement of this idea also leads to lower estimates of feedback entropy for inhomogeneous bilinear systems in terms of volume growth rates or Lyapunov exponents on unstable bundles, respectively. For nonlinear systems, in general only very rough estimates can be obtained by this method. However, a variation of the volume growth argument leads to a lower bound of the form

$$h_{\text{fb}}(K) \geq -\liminf_{\tau \rightarrow \infty} \frac{1}{\tau} \log \sup_{\underline{u}} \mu(K_{\tau, \underline{u}}),$$

where μ denotes a reference measure on the state space. The right-hand side of this inequality can be considered as a uniform *escape rate* from the set K , which under sufficiently strong hyperbolicity assumptions can be estimated in terms of other quantities such as Lyapunov exponents and dynamical entropies. Key references for escape rates in the classical theory of dynamical systems are (Young (1990) and Demers and Young (2006)).

Summary and Future Directions

The theory of feedback entropy for finite-dimensional deterministic systems is very far from being complete. The currently available results only give valuable information in very regular situations, and even those are not fully understood. For the further development of this theory, it will be necessary to combine control-theoretic methods with techniques from different fields such as classical, random, and nonautonomous dynamical systems. Some of the

main focuses of future research will probably be the following:

- The generalization of feedback entropy to more complex network topologies
- The formulation of a feedback entropy theory for stochastic systems
- The development of a probabilistic (resp. measure-theoretic) version of feedback entropy for both deterministic and stochastic systems, which is related to the topological version via a variational principle
- The numerical computation of feedback entropy

Cross-References

- [Networked Systems](#)
- [Quantized Control and Data Rate Constraints](#)

Bibliography

- Adler RL, Konheim AG, McAndrew MH (1965) Topological entropy. *Trans Amer Math Soc* 114:309–319
- Barreira L, Valls C (2008) Stability of nonautonomous differential equations. *Lecture notes in mathematics*, vol. 1926. Springer, Berlin
- Bowen R (1971) Entropy for group endomorphisms and homogeneous spaces. *Trans Am Math Soc* 153:401–414
- Colonius F (2010) Minimal data rates and invariance entropy. *Electronic Proceedings of the conference on mathematical theory of networks and systems (MTNS)*, Budapest, 5–9 July 2010
- Colonius F (2012) Minimal bit rates and entropy for stabilization. *SIAM J Control Optim* 50:2988–3010
- Colonius F, Kawan C (2009) Invariance entropy for control systems. *SIAM J Control Optim* 48:1701–1721
- Colonius F, Kawan C (2011) Invariance entropy for outputs. *Math Control Signals Syst* 22: 203–227
- Colonius F, Kliemann W (2000) *The dynamics of control*. Birkhäuser-Verlag, Boston
- Colonius F, Kawan C, Nair GN (2013) A note on topological feedback entropy and invariance entropy. *Syst Control Lett* 62:377–381
- Coron J-M (1994) Linearized control systems and applications to smooth stabilization. *SIAM J Control Optim* 32:358–386
- Da Silva AJ (2013) Invariance entropy for random control systems. *Math Control Signals Syst* 25: 491–516
- Demers MF, Young L-S (2006) Escape rates and conditionally invariant measures. *Nonlinearity* 19:377–397
- Hagihara R, Nair GN (2013) Two extensions of topological feedback entropy. *Math Control Signals Syst* 25:473–490
- Katok A (2007) Fifty years of entropy in dynamics: 1958–2007. *J Mod Dyn* 1:545–596
- Kawan C (2011a) Upper and lower estimates for invariance entropy. *Discret Contin Dyn Syst* 30:169–186
- Kawan C (2011b) Invariance entropy of control sets. *SIAM J Control Optim* 49:732–751
- Kawan C (2011c) Lower bounds for the strict invariance entropy. *Nonlinearity* 24:1910–1935
- Kawan C (2013) Invariance entropy for deterministic control systems – an introduction. *Lecture notes in mathematics* vol 2089. Springer, Berlin
- Nair GN, Evans RJ, Mareels IMY, Moran W (2004) Topological feedback entropy and nonlinear stabilization. *IEEE Trans Autom Control* 49:1585–1597
- Nair GN, Fagnani F, Zampieri S, Evans RJ (2007) Feedback control under data rate constraints: an overview. *Proc IEEE* 95:108–137
- Young L-S (1990) Large deviations in dynamical systems. *Trans Am Math Soc* 318:525–543

Deep Learning in a System Identification Perspective

Thomas B. Schön¹ and Lennart Ljung²

¹Department of Information Technology,
Uppsala University, Uppsala, Sweden

²Division of Automatic Control, Department of
Electrical Engineering, Linköping University,
Linköping, Sweden

Abstract

The use of deep learning for sequence learning problems and system identification are intimately linked, and interesting opportunities exist on this cross section. The aim of this chapter is to briefly introduce deep learning from a system identification perspective and to explain some of the most apparent links between these two topics.

Keywords

Neural networks · Stochastic optimization · Nonlinear dynamical systems · Sequence learning · Recurrent neural networks

Introduction

Deep learning is based on the idea that when it comes to representing a function, a deep, hierarchical model can be considerably more efficient than a shallow model. While some theoretical explanations are available, the most compelling evidence is provided by all the practical applications where deep learning have changed the landscape by significantly improving the performance compared to earlier technologies in, for example, computer vision (Krizhevsky et al. 2012), speech recognition (Hinton et al. 2012), and natural language processing (Mikolov et al. 2013). While the model underlying all of deep learning – the neural network – has been around for more than 70 years (McCulloch and Pitts 1943), all recent success stories like the ones referenced above are among the key reasons for why the neural network model has yet again become so popular. Other important factors contributing to the current popularity of deep learning are massively increased datasets, availability of larger-scale compute resources, significantly improved software tools, substantial investment from industry, and finally some architectural and algorithmic innovations.

System identification is about building mathematical models of dynamic systems using observed input-output data; see, e.g., Cross-Ref ► “[System Identification: An Overview](#)”. So, system identification and deep learning are both about inferring models from observed data. This contribution is about exploring some links between the topics.

One link is that the conceptual problem structures for the two problems are quite similar. This will be exploited in sections “[The Setup and the Model Structure](#)” and “[Stochastic Optimization Is Used to Train the Model](#)”.

The other link is that system identification of nonlinear systems really amounts to estimation of a static nonlinearity in a suitable node in a signal network; see Schoukens and Ljung (2019). This means that applying deep learning to learn that nonlinearity implies that deep learning can be directly incorporated into the system identification world. This will be discussed further in

section “[Constructing Deep Learning Architectures Suitable for System Identification](#)”.

Deep Learning Background

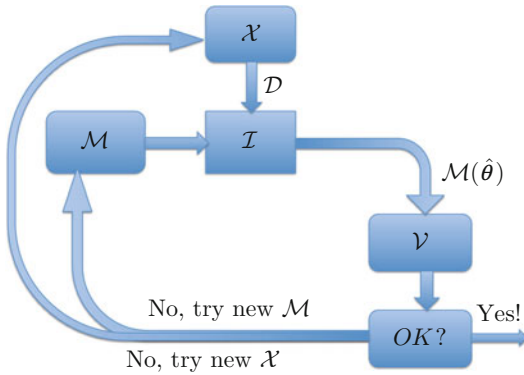
The Setup and the Model Structure

Machine learning and deep learning are about inferring models from observed data. Such models can reproduce, if not explain, the behaviour of the system that generated the data. The core of all learning can be said to be *function estimation*, i.e., measure noisy observations of a function between two spaces $\mathcal{X}$ to $\mathcal{Y}$

$$y_t = \mathcal{F}(x_t) + \text{noise} \quad t = 1, \dots, T \quad (1)$$

and infer the function $\mathcal{F}$, where $x_t \in \mathcal{X}$ denotes the input and $y_t \in \mathcal{Y}$ denotes the output. This is like a classical curve-fitting problem, but the spaces may be of very general nature: $\mathcal{X}$ could be signals, images, texts, etc., and $\mathcal{Y}$ could be numerical values or classifications (like “text is offensive”). It is the broad range of possible spaces that allows the problem formulation to be applied to a wide variety of tasks, which is the reason for the current intense interest in the field. But for standard numerical spaces $\mathcal{X}$ and $\mathcal{Y}$, the formulation (1) remains an example of standard *nonlinear regression*. Most applications of machine learning have a *black-box* view of $\mathcal{F}$, i.e., no physical knowledge about the function is employed, and flexible mappings are used when estimating it.

This also means that the system identification perspective (see Fig. 1) can be applied to learning problems. The key elements are the method of model fitting, block $\mathcal{I}$, and the model structure $\mathcal{M}$ in that figure. For successful learning, the fitting method should allow handling large model structures with many parameters, and the model structure should be able to capture with good accuracy any reasonable function $\mathcal{F}$ that can be encountered in the application: The model structure should thus be a *universal approximator*. Section “[Stochastic Optimization Is Used to Train the Model](#)” (below) deals with optimization methods for fitting, and Section



Deep Learning in a System Identification Perspective, Fig. 1 The identification work loop. See also Fig. 4 in Cross-Ref “► [System Identification: An Overview](#)”

“[Constructing Deep Learning Architectures Suitable for System Identification](#)” deals with particular architectures for universal approximation structures used in deep learning.

But, simplistically, the bottom line is that the model for $\mathcal{F}$ can be seen as a piecewise linear, continuous function over an adaptive grid in $\mathcal{X}$. That means that the structure is defined by a number of grid points in $\mathcal{X}$, often chosen guided by the data $\{x_t, y_t\}$. The function is then defined by linear interpolation between the values at the grid points.

Such piecewise linear approximations can be created in several ways, e.g., in *regression trees*, (Breiman et al. 1984), but in machine and deep learning, the model structure is typically constructed from *neural networks* (see, e.g., Eq. (18) in the section “Nonlinear Function Estimators” in crossref “► [Nonlinear System Identification: An Overview of Common Approaches](#)”). Basically, in the simple *one-hidden-layer neural network* model, the mapping from x to y in a feedward neural network is (think of x as a scalar for simplicity)

$$y_t = \mathcal{F}(x_t) = \sum_{l=1}^m \gamma_l \kappa(\alpha_l(x_t - \beta_l)). \quad (2)$$

The so-called activation function $\kappa(z)$ can be chosen as a nonlinear function in various ways. A classical choice is the *sigmoid*

$$\kappa(z) = \frac{1}{1 + e^{-z}} \quad (3)$$

giving rise to a *sigmoidal network*. Nowadays, a dominating choice is the rectified linear unit (ReLU) function

$$\kappa(z) = \max(z, 0), \quad (4)$$

which means that the sum (2) will be piecewise linear and continuous with breakpoints in $\beta_l, l = 1, \dots, m$. The numbers $\theta = \{\alpha_l, \beta_l, \gamma_l\}_{l=1}^m$ constitute the parameterization of the model.

As in all estimation problems, the trade-off between *bias* and *variance* is crucial. See section with that name in Cross-Ref “► [System Identification: An Overview](#)”. Typically, however, this aspect is not so much stressed in most of the contemporary machine/deep learning literature. An important early contribution is Geman et al. (1992).

The bias is determined by how well the grid can approximate the function. The more grid points, the better fit is possible and the less bias. Let the number of grid points be m . The variance is determined by the number of parameters $d = \dim \theta$ that are estimated in the model structure. More parameters give higher variance. The total error in the function fit is the sum of bias (squared) and variance, so a small error results for a large m in combination with a small d . This is the crucial dilemma in function estimation.

For a simple feedforward neural network with the ReLU activation function, the number of parameters d is 3 times the number of breakpoints m . This illustrates the problem of doing advanced bias/variance trade-offs.

Deep neural networks are model structures created from simple feedforward neural networks by concatenation of the regressors in several *layers*: The m terms in (2), $x_t^{(2)} = \kappa(\alpha_l(x_t - \beta_l))$ (“the hidden units”) can be

interpreted as new regressors and be subjected to the same mapping (2):

$$y_t = \mathcal{F}(x_t) = \sum_{l=1}^{m^{(2)}} \gamma_l^{(2)} \kappa(\alpha_l^{(2)}(x_t^{(2)} - \beta_l^{(2)})), \quad (5)$$

thus giving a net with a *second layer* with $m^{(2)}$ hidden units. This can be repeated a number of times, viewing $x_t^{(3)} = \kappa(\alpha_l^{(2)}(x_t^{(2)} - \beta_l^{(2)}))$ as regressors in a third layer, etc. This is the way to create deep networks with an arbitrary number of hidden layers for the simple feedforward net. (A concrete example will be given in section “[A Case Study: The Silverbox Example](#)” below.) The specific design of a neural network is very much an art that still impacts the overall performance significantly.

A deep neural network constructed from many simple ReLU layers will still be a piecewise linear function over an adaptive grid. So what is then gained? *The point is that while the number of breakpoints will increase drastically with the depth of the net, the number of parameters will not increase at the same rate in a deep network*, so there is a potential for a better bias/variance trade-off. This of course means that there will be hidden dependencies between the breakpoints. The parameterization is so to speak less generous than the possible model flexibility. This necessitates “subroutines” in the parameterization to deal with patterns that reoccur in different parts of the function. But it also allows for the possibility to form these automatically as part of the parameter fitting process. This makes it possible to handle the bias/variance trade-off more efficiently.

Stochastic Optimization Is Used to Train the Model

To make use of deep learning, the unknown parameters θ of the underlying deep neural network model needs to be trained from data. The standard solution involves solving a non-convex optimization problem of the form

$$\hat{\theta} = \arg \min_{\theta} J(\theta), \quad \text{where}$$

$$J(\theta) = \frac{1}{T} \sum_{t=1}^T L(x_t, y_t, \theta). \quad (6)$$

The cost function $J(\theta)$ is a sum of the loss function $L(x_t, y_t, \theta)$ over all T samples in the available estimation (training) dataset. A commonly used loss function in this setting is provided by the squared prediction error, namely, $L(x_t, y_t, \theta) = (y_t - \hat{y}_t)^2$, where $\hat{y}_t$ denotes the prediction of y_t generated by the model. For classification problems, the cross-entropy loss is popular, which for two class problems is given by $L(x_t, y_t, \theta) = -(y_t \log(\hat{y}_t) + (1 - y_t) \log(1 - \hat{y}_t))$. Besides the two loss functions explicitly mentioned above, there are certainly more options (see, e.g., for Goodfellow et al. (2016) details). Due to the non-convex nature of (6), we have to resort to numerical solutions, most of which proceed in an iterative fashion according to:

1. Select an initial value θ_0 .
2. Update via a step in the search direction p_t according to

$$\theta_{i+1} = \theta_i + \gamma p_t, \quad \text{for } i = 1, 2, \dots \quad (7)$$

where the length of the step is controlled by the *learning rate* λ . A simple search direction is provided by the negative gradient direction $p_t = -\nabla_{\theta} J(\theta_t)$.

3. Terminate when a certain criterion is fulfilled and use the last θ_i as $\hat{\theta}$.

Variants of the above scheme – often with a carefully computed search direction p_t – are indeed also used in many approaches to system identification.

In most deep learning applications, the sheer size of the dataset prevents us from evaluating the cost function (6) and its derivatives using all the data at the same time. Instead, the estimation dataset is partitioned into many smaller sets, so-called *mini-batches* typically containing $n_b=10$,

$n_b = 100$ or $n_b = 1000$ data samples each. The model is then trained according to a scheme similar to the one above on each mini-batch. Since we are only using a small part of the dataset for each mini-batch, the values we obtain for the cost function and its derivatives are merely estimates of the corresponding true values, implying that we are now in fact faced with a stochastic optimization problem.

We transition from one mini-batch to the next by using the estimate from the current mini-batch as initial value for the next mini-batch. The resulting estimates are indeed noisy, and there are elaborate techniques – referred to as noise reduction – for averaging these values to compute a final point estimate (see Bottou et al. (2018) for a recent overview). These noise reduction approaches work well in practice, and they can also be analyzed theoretically. The n_b samples constituting each mini-batch are typically drawn at random, often implemented by a random shuffle of the estimation data before dividing it into mini-batches. One complete pass through the estimation data is called an *epoch*. The stochastic optimizers used are often improved developments of the classic Robbins-Monro stochastic approximation algorithm (Robbins and Monro 1951). A good contemporary introduction is provided in Bottou et al. (2018), clearly showing that stochastic optimization is currently a very active research problem due to its importance in deep learning and other large-scale machine learning problems. It is also worth mentioning that the structure of the neural network should be exploited – via the chain rule and reuse of information – in computing the gradient, which is known as backpropagation (Williams et al. 1986).

Constructing Deep Learning Architectures Suitable for System Identification

Using the language of machine learning, the system identification problem is a so-called *sequence learning* problem, where the task is to learn a model for data arriving in a sequential fashion. We will in this section provide a very

brief overview of current trends when it comes to suitable deep learning models for system identification.

There exist a multitude of deep learning models specifically tailored for sequence learning. The classic example is the recurrent neural network (RNN) (Williams et al. 1986), which has an internal hidden (deterministic) state h_t – akin to a state space model – allowing an RNN to model temporal sequences. The RNN is described by the following dynamical model

$$h_t = f_\theta(x_t, h_{t-1}), \quad (8a)$$

$$y_t = h_\theta(x_t, h_t), \quad (8b)$$

where f_θ and h_θ are deterministic nonlinear functions and θ denotes the parameters governing them. Note that this is closely related to the standard nonlinear state space model. While the information flow in a deep feedforward neural network is fairly straightforward, this is not the case for an RNN, in which feedback effects must be considered; an informative exploration of this topic is available in Pascanu et al. (2014).

There are three basic ways in which we can construct a deep RNN, namely, in using deep models to describe the hidden-to-hidden transition (h_{t-1} to h_t), the input-to-hidden transition (x_t to h_t), and finally the hidden-to-output transition (h_t to y_t). The key difference is that in a feedforward neural network, the information flows from the input x_t sequentially through the network to the output y_t , whereas in an RNN, the information travels in *loops* through the network. This implies that information that is fed into an RNN can stay in its memory (via the hidden states h_t) for an extended period of time, recall (8).

Two popular extensions of the RNN are the long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) and the gated recurrent unit (GRU) (Cho et al. 2014). These architectures are specifically designed to remember information about states in the more distant past, often referred to as long-term dependencies. If the underlying phenomenon is Markovian, then we know that the current state is all we need in predicting the future and the basic RNN (8) is all

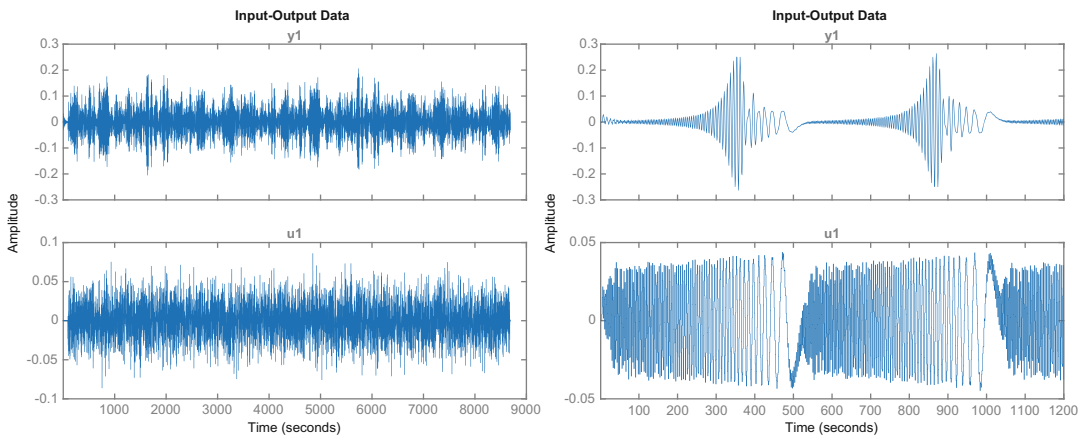
we need. However, there are many phenomena that are non-Markovian, implying that we also need information about states in the more distant past, often referred to as long-term dependencies, in which case the basic (possibly deep) RNN is not enough. The LSTM and the GRU are specifically designed for learning these long-term dependencies.

It has also been shown (Bai et al. 2018) that variations of the popular convolutional neural network (CNN) is highly effective for problems with sequential structure. A CNN can be described as a regularized version of a multilayer perceptron, often used in image processing. See Andersson et al. (2019) for some initial investigations of the use of CNNs for nonlinear system identification.

This is not the place for introducing these deep neural network models in detail, but it is important to note their shared key characteristics, namely, the deep neural networks ability to *automatically* extract highly useful features from the raw data. This is the reason for the name *end-to-end learning*, since it alleviates painful manual preprocessing of the data. For many applications of deep learning, it is indeed so that manual intervention degrades the overall performance. This automatic feature extraction is intimately linked to the fact that – as was mentioned above – patterns recurring in different parts of the function are handled more efficiently. A problem

inherent in deep neural networks is that they do not encode much structural knowledge and this is where classical system identification models become interesting and useful.

Classical models employed in solving system identification problems include, for example, autoregressive (AR), autoregressive moving average (ARMA), Box-Jenkins (BJ), and the state space model (SSM). A straightforward merger of the deep learning models and these classical model structures involves simply representing the unknown functions with a deep neural network. For example, in the case of a nonlinear AR model where $y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-n_y}) + e_t$ and e_t describes the noise, the function f can be encoded using a deep neural network. However, there is currently a clear trend towards more elaborate mergers of these models. One natural idea that remains active is to augment the deterministic RNN (8) with a latent *random* variable (Boulanger-Lewandowski et al. 2012; Osendorfer and Bayer 2014; Fabius et al. 2014; Chung et al. 2015; Rangapuram et al. 2018). In this way the model gains additional expressiveness in that it can now also make use of – carefully designed – random noise in explaining the available measurements. As one concrete example, we mention the construction offered by Rangapuram et al. (2018), involving a deterministic RNN where each hidden state feeds into the parameters of a state space model.



Deep Learning in a System Identification Perspective, Fig. 2 The Silverbox data. Left: estimation data, Right: validation data

A Case Study: The Silverbox Example

A data set that has often been used as a nonlinear test case is the so-called Silverbox example. This is a forced Duffing oscillator that is an electronic circuit that mimics a mechanical system with a cubic hardening spring. The experiment is described in Schoukens et al. (2016), and the data is available on www.nonlinearbenchmark.org/#Silverbox. Two sets of data were collected with different excitations, depicted in Fig. 2. The

estimation data are used to fit a model, and the validity of the model is checked by how well it reproduces the validation data. That is, the model is simulated using the validation data input, and the discrepancy between model output and measured validation data output is determined. This is a standard system identification task.

To load the estimation and validation data, the following MATLAB sequence can be used (files obtained from the website above)

```
load SNLS80mV.mat
fs = 1e7/2^14;
edat = iddata((V2(30550:38500)', V1(30550:38500)', 1/fs)
load Schroeder80mV
ySchroeder = V2(10585:10585+1023);
uSchroeder = V1(10585:10585+1023);
vdat=iddata(ySchroeder', uSchroeder', 1/fs);
```

Note that the validation data (essentially swept sinusoids) have a quite different character from the estimation data (random phase multisines). This means special demands for the model to pick up essential features in the generation of the output – not only to reproduce observed output behavior.

To check models, first estimate a common linear, so-called Box-Jenkins model (see, e.g.,

Chapter 4 in Ljung 1999) with four poles, four zeros, and a second-order noise transfer function

```
mbj=bj(edat, [4 4 2 2 0]);
```

The accuracy of this model on validation data is computed by

```
[ys, Fit]=compare(vdat, mbj)
Fit = 29.71 % (percent of the output variation reproduced
              by the model)
```

A sigmoidal one layer neural network (2) and (3) model with $m = 36$ hidden nodes is estimated and validated by

```
mnn = nlarx(edat, [4 4 0], sigmoidnet('NumberOfUnits', 36));
[ys, Fit]=compare(vdat, mnn)
Fit = 60.04
```

A deep learning model (5) with six layers of cascaded feedforward nets with six nodes each is estimated and validated by

```
net=cascadeforwardnet([6,6,6,6,6,6]);N2=neuralnet(net);
mN2=nlrx(edat,[4 4 0],N2);
[Ys, Fit] = compare(vdat,mN2)
Fit = 99.30
```

The example (that uses MATLAB's system identification and deep learning toolboxes) illustrates a number of things:

- Deep learning for modeling dynamic systems can be approached very much like a conventional system identification problem with the same type of syntax.
- The deep model shows considerable advantages compared to linear models and conventional neural network models with the same number of nodes.

Summary and Future Directions

Deep learning has transformed several scientific fields, but it has so far reached surprisingly little traction within the field of system identification. We believe that this is about to change, and with this chapter, we have tried to establish a common ground for bringing the two areas closer to each other. As concrete ideas for how to make this happen, we end by outlining a few ideas for future work.

The current system identification working cycle involves a fair bit of pre-processing work by the engineer. While the deep learning models also need certain pre-processing, they offer a highly interesting and remarkable ability of learning complex patterns automatically from raw data. This capability is indeed the key reason for the recent success of deep learning. There are interesting possibilities in exploiting these capabilities in system identification problems involving large datasets. Secondly, we recall the possibilities of combining deep learning architectures with standard models of system identification, like the state space model. This represents an active strand of research, and there are several constructions available already (see, e.g., Boulanger-Lewandowski et al. (2012),

Osendorfer and Bayer (2014), Fabius et al. (2014), Chung et al. (2015), and Rangapuram et al. (2018)). Inspired by the initial success of these constructions, we believe that there is more to be understood here and more models to be explored. Thirdly, we mention the possibility of making use of theoretical tools from systems theory to help understand why deep learning works so well and to establish fundamental properties describing deep learning. This is also likely to be a fruitful path for developing new and improved stochastic optimization techniques to be used in training deep neural networks. Finally, we end by calling for thorough case studies on real-world data, where the new deep learning models are compared to more traditional system identification approaches.

Cross-References

- [Nonlinear System Identification: An Overview of Common Approaches](#)
- [System Identification: An Overview](#)

Recommended Reading

There is by now a massive amount of literature of various aspects of deep learning, and it is almost impossible to get a grasp of it all. However, a good starting point is provided by the executive summary provided in LeCun et al. (2015). For a more thorough starting point, the textbook (Goodfellow et al. 2016) is a good option. The literature on deep learning for system identification is still rather scarce, but if we consider the more general topic of sequence learning, there are a lot of interesting developments available (see, e.g., Boulanger-Lewandowski et al. (2012), Osendorfer and Bayer (2014), Fabius et al. (2014), Chung et al. (2015), and Rangapuram et al. (2018)) to get started.

Bibliography

- Andersson C, Ribeiro AH, Tiels K, Wahlström N, Schön TB (2019) Deep convolutional networks are useful in system identification. In: Proceedings of the IEEE 58th IEEE Conference on Decision and Control (CDC), Nice
- Bai S, Kolter JZ, Koltun V (2018) An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. Technical report. arXiv:1803.01271
- Bottou L, Curtis FE, Nocedal J (2018) Optimization methods for large-scale machine learning. *SIAM Rev* 60(2):223–311
- Boulanger-Lewandowski N, Bengio Y, Vincent P (2012) Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription. In: Proceedings of the 29 th International Conference on Machine Learning (ICML), Edinburgh
- Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Belmont, Wadsworth
- Cho K, Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha
- Chung J, Kastner K, Dinh L, Goel K, Courville AC, Bengio Y (2015) A recurrent latent variable model for sequential data. In: Advances in Neural Information Processing Systems (NIPS)
- Fabius O, van Amersfoort JR, Kingma DP (2014) Variational recurrent auto-encoders. Technical report. arXiv:1412.6581
- Geman S, Bienenstock E, Doursat R (1992) Neural networks and the bias/variance dilemma. *Neural Comput* 4(1):1–58
- Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge/London
- Hinton G, Deng L, Yu D, Dahl GE, Mohamed A, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN, Kingsbury B (2012) Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process Mag* 29(6):82–97
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
- Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems NIPS, pp 1097–1105
- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521:436–444
- Ljung L (1999) System identification – theory for the user, 2nd edn. Prentice Hall, Upper Saddle River
- McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5(4):115–133
- Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. Technical report. arXiv:1301.3781
- Osendorfer C, Bayer J (2014) Learning stochastic recurrent networks. Technical report. arXiv:1411.7610
- Pascanu R, Gulcehre C, Cho K, Bengio Y (2014) How to construct deep recurrent neural networks. In: Proceedings of the 2nd International Conference on Learning Representations (ICLR)
- Rangapuram SS, Seeger MW, Gasthaus J, Stella L, Wang Y, Januschowski T (2018) Deep state space models for time series forecasting. In: Advances in Neural Information Processing Systems (NIPS), pp 7785–7794
- Robbins H, Monro S (1951) A stochastic approximation method. *Ann Math Stat* 22(3):400–407
- Schoukens J, Ljung L (2019) Nonlinear system identification – a user-oriented roadmap. *IEEE Control Syst Mag* 39(6):28–99. <https://10.1109/MCS.2019.2938121>, IEEE <https://ieeexplore.ieee.org/document/8897147>
- Schoukens J, Vaes M, Pintelon R (2016) Linear system identification in a nonlinear setting: nonparametric analysis of the nonlinear distortion and their impact on the best linear approximation. *IEEE Control Syst Mag* 36:38–69
- Williams RJ, Hinton GE, Rumelhart DE (1986) Learning representations by back-propagating errors. *Nature* 323:533–536

Demand Response: Coordination of Flexible Electric Loads

Johanna L. Mathieu

Department of Electrical Engineering and
Computer Science, University of Michigan,
Ann Arbor, MI, USA

Abstract

Demand-responsive electric loads respond to prices or other signals sent by the electric utility or power system operator by decreasing consumption (shedding) and/or shifting load in time. Demand response provides a variety of technical and economic benefits to the grid, usually causing minimal inconvenience to consumers. Traditionally, demand response has been used to decrease system-wide peak consumption. In that case, demand response events are called rarely and seek to achieve

large changes in load. Today, demand response is also used to provide a variety of other services to power systems. In some cases, loads are expected to respond continuously while still meeting the needs of consumers. In this article, we describe the types of loads well-suited to demand response and their response methods, typical demand response program designs and control methods, and emerging applications of demand response.

Keywords

Demand response · Electric loads · Peak load shedding · Ancillary services

Introduction

Many parts of the world have shifted to using electricity markets to transact electrical energy and reliability services, instead of letting monopolistic vertically integrated utilities decide how best to match demand with supply. However, these markets are usually one-sided in that electric power generation companies compete to sell power (supply), but electric load (demand) is nonresponsive to market forces, i.e., inelastic. Demand response is an attempt to make electric load elastic. It is defined by the US Federal Energy Regulatory Commission (FERC) as “changes in electric usage by demand-side resources from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized” (Federal Energy Regulatory Commission 2019). These changes include net reductions in energy usage through *shedding*, especially during times of system-wide peak consumption, or through energy-neutral *shifting*.

Demand response can reduce wholesale electricity market prices and their volatility, improve power quality and system reliability, defer the need for grid infrastructure investments, and support the integration of fluctuating

renewable energy resources such as wind and solar by shifting load to better match production. While various forms of demand response have been used since the creation of the first power networks (e.g., time-of-use electricity pricing, load curtailment), demand response programs are currently expanding in size and scope, aided by lower-cost sensing and communication systems that enable faster and more accurate responses that provide more value to the power grid.

Demand-Responsive Loads and Response Strategies

A variety of residential, commercial, and industrial/other loads can provide demand response.

Residential

Many utility companies offer demand response programs to residential customers with air conditioning or electric heating systems, which can be switched off or cycled less frequently during times of system-wide peak consumption. These loads can also be shifted in time through pre-cooling/preheating strategies. Other residential loads such as dishwashers, clothes washers and dryers, and electric vehicles need to consume a certain amount of energy to perform their function, but the exact timing of their power draws is flexible; these loads can be deferred. For example, electric vehicles can be charged in the middle of the night, instead of immediately upon plug-in.

Commercial

Commercial building heating, ventilation, and air conditioning (HVAC) systems can be controlled to provide demand response. Motegi et al. (2007) list ten strategies including adjusting room temperatures, limiting fan variable-frequency drives, and adjusting chilled water temperatures. These strategies are useful for load shedding, and some are useful for load shifting. Lighting levels can also be modulated for demand response; however, unlike heat which can be stored, light cannot, and so lighting control is only suitable for load shedding, not load shifting.

Industrial/Other

Industrial processes can be scheduled to leverage flexibility in the timing of electricity-consuming steps and used for demand response. For example, Mathieu et al. (2011) analyze the data from an industrial-scale bakery that schedules pan washing to avoid times of system-wide peak consumption. Some industrial processes, such as aluminum smelting, can be modulated to provide load shifting in response to real-time signals from the power system (Todd et al. 2009). Electric pumping in agricultural systems, drinking water transport systems, and water treatment systems can be shifted in time to provide demand response.

Demand Response Programs

There are a number of taxonomies classifying demand response programs, e.g., Albadi and El-Saadany (2008). Here we split programs into three types: price-based, incentive-based, and market-based.

Price-Based

Price-based programs illicit a response by sending time-varying electricity prices to consumers, encouraging them to reduce consumption when prices are high (and, possibly, increase consumption when prices are low). Time-of-use prices follow a schedule known to consumers, while dynamic prices change over time in response to electricity market conditions (or proxies for those conditions like outdoor air temperature, which affects system-wide load and wholesale prices). There are a large variety of dynamic pricing designs including peak prices, where prices surge a small number of days/times per year, and real-time prices directly tied to wholesale electricity market prices.

The *utility's problem* is to design prices to achieve system cost and reliability goals while considering the impact on consumers. In practice, electricity rates must be approved by regulatory bodies responsible for protecting consumers. The *consumer's problem* is to respond to electricity prices. If they are known in advance

(e.g., time-of-use prices), a consumer could solve a multi-period optimization problem to minimize their energy costs subject to constraints capturing the flexibility of their electric loads. Responding to real-time electricity prices is more difficult; in practice, consumers often apply threshold policies to switch on/off loads if prices go below/above certain thresholds.

Incentive-Based

Incentive-based programs illicit a response by providing incentives, usually payments or electricity bill reductions, for demand response program participation and/or actions. For example, some utilities offer demand response programs that pay consumers for shedding load during times of system-wide peak consumption. Load sheds are computed by comparing actual load to an estimate of what the load would have been without demand response, known as a *baseline*. A variety of methods exist for estimating baselines including averaging historical load profiles and regression models built from historical data.

Some incentive-based demand response programs require consumers to sign a *contract* with the utility or a third-party company offering services to the utility or the power market operator. The contract defines the demand response actions, constraints, and payments. For example, for a consumer participating in an air-conditioning cycling program, the action could be utility-controlled air conditioner cycling on up to 12 hot summer days/year, the constraint could be the ability of the consumer to override cycling when indoor temperatures deviate more than 3°C from normal, and the payment could be based on program participation and/or on the number of days the air conditioner is actually cycled.

The *utility's problem* is to design incentives (or contracts including incentives) to achieve system cost and reliability goals while considering the impact on consumers. The *consumer's problem* is to use their load flexibility to maximize their incentives (or to choose the contract that maximizes their incentives) while ensuring that their electricity needs are met.

Market-Based

Market-based programs are similar to price-based programs in that consumers respond to time-varying electricity prices; however, in market-based programs, consumer actions impact price formation. Specifically, consumers or third-party companies acting on behalf of groups of consumers provide demand curves specifying their load as a function of the electricity price. The market operator collects all of the demand curves from the loads and supply curves from the generation companies to clear the market and determine electricity prices. Consumers respond to prices according to their demand curves. While price-based programs are simpler, they may lead to load synchronization and price oscillations (Callaway and Hiskens 2011). Market-based programs can mitigate these issues, but not completely eliminate them.

The *market operator's problem* is to design the market to maximize social welfare. The *consumer's problem* is to design their demand curve to minimize their energy costs considering their electricity needs and the flexibility of their electric loads. Game theoretic approaches have been used to study this problem.

Emerging Applications

Responsive demand is now providing a variety of services to power systems, and researchers continue to investigate new demand response applications and load coordination architectures that are well beyond the scope of the FERC demand response definition. Here, we describe three emerging applications.

Load Coordination for Balancing Reserves

Power system operators procure balancing reserves to correct differences in electricity supply and demand on timeframes of seconds to minutes. Traditionally, balancing reserves are provided by fast-acting generators, like natural gas power plants, that operate below their maximum output level, enabling them to decrease and increase power production as load decreases and increases. Loads can also

supply balancing reserves by increasing and decreasing power consumption with respect to their baseline, ideally through load shifting. For example, the power consumption of thousands of residential air conditioning systems can be coordinated to provide frequency regulation (also known as secondary frequency control). A key challenge is to design a coordination scheme that produces accurate responses, is non-disruptive to customers, and is low cost in terms of both infrastructure and operational costs. The problem can be formulated as a real-time control problem, and researchers have investigated reduced-form dynamical system modeling, state estimation to reduce sensing/communication needs, and control design for large-scale, uncertain, nonlinear systems. Some strategies have been piloted and deployed in practice. This application is best achieved through incentive-based demand response programs with contracts.

Distribution Network Upgrade Deferral

Addition of more large loads such as electric vehicles and distributed generation such as solar photovoltaics to electricity distribution networks stresses the network and can lead to a variety of problems including low/high voltages and transformer overloading. Rather than upgrade these networks with new equipment, demand response can be used to mitigate these problems, deferring upgrades and reducing costs. For example, electric vehicles themselves could participate in demand response mitigating their own impacts, or other loads like residential air conditioning systems and commercial HVAC systems could support electric vehicles and/or solar photovoltaics. Price-based or incentive-based demand response programs could be used to achieve this application, though incentive-based programs with contracts would achieve more reliable responses and, subsequently, a more valuable service.

Maximizing Self-Consumption of Solar Power

Some countries use feed-in tariffs for residential solar photovoltaic generation. Specifically, consumers are paid for each kWh fed into the grid.

When the feed-in tariff is lower than the price of electricity, it encourages consumers to maximize self-consumption of the solar power they produce, which can be achieved by load shifting. No demand response program is necessary; the solar pricing policy drives demand response actions.

Summary and Future Directions

This article provided an introduction to demand response including an overview of demand-responsive loads, demand response program designs, and emerging applications. It highlighted various optimization and control problems corresponding to different program designs and applications. It also emphasized that demand response is much broader than peak load management. Responsive loads can provide a large variety of services to the electric power system, and some of these services may become particularly valuable in power systems with significant penetrations of fluctuating renewable energy resources.

Future directions include the development of additional demand response applications (services and value streams) and load coordination architectures that better balance technological, economic, social/behavioral goals and constraints.

Cross-References

- [Building Control Systems](#)
- [Building Energy Management System](#)
- [Coordination of Distributed Energy Resources for Provision of Ancillary Services: Architectures and Algorithms](#)

Recommended Reading

Callaway and Hiskens (2011) describes how highly distributed electric loads can be aggregated and used to provide services to power systems, including a discussion of control objectives and frameworks. (Borenstein et al. 2002) describes the economic underpinnings of

a responsive demand-side via dynamic electricity pricing. Cappers et al. (2010) analyzes empirical evidence from U.S. demand response programs.

Bibliography

- Albadi M, El-Saadany E (2008) A summary of demand response in electricity markets. *Electr Power Syst Res* 78:1989–1996
- Borenstein S, Jaske M, Rosenfeld A (2002) Dynamic pricing, advanced metering and demand response in electricity markets. Tech. Rep. CSEMWP-105, University of California Energy Institute: Center for the Study of Energy Markets
- Callaway D, Hiskens I (2011) Achieving controllability of electric loads. *Proc IEEE* 99(1):184–199
- Cappers P, Goldman C, Kathan D (2010) Demand response in U.S. electricity markets: Empirical evidence. *Energy* 35:1526–1535
- Federal Energy Regulatory Commission (2019) Reports on demand response & advanced metering. <https://www.ferc.gov/industries/electric/indus-act/demand-response/dem-res-adv-metering.asp>
- Mathieu J, Price P, Kiliccote S, Piette M (2011) Quantifying changes in building electricity use, with application to demand response. *IEEE Trans Smart Grid* 2(3): 507–518
- Motegi N, Piette M, Watson D, Kiliccote S, Xu P (2007) Introduction to commercial building control strategies and techniques for demand response. Tech. Rep. LBNL-59975, Lawrence Berkeley National Laboratory
- Todd D, Caufield M, Helms B, Starke M, Kirby B, Kueck J (2009) Providing reliability services through demand response: A preliminary evaluation of the demand response capabilities of alcoa inc. Tech. Rep. ORNL/TM-2008/233, Oak Ridge National Laboratory

Demand-Driven Automatic Control of Irrigation Channels

Michael Cantoni¹ and Iven Mareels^{1,2}

¹Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, VIC, Australia

²IBM Research Australia, Southbank, VIC, Australia

Abstract

Large-scale networks of reservoirs and open channels are widely used in agricultural set-

tings for distributing water to irrigators. A demand-driven approach to the automatic control of irrigation channels is described in this entry. The approach involves online measurement of water levels along the channel and distributed feedback control of flow regulation structures to manage the capacity to supply water off-takes under the power of gravity.

Keywords

Distributed/decentralized control ·
Large-scale systems · Water resource
management

Introduction

Humans have engineered water distribution systems for millennia; e.g., see Mays (2010) and Ortloff (2009). Large-scale networks of open channels continue to be used widely for redirecting water from the environment to support food production. It is noted in UNESCO (2019) that, while only around 20% of the world's cultivated land is currently irrigated, this accounts for more than 40% of global food production. Furthermore, it is noted in this report that the yield of certain crops can double or even triple when supplemented with irrigation. Approximately 70% of the water diverted from rivers and groundwater worldwide goes to irrigation, but less than 60% of this is then used productively (Schultz et al. 2005). Water is not lost, because it remains part of the freshwater cycle. However, any water taken for irrigation is no longer available for other uses, such as maintaining environmental flows or supporting other human activities.

It is estimated that irrigation inefficiency is equally split between conveyance network losses and on-farm inefficiencies, although this can vary significantly from region to region and year to year. Pre-modernization losses in Australian irrigation networks were around 30% of water released at the source on average. The largest fraction of this (at around half) was due to oversupply (Marsden Jacob Associates 2003). Similar or higher conveyance losses have been

reported for Spain (Playan and Mateos 2004), Cyprus, Jordan, Egypt (Lamaddalen et al. 2004), and India (Puram and Sewa Bhawan 2014). The opportunity for improvement in the operation of irrigation network infrastructure is evidently large. The expected increase in demand for precision water delivery, in flow rate and timing, is also a driver for better management. The significance of this relates to food production pressures as the global population grows and living standards rise in developed and developing countries (Davis and Hirji 2003). With the expansion of channel networks in response to increasing demand for irrigation, and to the impacts of climate change, the modernization of operational practices is also relevant in terms of providing value for the often large public investment involved. The economic case for water efficiency is made in Water Resources Group (2009).

In the absence of constraints on the sources of water for an irrigation channel network, it is natural to oversupply, since a potential penalty for shortfalls is crop failure. For a channel that is operated at maximum flow capacity, off-takes to farms can be (in principle) scheduled subject to the supply limit alone, without much regard for the channel dynamics. However, this limits the flexibility with which farmers can conduct their business. Further, deviations from the schedule can lead to irrecoverable outflows at the bottom of the channel network, resulting in poor distribution efficiency. By contrast, an on-demand operating regime requires regulation of water levels at the gravity-powered supply points to farms. While this provides farmers with the flexibility needed for improving the efficiency of on-farm operations, it cannot be readily achieved under infrequent manual adjustment of the flow regulation structures along a channel. Moreover, proper consideration of the channel dynamics becomes important.

Advances in sensor, actuator, and telecommunication technologies underpin the scope for automating the operation of irrigation channel networks. Online measurement of water levels at the supply points along a channel enables the use of feedback control methods to automatically

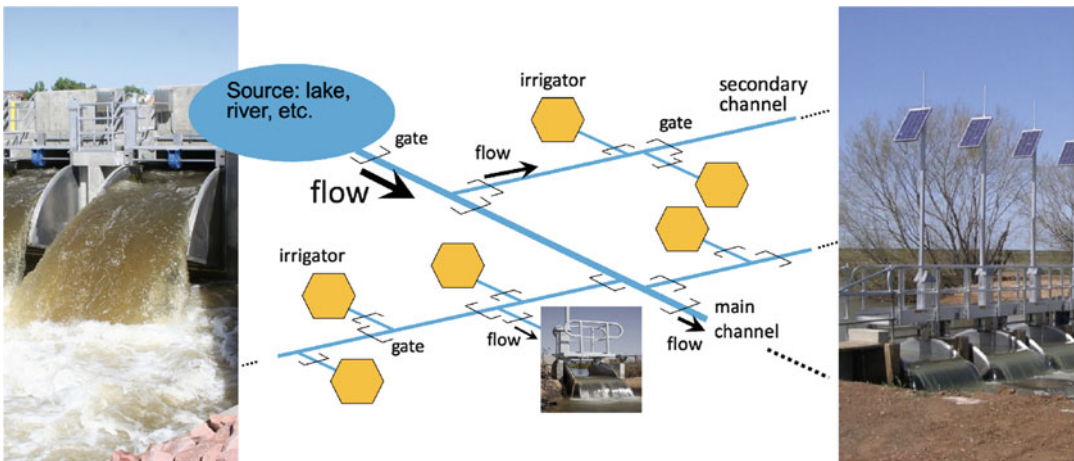
adjust flow regulation structures in response to varying demand for off-take flows. The purpose of this entry is to present a so-called distant-downstream feedback control architecture. This architecture results in demand-driven release of water from the source and, thus, zero or specified outflow at the end of the channel. The underlying modelling and distributed control design framework is discussed, along with limitations regarding the spatial propagation of transients in response to changes in flow demand. Perspectives derived from long-term operation of such controllers on irrigation channels in southeastern Australia are also provided. To conclude, some directions of ongoing research are summarized.

Distributed Distant-Downstream Control Architecture

A top view of an irrigation channel network is shown in Fig. 1. Such networks often span thousands of square kilometers. Each channel is a cascade of open water reaches called pools, with lengths that can range from hundreds to thousands of meters. These are separated by flow regulation structures called gates. Figure 2 illustrates the side view of a pool with overshoot gates; the use of undershot gates is also common in practice. Each gate can be used to set the corresponding flow downstream, provided suffi-

ciently frequent adjustments can be made to the gate position relative to the upstream water level. Supply points to farms tend to be situated at the downstream ends of the pools. As such, the gravity-fed capacity of a pool to convey flow further along the channel and to supply farms locally is determined by the corresponding downstream water level.

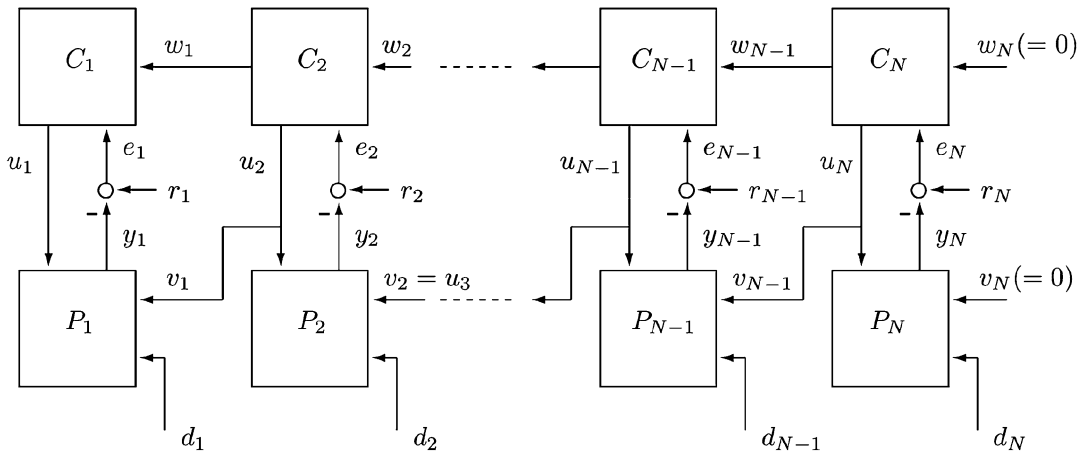
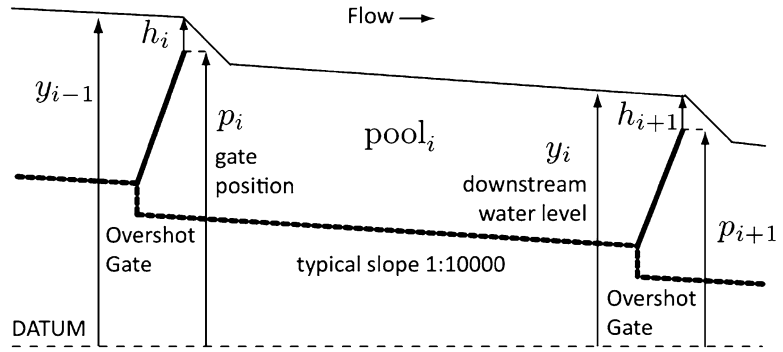
Regulation of the downstream water level y_i in pool_{*i*} can be achieved by feedback control (Malaterre 1995; Schuurmans et al. 1999a; Weyer 2002; Mareels et al. 2005). An instance of such a control system adjusts the upstream inflow $u_i = \gamma_i h_i^{3/2}$ by manipulating the gate position p_i to set the head over the gate $h_i = y_{i-1} - p_i$, where the constant γ_i is a discharge coefficient. This control action is made a function of the water level error $e_i = r_i - y_i$ for the given reference set point r_i . A standard approach is to use integral action in the controller to ensure zero steady-state water-level error in response to step changes of the local off-take and downstream flow load disturbance to the closed loop. It is of note that information about the instantaneous downstream flow load is known, since it is the result of control action. Thus, it can be fed forward (upstream) to improve system-wide performance to some extent, as discussed below in line with the investigations reported in Li et al. (2005) and Cantoni et al. (2007). The architecture of such a control system is illustrated



Demand-Driven Automatic Control of Irrigation Channels, Fig. 1 Top view of an irrigation channel network

Demand-Driven Automatic Control of Irrigation Channels,

Fig. 2 Side view of an irrigation channel pool. (Adapted from Cantoni et al. 2007)



Demand-Driven Automatic Control of Irrigation Channels, Fig. 3 Distributed distant-downstream control architecture. (Adapted from Cantoni et al. 2007)

in Fig. 3. Subsystem P_i represents the dynamics of pool_{*i*}, with inflow u_i and outflow load $v_i + d_i$, in which d_i is the local off-take and v_i is the downstream flow. The subsystem C_i denotes the corresponding component of the distributed controller.

The start and end of an off-take changes the net flow into the corresponding pool. The transient water level error caused by this results in local control action, in order to adjust the upstream flow. This in turn changes the net flow into the pool immediately upstream, which results in further control action there, until the inflow at the top of the channel is adjusted to ultimately match the change in load. In this way, the release of upstream flow is demand driven. As such, under the so-called distant-downstream control architecture just described, it is possible to ensure specified (e.g., zero) outflow

at the end of channel. By contrast, alternative control architectures that rely on adjustment of the downstream gate of a pool to regulate the downstream water level do not exhibit this property. It is of note that similarly structured control systems play a role in the management of sewer networks (Marinaki and Papageorgiou 2005).

An approach to the design of the components of the distant-downstream control architecture is discussed below. From a practical perspective, an important feature of the approach relates to the scalability of the design process and maintainability of the control system. In particular, design of the distributed controllers is conducted on a pool-by-pool basis, using only local model information. Moreover, the implementation of the control system requires only localized information exchange from the downstream gate to

the upstream gate of each pool. Limitations on achievable performance under this purely reactive regime are also discussed.

Modelling Channel Dynamics for Control Design

A physics-based approach to modelling the dynamics of a pool leads to the so-called St Venant equations (Chaudry 2007; Litrico and Fromion 2009), given by

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} + q_l = 0, \quad (1)$$

$$\begin{aligned} \frac{\partial Q}{\partial t} + \left(\frac{gA}{B} - \frac{Q^2}{A^2} \right) \frac{\partial A}{\partial x} + 2 \frac{Q}{A} \frac{\partial Q}{\partial x} \\ + gA(S_f - S_0) = 0, \end{aligned} \quad (2)$$

where $A(x, t)$ is the cross section of water at time $t \geq 0$ and position $x \in [0, L]$ along the stretch of length L ; the total flow through this cross section is $Q(x, t)$; the outflow $q_l(x, t)$ per unit length accounts for off-takes; $B(x)$ is the bottom width; $S_0(x)$ is the per unit length longitudinal bottom slope; $S_f = n^2 Q^2 / A^2 R^{4/3}$ represents the friction slope, in which n is Manning's constant and R is the hydraulic radius; and g is gravitational acceleration. These are a one-dimensional version of the Navier-Stokes equations, representing conservation mass and momentum along the length of the pool. Accounting for channel state-dependent seepage and evaporation further complicates the model. As a first-order approximation, these effects can be lumped into $q_l(x, t)$.

The boundary conditions for (1)–(2) are $Q(0, t)$ and $Q(L, t)$. These are determined by the hydraulic characteristics and the positioning of the flow regulation structures at the upstream and downstream end of the pool, which provide communication between the adjacent stretches of open water along the channel. For an overshoot gate in free flow (i.e., not submerged), the sharp crested weir model is appropriate (Bos 1989): $Q(L, t) = \gamma h(t)^{3/2}$, where $h(t) = y(t) - p(t)$ is the head over the gate, $y(t) = f(A(L, t))$ is the downstream water level at $x = L$

(i.e., immediately upstream of the gate), and $p(t)$ is the position of the gate at time $t \geq 0$. The constant γ is a discharge coefficient.

The St Venant equations have been used for control system design (de Halleux et al. 2003; Litrico et al. 2005). However, this partial differential equation (PDE) model is of limited value in practice because it can be difficult to calibrate on the basis of observable data. Furthermore, it reflects more than is required from the perspective of regulating the downstream water levels; indeed, the water level and flow are predicted along the entire length of each pool. For the purpose of feedback controller design, it has been found that a much simpler grey-box model is more suitable (Schuurmans et al. 1995, 1999b; Weyer 2001).

With reference to Fig. 2, the grey-box model proposed in Weyer (2001) takes the form

$$\begin{aligned} \pi_i \left(\frac{d}{dt} \right) y_i(t) = \gamma_i h_i^{3/2}(t - \tau_i) \\ - \gamma_{i+1} h_{i+1}^{3/2}(t) - d_i(t), \end{aligned} \quad (3)$$

where y_i is the downstream water level in pool_{*i*}, the terms $h_i = y_{i-1} - p_i$ and $h_{i+1} = y_i - p_{i+1}$ are the head over the upstream and downstream gate, respectively, γ_i and γ_{i+1} are constant discharge parameters, d_i models the local off-take load, and τ_i is the time delay associated with surface wave propagation along the length of the pool. For a pool of up to several kilometers in length, the polynomial $\pi_i(\cdot)$ can be taken to be of third order, to yield an integrator mode for mass balance and a lightly damped oscillatory mode to capture the dominant standing wave. The model (3) is much simpler than the traditional St Venant equations (1)–(2). Its parameters can be readily calibrated to observed data by standard system identification techniques. It has been seen to explain experimental data very well (Weyer 2001; Ooi et al. 2005). Moreover, the model (3) is more tractable in terms of designing feedback controllers to achieve water level regulation under variable and uncertain load conditions. In fact, provided the local control-loop bandwidth lies below the dominant standing wave frequency, as

required anyway to achieve closed-loop stability in the presence of the related time delay, the first-order linear model

$$\alpha_i \frac{d}{dt} y_i(t) = u_i(t - \tau_i) - (v_i(t) + d_i(t)) \quad (4)$$

is adequate for control design. Here, α_i relates to the pool surface area, $u_i(t) = \gamma_i h_i(t)^{3/2}$ is the upstream inflow, and $v_i(t) = \gamma_{i+1} h_{i+1}(t)^{3/2}$ is the downstream flow load at time $t \geq 0$.

Design and Limitations of Distributed Distant-Downstream Controllers

Taking the Laplace transform of (4) yields the frequency domain model

$$P_i : \quad Y_i(s) = \frac{1}{s \alpha_i} \left(\exp(-s\tau_i) U_i(s) - V_i(s) - D_i(s) \right), \quad (5)$$

where $V_i(s) = U_{i+1}(s)$ for $i \in \{1, \dots, N-1\}$, with the boundary condition $V_N(s) = \mathcal{L}[v_n](s)$ for the specified time-domain outflow v_n at the end of the channel; e.g., $v_n = 0$ for no outflow.

Given such a model, frequency-domain loop-shaping principles of the kind described in Doyle et al. (1992) and Åström and Murray (2010), for example, can be applied on a pool-by-pool basis, which is scalable and advantageous from an engineering maintenance perspective. With reference to Fig. 3, this classical approach is used in practice to design decentralized PI feedback compensators of the form

$$C_i : \quad U_i(s) = \frac{\kappa_i (1 + s\phi_i)}{s (1 + s\psi_i)} E_i(s), \quad (6)$$

where $E_i(s) = Y_i(s) - R_i(s)$ and $R_i(s) = \rho_i/s$ for the reference water level $\rho_i > 0$. The compensator parameters ϕ_i, ψ_i, κ_i are positive constants selected to achieve closed-loop stability and desirable transient performance in response changes in the load $V_i + D_i$ and to achieve robustness to high-frequency modelling uncertainty. Specifically, the compensator

parameters κ_i and ϕ_i are set to introduce phase lead at the desired loop gain bandwidth in line with transient performance specifications and to accommodate a level of uncertainty in the delay. Additional roll-off beyond this to ensure low gain at the dominant wave frequency is achieved by appropriate selection of the constant ψ_i . By the internal model principle, step load disturbances are rejected in steady state along the channel because of integral action in the controller (see the pole at $s = 0$). The feedback control system effectively measures the off-take load and sets the flows over upstream gates in order to match the supply to the demand.

It is important to note that, in addition to the local transient response to changes in load, the closed-loop system for the channel as a whole exhibits spatial dynamics. Specifically, transient responses to changes in off-take load propagate upstream. Control action taken in response to a change in load locally acts as disturbance on the upstream pool (Li et al. 2005; Cantoni et al. 2007). This disturbance is known and can be fed forward by using distributed compensators of the form

$$C_i : \quad U_i(s) = \frac{\kappa_i (1 + s\phi_i)}{s (1 + s\psi_i)} E_i(s) + F_i(s) W_i(s),$$

where $W_i(s) = U_{i+1}(s)$. In practice, a simple constant feedforward compensator $F_i(s) = \varphi$ with $\varphi \in (0, 1)$ is used. This can yield better spatial transient propagation characteristics relative to the case of $F_i(s) = 0$. However, the requirement of zero steady-state error for on/off type off-takes leads to fundamental limitations. In particular, string instability of the kind that arises in homogeneous vehicle platoons (e.g., see Seiler et al. 2004) can be observed in sections of channel comprising cascades of pools with similar dynamics (Li et al. 2005; Cantoni et al. 2007). Specifically, there can be amplification of the peak flows over the gates as the transient response propagates upstream toward the source of the channel. Alternative distributed distant-downstream compensator forms that sacrifice steady-state water level error to achieve string stable dynamics are considered in Soltanian and

Cantoni (2015). In practice, variation in the dynamics of adjacent pools tends to moderate string stability issues. Practical implementations also include anti-windup compensation for managing flow regulation structure limits; e.g., see Zaccarian et al. (2007) and Li and Cantoni (2007).

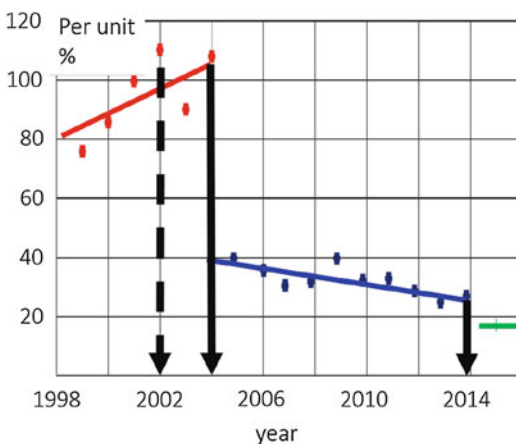
Results from Operational Channels

The focus of the Northern Victoria Irrigation Renewal Project (NVIRP 2009) was to achieve significant water efficiency gains in Australia's largest irrigated agricultural region. The project aim was to balance the needs of rural communities, to preserve food production and to improve agricultural sustainability and conservation of natural ecology. The on-demand system described above was implemented in the upgrade of the gravity-fed network in the Goulburn Murray Irrigation District (GMID). Similar projects are underway elsewhere in Australia and around the globe, including in the USA, China, and India.

The operational losses in one of the first Australian irrigation networks to be automated are displayed in Fig. 4 for the period 1998–2016, based on data from Hawke (2016). Losses

are shown in per-unit, with 100 representing pre-automation losses under normal allocation conditions; note that the values of less than 100 seen in 1999, 2000 and 2003 correspond to seasons in which the allocation of water for irrigation was less than usual. Before 2004, classical manual operation with SCADA support was employed across the district. The period 2002–2003 included the major upgrade of all regulating structures and off-take points in the channels and deployment of the distributed distant-downstream control system described above. This control system was made operational from the irrigation season 2004 onward. This operating regime is autonomous in the sense that, based on the online measurement of water levels across the network, the control system is entirely responsible for meeting water demand at the requested outlets. Automation resulted directly in a 50% reduction in water losses. Similar improvements in distribution efficiency are reported for the whole GMID in the recent audit report (Cardno 2017), which constitutes one of the largest deployments of the technology to date.

As operators became more familiar with the capacity of the on-demand system, and the data furnished by the system was used to better understand seepage losses, for example, small and steady improvements ensued over the decade 2004–14; see Fig. 4. This nudged annual losses to about 30% of pre-automation losses. Importantly, it was found that a small number of the main channels were responsible for the major seepage losses, which allowed management to direct the maintenance money to target these channels for lining and general upgrades. In 2014, the district board instigated a major data mining and optimization task force that exploited the last decade of automation data to find better scheduling guidelines and water set points for the entire system. It was further motivated to do this, because under automation the behavior of irrigators changed significantly as compared to the previous system. So the new norm of supply and demand conditions required a new operational irrigation regime for the controller to implement. This work resulted in a further 10% reduction in



Demand-Driven Automatic Control of Irrigation Channels, Fig. 4 Annually operational losses in per unit from 1998 to 2016 for an automated irrigation district in southeast Australia. (Adapted from Hawke 2016)

annual losses, bringing losses to about 16% of pre-automation days.

It has been found that the water regulating structures are moving on average every 10 min, across the entire season, compared to half-daily or even less frequent adjustment under manual operation. Moreover, in the automation regime, the behavior of the farmers with respect to using irrigation water has changed. The flexibility offered by water on demand allows planning in response to crop needs and the pursuit of better yields for their particular farm. Often larger flows over a shorter supply period are requested, which has in turn led to a redesign of onto-farm off-takes to cater for these conditions. The typical flow conditions on the channels are totally different from those observed before automation.

Summary and Future Directions

The technology and ideas described above have been patented (e.g., see Aughton et al. 2002a,b) and made commercially available through Rubicon Water (Pty Ltd). Automatic control of the kind described transforms gravity-fed open channel irrigation networks to enable near on-demand water distribution.

Demand-driven water management improves not only the conveyance efficiency, from reservoir to farm, but also farm water efficiency. Indeed it reduces the farmer's risk in crop management considerably. The ability to time water in response to crop needs improves crop yield and reduces fertilizer consumption. This improves the economic productivity of the farm, which is perhaps the ultimate driver for the implementation of channel network automation. The reduction in fertilizer requirements, combined with a reduction in on-farm water runoff, brings with it a welcome reduction in environmental stress.

The information infrastructure underpinning channel network automation also provides operators with a comprehensive set of data about water flows, water consumption, and water levels across the entire irrigation district. This allows for better management of the channel infrastructure and the available water resource. This data can form the

basis for how the limited available water resource can be optimally used (within the limitations of the infrastructure) to trade off water needs, incorporating rural, agricultural, urban, and ecological perspectives. This becomes a very worthwhile pursuit within the context of an economic market for water trading.

The demand-driven automatic control system described above regulates water levels to set points in the face of varying water demand. The water level set points are governed by the system operator. One can thus conceive supervisory control systems for determining set points that are suited to the operational mode at hand, as patented in Choy et al. (2013), for example. This can range from the use of short-term demand forecasts to improve the limitations of purely reactive feedback control at the lowest level, to methods for emptying channels (for maintenance), or filling channels (after maintenance or for ecological reasons), or passing water at maximum flow. It is but a small step, to envisage a supervisory control strategy that exploits the irrigation infrastructure for flood mitigation, spreading water over a larger area, storing a significant amount in channels, but more importantly routing water away to areas where excess water causes the least damage. Given the economic impact, such flood mitigation control strategies deserve much more attention.

Supervisory control problems of the kind just described can be addressed by using receding horizon optimal control techniques. For example, see van Overloop (2006). However, it transpires that model predictive control (MPC) can become computationally challenging when all of the physical constraints are incorporated for the channel network as a whole in a global formulation of such a scheme. Finding good structure exploiting methods for solving large-scale problems is an avenue for future research. Initial work in this direction is reported in Cantoni et al. (2017) and Zafar et al. (2019), for example. Alternative multi-agent approaches that seek to contain the size of the individual optimization problems to be solved, and the communication overhead required for cooperation, are considered in Negenborn et al. (2009) and Fele et al. (2014),

for example. Of course, there is also the challenge of managing the inevitable uncertainty in demand forecasts that may be available for use in the supervisory problem. See Nasir et al. (2019) for work on a stochastic approach to this issue.

In the end, the only reason water is required for irrigation is to grow crops. How to best manage the feedback loop from available water to crop production and the human lifestyle it supports is the real question that has to be addressed. This belongs to *circular economics* thinking and goes well beyond mere automation. It brings into focus new ways of farming, new crops, energy consumption, and population health issues. Nevertheless, automation will remain the key technology to pursue productivity and efficiency gains, as illustrated in the present context of improving gravity-fed open-channel water distribution.

Cross-References

- [Boundary Control of 1-D Hyperbolic Systems](#)
- [Classical Frequency-Domain Design Methods](#)
- [Control Structure Selection](#)
- [Distributed Sensing for Monitoring Water Distribution Systems](#)
- [Model Building for Control System Synthesis](#)
- [PID Control](#)
- [Vehicular Chains](#)

Bibliography

- 2030 Water Resources Group (2009) Charting our water future. Online: <https://www.2030wrg.org/charting-our-water-future/>
- Aström KJ, Murray R (2010) Feedback systems: an introduction for scientists and engineers. Princeton University Press, New Jersey
- Aughton D, Mareels I, Weyer E (2002a) WIPO Publication Number WO2002016698: CONTROL GATES. US Pat. 7,083,359 issued 1 Aug 2006
- Aughton D, Mareels I, Weyer E (2002b) WIPO Publication Number WO2002071163: FLUID REGULATION. US Pat. 7,152,001 issued 19 Dec 2006
- Bos M (1989) Discharge measurement structures, publication 20 International Institute for Land Reclamation and Improvement/ILRI, Wageningen, The Netherlands
- Cantoni M, Weyer E, Li Y, Ooi SK, Mareels I, Ryan M (2007) Control of large-scale irrigation networks. Proc IEEE 95(1):75–91
- Cantoni M, Farokhi F, Kerrigan E, Shames I (2017) Structured computation of optimal controls for constrained cascade systems. Int J Control. <https://doi.org/10.1080/00207179.2017.1366668>
- Cardno Report no. 3606-64 (2017) Audit of Irrigation Modernization Water Recovery 2016/17 Irrigation Season. Online: https://www.water.vic.gov.au/_data/assets/pdf_file/0022/112954/Audit-of-Irrigation-Modernisation-Water-Recovery-2017-v.2.0-FINAL.pdf
- Chaudhry M (2007) Open-channel flow. Springer Science & Business Media, New York
- Choy S, Cantoni M, Dower P, Kearney M (2013) WIPO Publication Number WO2013149304: SUPERVISORY CONTROL OF AUTOMATED IRRIGATION CHANNELS. US Pat. 9,952,601 issued 24 Apr 2018
- Davis R, Hirji R (2003) Irrigation and drainage development. Water Resources and Environment Technical Report E 1. The Worldbank, Washington, DC. Online: <http://documents.worldbank.org/curated/en/2003/03/9291611/>
- Doyle JC, Francis BA, Tannenbaum AR (1992) Feedback control theory. Macmillan, New York
- Fele F, Maestre JM, Hashemy SM, de la Pena DM, Camacho EF (2014) Coalitional model predictive control of an irrigation canal. J Process Control 24(4): 314–325
- de Halleux J, Prieur C, Coron JM, D’Andrea-Novell B, Bastin G (2003) Boundary feedback control in networks of open channels. Automatica 39(8): 1365–1376
- Hawke G (2016) Irrigation in Australia: unprecedented investment, innovation and insight – keynote address. In: 2016 irrigation Australia international conference and exhibition, Melbourne. Online: <https://www.irrigationaustralia.com.au/documents/item/320>
- Lamaddalena N, Lebdi F, Todorovic M, Bogliotti C (eds) (2004) Proceedings of the 2nd WASAMED (Water SAVING in MEDiterranean agriculture) Workshop (Irrigation Systems Performance). International Centre for Advanced Mediterranean Agronomic Studies. Online: https://www.um.edu.mt/_data/assets/pdf_file/0014/102614/WASAMED_options52.pdf
- Li Y, Cantoni M, Weyer E (2005) On water-level error propagation in controlled irrigation channels. In: Proceedings of the 44th IEEE conference on decision and control, pp 2101–2106
- Li Y, Cantoni M (2007) On distributed anti-windup compensation for distributed linear control systems. In: Proceedings of the 46th IEEE conference on decision and control, pp 1106–1111
- Litrico X, Fromion V (2009) Modeling and control of hydrosystems. Springer Science & Business Media, London
- Litrico X, Fromion V, Baume J-P, Arranja C, Rijo M (2005) Experimental validation of a methodology to control irrigation canals based on Saint Venant equations. Control Eng Pract 13(11):1425–1437
- Malaterre PO (1995) Regulation of irrigation canals. Irrig Drain Syst 9(4):297–327

- Marinaki M, Papageorgiou M (2005) Optimal real-time control of sewer networks. Springer Science & Business Media, London/New York
- Mareels I, Weyer E, Ooi SK, Cantoni M, Li Y, Nair G (2005) Systems engineering for irrigation systems: successes and challenges. *Annu Rev Control (IFAC)* 29(2):169–278
- Marsden Jacob Associates (2003) Improving water-use efficiency in irrigation conveyance systems A study of investment strategies. Land & Water Australia. ISBN 0642 760 993 – print. Online: <http://lwa.gov.au/products/pr030566>, <http://www.insidecotton.com/jspui/bitstream/1/1756/2/pr030516.pdf>
- Mays LW (ed) (2010) Ancient water technologies. Springer Science & Business Media, Dordrecht
- Nasir HA, Cantoni M, Li Y, Weyer E (2019) Stochastic model predictive control based reference planning for automated open-water channels. *IEEE Trans Control Syst Technol*. <https://doi.org/10.1109/TCST.2019.2952788>
- Negenborn RR, van Overloop PJ, Keviczky T, De Schutter B (2009) Distributed model predictive control of irrigation canals. *NHM* 4(2):359–380
- Northern Victorian Irrigation Renewal Project (2009). Online: https://www.parliament.vic.gov.au/images/stories/documents/council/SCFPA/water/Transcripts/NVIRP_Presentation.pdf
- Ooi SK, Krutzen MPM, Weyer E (2005) On physical and data driven modeling of irrigation channels. *Control Eng Pract* 13(4):461–471
- Ortloff CR (2009) Water engineering in the ancient world: archaeological and climate perspectives on societies of ancient South America, the middle east, and south-east Asia. Oxford University Press, Oxford, UK
- van Overloop PJ (2006) Model predictive control on open water systems. Ph.D. Thesis, Delft University of Technology, Delft
- Playan E, Mateos L (2004) Modernization and optimization of irrigation systems to increase water productivity. In: Proceedings of the 4th international crop science congress. Online: [http://www.cropscience.org.au/icsc2004/pdf/143_playane.pdf](http://www.cropsscience.org.au/icsc2004/pdf/143_playane.pdf)
- Puram RK, Sewa Bhawan S (2014) Guidelines for improving water use efficiency in irrigation, domestic & industrial sectors. Ministry of Water Resources, Government of India. Online: http://mowr.gov.in/sites/default/files/Guidelines_for_improving_water_use_efficiency_1.pdf
- Schultz B, Thatte CD, Labhsetwar VK (2005) Irrigation and drainage: main contributors to global food production. *Irrig Drain* 54(3):263–278
- Schuermans J, Bosgra OH, Brouwer R (1995) Open-channel flow model approximation for controller design. *Appl Math Model* 91:525–530
- Schuermans J, Hof A, Dijkstra S, Bosgra OH, Brouwer R (1999a) Simple water level controller for irrigation and drainage canals. *J Irrig Drain Eng* 125(4):189–195
- Schuermans J, Clemmens AJ, Dijkstra S, Hof A, Brouwer R (1999b) Modeling of irrigation and drainage canals for controller design. *J Irrig Drain Eng* 125(6):338–344
- Seiler P, Pant A, Hedrick K (2004) Disturbance propagation in vehicle strings. *IEEE Trans Autom Control* 49(10):1835–1841
- Soltanian L, Cantoni M (2015) Decentralized string-stability analysis for heterogeneous cascades subject to load-matching requirements. *Multidim Syst Signal Process* 26(4):985–999
- UNESCO (2019) The United Nations World Water Development Report 2019: Leaving no one behind – Executive Summary. Technical report, UNESCO. Online: <https://unesdoc.unesco.org/ark:/48223/pf0000367303>
- Weyer E (2002) Decentralised PI control of an open water channel. In: Proceedings of the 15th IFAC world congress
- Weyer E (2001) System identification of an open water channel. *Control Eng Pract* 9:1289–1299
- Zaccarian L, Li Y, Weyer E, Cantoni M, Teel AR (2007) Anti-windup for marginally stable plants and its application to open water channel control systems. *Control Eng Pract* 15(2):261–272
- Zafar A, Cantoni M, Farokhi F (2019) Optimal control computation for cascade systems by structured Jacobi iterations. *IFAC-PapersOnLine* 52(20):291–296

DES

- [Models for Discrete Event Systems: An Overview](#)

Descriptor System Techniques and Software Tools

Andreas Varga
Gilching, Germany

Abstract

The role of the descriptor system representation as a basis for reliable numerical computations for system analysis and synthesis, and in particular, for the manipulation of rational matrices, is discussed and available robust numerical software tools are described.

Keywords

CACSD · Modeling · Differential-algebraic systems · Rational matrices · Numerical analysis · Software tools

Introduction

A *linear time-invariant* (LTI) continuous-time descriptor system is described by the equations

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the input vector, $y(t) \in \mathbb{R}^p$ is the output vector, and $A, E \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$. The square matrix E is possibly singular, but we assume that the linear matrix pencil $A - \lambda E$, with λ a complex parameter, is regular (i.e., $\exists \lambda \in \mathbb{C}$ such that $\det(A - \lambda E) \neq 0$). A LTI discrete-time descriptor system has the form

$$\begin{aligned} Ex(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k) + Du(k). \end{aligned} \quad (2)$$

Descriptor system representations of the forms (1) and (2) are the most general descriptions of LTI systems. Standard LTI state-space systems correspond to the case $E = I_n$. We will alternatively denote the LTI descriptor systems (1) and (2) with the quadruples $(A - \lambda E, B, C, D)$ or (A, B, C, D) if $E = I_n$.

Continuous-time descriptor systems frequently arise when modeling interconnected systems involving linear differential equations and algebraic relations, and are also common in modeling constrained mechanical systems (e.g., contact problems). Discrete-time descriptor representations are encountered in the modeling of some economic processes. The descriptor system representation is instrumental in devising general computational procedures (even for standard LTI systems), whose intermediary steps involve operations leading to descriptor representations (e.g., system inversion or system conjugation in the discrete-time case).

The input-output behavior of the LTI systems (1) and (2) can be described in the form

$$\mathbf{y}(\lambda) = G(\lambda)\mathbf{u}(\lambda), \quad (3)$$

where $\mathbf{u}(\lambda)$ and $\mathbf{y}(\lambda)$ are the transformed input and output vectors, using, in the continuous-time

case, the Laplace transform with $\lambda = s$, and, in the discrete-time case, the Z-transform with $\lambda = z$, and where

$$G(\lambda) = C(A - \lambda E)^{-1}B + D \quad (4)$$

is the *transfer function matrix* (TFM) of the system. The TFM $G(\lambda)$ is a rational matrix having entries which are rational functions in the complex variable λ (i.e., ratios of two polynomials in λ). $G(\lambda)$ is *proper* (*strictly proper*) if each entry of $G(\lambda)$ has the degree of its denominator larger than or equal to (larger than) the degree of its numerator. If E is singular, then $G(\lambda)$ could have entries with the degrees of numerators exceeding the degrees of the corresponding denominators, in which case $G(\lambda)$ is *improper*. We will use the alternative notation

$$G(\lambda) := \left[\begin{array}{c|c} A - \lambda E & B \\ \hline C & D \end{array} \right], \quad (5)$$

to relate the TFM $G(\lambda)$ in (4) to a particular quadruple $(A - \lambda E, B, C, D)$.

An important application of LTI descriptor systems is to allow the numerically reliable manipulation of rational matrices (in particular, also of polynomial matrices). This is possible, because for any rational matrix $G(\lambda)$, there exists a quadruple $(A - \lambda E, B, C, D)$ with $A - \lambda E$ regular, such that (4) is fulfilled. Determining the matrices A, E, B, C , and D for a given rational matrix $G(\lambda)$ is known as the realization problem, and the quadruple $(A - \lambda E, B, C, D)$ is called a descriptor realization of $G(\lambda)$. The solution of the realization problem is not unique. For example, if U and V are invertible matrices of the same order as A , then two realizations $(A - \lambda E, B, C, D)$ and $(\tilde{A} - \lambda \tilde{E}, \tilde{B}, \tilde{C}, D)$ related as

$$\tilde{A} - \lambda \tilde{E} = U(A - \lambda E)V, \quad \tilde{B} = UB, \quad \tilde{C} = CV, \quad (6)$$

have the same TFM. The relations (6) define a (restricted) *similarity transformation* between the two descriptor system representations. Performing similarity transformations is a basic tool to manipulate descriptor system representations. An important aspect is the existence of minimal

realizations, which are descriptor realizations with the smallest possible state dimension n . The characterization of minimal descriptor realizations in terms of systems properties (i.e., controllability, observability, absence of non-dynamic modes) is done in Verghese et al. (1981). To simplify the presentation, we will assume in most of the cases discussed that the employed realizations of rational matrices are minimal.

Complex synthesis approaches of controllers and filters for plants modeled as LTI systems are often described as conceptual computational procedures in terms of input-output representations via TFMs. Since the manipulation of rational matrices is numerically not advisable because of the potential high sensitivity of polynomial based representations, it is generally accepted that the manipulation of rational matrices is best performed via their equivalent descriptor realizations. In what follows, we focus on discussing a selection of descriptor system techniques which are frequently encountered as computational blocks of the synthesis procedures. However, we refrain from discussing computational details, which can be found in the cited references. We conclude our presentation with a short overview of available software tools for descriptor systems.

Basics of Manipulating Rational Matrices

In this section, we present the basic manipulations of rational matrices, which represent the building blocks of more involved manipulations.

Basic Operations

We consider some operations involving a single TFM $G(\lambda)$ with the descriptor realization $(A - \lambda E, B, C, D)$. The *transposed* TFM $G^T(\lambda)$ corresponds to the *dual descriptor system* with the realization

$$G^T(\lambda) = \left[\begin{array}{c|c} A^T - \lambda E^T & C^T \\ \hline B^T & D^T \end{array} \right].$$

If $G(\lambda)$ is invertible, then an inversion free realization of the *inverse* TFM $G^{-1}(\lambda)$ is given by

$$G^{-1}(\lambda) = \left[\begin{array}{c|c} A - \lambda E & B \\ \hline C & D \\ \hline 0 & -I \end{array} \right].$$

This realization is not minimal, even if the original realization is minimal. However, if D is invertible, then an alternative realization of the inverse is

$$G^{-1}(\lambda) = \left[\begin{array}{c|c} A - BD^{-1}C - \lambda E & -BD^{-1} \\ \hline D^{-1}C & D^{-1} \end{array} \right],$$

which is minimal if the original realization is minimal. Notice that this operation may generally lead to an improper inverse even for standard state-space realizations (A, B, C, D) with singular D .

The *conjugate* (or *adjoint*) TFM $G^\sim(\lambda)$ is defined in the continuous-time case as $G^\sim(s) = G^T(-s)$ and has the realization

$$G^\sim(s) = \left[\begin{array}{c|c} -A^T - sE^T & C^T \\ \hline -B^T & D^T \end{array} \right],$$

while in the discrete-time case $G^\sim(z) = G^T(z^{-1})$ and has the realization

$$G^\sim(z) = \left[\begin{array}{c|c} E^T - zA^T & 0 \\ \hline zB^T & I \\ \hline 0 & I \end{array} \right].$$

This operation may generally lead to a conjugate system with improper $G^\sim(z)$, for a standard discrete-time state-space realization (A, B, C, D) with singular A .

Basic Couplings

Consider now two LTI systems with the rational TFMs $G_1(\lambda)$ and $G_2(\lambda)$, having the descriptor realizations $(A_1 - \lambda E_1, B_1, C_1, D_1)$ and $(A_2 - \lambda E_2, B_2, C_2, D_2)$, respectively. The product $G_1(\lambda)G_2(\lambda)$ represents the *series coupling* of the two systems and has the descriptor realization

$$G_1(\lambda)G_2(\lambda) := \left[\begin{array}{cc|c} A_1 - \lambda E_1 & B_1 C_2 & B_1 D_2 \\ 0 & A_2 - \lambda E_2 & B_2 \\ \hline C_1 & D_1 C_2 & D_1 D_2 \end{array} \right].$$

The *parallel coupling* corresponds to the sum $G_1(\lambda) + G_2(\lambda)$ and has the realization

$$G_1(\lambda) + G_2(\lambda) := \left[\begin{array}{cc|c} A_1 - \lambda E_1 & 0 & B_1 \\ 0 & A_2 - \lambda E_2 & B_2 \\ \hline C_1 & C_2 & D_1 + D_2 \end{array} \right].$$

The *column concatenation* of the two systems corresponds to building $\begin{bmatrix} G_1(\lambda) \\ G_2(\lambda) \end{bmatrix}$ and has the realization

$$\begin{bmatrix} G_1(\lambda) \\ G_2(\lambda) \end{bmatrix} = \left[\begin{array}{cc|c} A_1 - \lambda E_1 & 0 & B_1 \\ 0 & A_2 - \lambda E_2 & B_2 \\ \hline C_1 & 0 & D_1 \\ 0 & C_2 & D_2 \end{array} \right].$$

Similar realizations for the *row concatenation* and *diagonal stacking* of two systems result, using straightforward extensions of the standard systems case.

Canonical Forms of Linear Pencils

The main appeal of descriptor system techniques lies in their ability to address various analysis and synthesis problems of LTI systems in the most general setting, both from theoretical and computational standpoints. The basic mathematical ingredients for addressing analysis and synthesis problems of descriptor systems are two canonical forms of linear matrix pencils: the *Weierstrass canonical form* of a regular pencil and the *Kronecker canonical form* of a singular pencil. For a given a linear pencil $M - \lambda N$ (regular or singular), the corresponding canonical form $\tilde{M} - \lambda \tilde{N}$ can be obtained using a pencil similarity transformation of the form $\tilde{M} - \lambda \tilde{N} = U(M - \lambda N)V$, where U and V are suitable invertible matrices.

If the pencil $M - \lambda N$ is regular and $M, N \in \mathbb{C}^{n \times n}$, then, there exist invertible matrices $U \in \mathbb{C}^{n \times n}$ and $V \in \mathbb{C}^{n \times n}$ such that

$$U(M - \lambda N)V = \begin{bmatrix} J_f - \lambda I & 0 \\ 0 & I - \lambda J_\infty \end{bmatrix}, \quad (7)$$

where J_f is in a (complex) Jordan canonical form

$$J_f = \text{diag}(J_{s_1}(\lambda_1), J_{s_2}(\lambda_2), \dots, J_{s_k}(\lambda_k)), \quad (8)$$

with $J_{s_i}(\lambda_i)$ an elementary $s_i \times s_i$ Jordan block of the form

$$J_{s_i}(\lambda_i) = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}$$

and J_∞ is nilpotent and has the (nilpotent) Jordan form

$$J_\infty = \text{diag}(J_{s_1^\infty}(0), J_{s_2^\infty}(0), \dots, J_{s_h^\infty}(0)). \quad (9)$$

The Weierstrass canonical form (7) exhibits the finite and infinite eigenvalues of the pencil $M - \lambda N$. Overall, by including all multiplicities, there are $n_f = \sum_{i=1}^k s_i$ *finite eigenvalues* and $n_\infty = \sum_{i=1}^h s_i^\infty$ *infinite eigenvalues*. Infinite eigenvalues with $s_i^\infty = 1$ are called *simple infinite eigenvalues*. If M and N are real matrices, then there exist real matrices U and V such that the pencil $U(M - \lambda N)V$ is in a *real Weierstrass canonical form*, where the only difference is that J_f is in a real Jordan form. In this form, the elementary real Jordan blocks correspond to pairs of complex conjugate eigenvalues. If $N = I$, then all eigenvalues are finite and J_f in the Weierstrass form is simply the (real) Jordan form of M . The transformation matrices can be chosen such that $U = V^{-1}$.

If $M - \lambda N$ is an arbitrary (singular) pencil with $M, N \in \mathbb{C}^{m \times n}$, then, there exist invertible matrices $U \in \mathbb{C}^{m \times m}$ and $V \in \mathbb{C}^{n \times n}$ such that

$$U(M - \lambda N)V = \begin{bmatrix} K_r(\lambda) & & \\ & K_{reg}(\lambda) & \\ & & K_l(\lambda) \end{bmatrix}, \quad (10)$$

where:

- (1) The full row rank pencil $K_r(\lambda)$ has the form

$$K_r(\lambda) = \text{diag} (L_{\varepsilon_1}(\lambda), L_{\varepsilon_2}(\lambda), \dots, L_{\varepsilon_{v_r}}(\lambda)),$$

with $L_i(\lambda)$ ($i \geq 0$) an $i \times (i + 1)$ bidiagonal pencil of form

$$L_i(\lambda) = \begin{bmatrix} -\lambda & 1 & & \\ & \ddots & \ddots & \\ & & -\lambda & 1 \end{bmatrix}; \quad (11)$$

- (2) The regular pencil $K_{reg}(\lambda)$ is in a Weierstrass canonical form

$$K_{reg}(\lambda) = \begin{bmatrix} \tilde{J}_f - \lambda I & \\ & I - \lambda \tilde{J}_\infty \end{bmatrix},$$

with $\tilde{J}_f$ in a (complex) Jordan canonical form as in (8) and with $\tilde{J}_\infty$ in a nilpotent Jordan form as in (9);

- (3) The full column rank $K_l(\lambda)$ has the form

$$K_l(\lambda) = \text{diag} (L_{\eta_1}^T(\lambda), L_{\eta_2}^T(\lambda), \dots, L_{\eta_{v_l}}^T(\lambda)).$$

As it is apparent from (10), the Kronecker canonical form exhibits the right and left singular structures of the pencil $M - \lambda N$ via the full row rank block $K_r(\lambda)$ and full column rank block $K_l(\lambda)$, respectively, and the eigenvalue structure via the regular pencil $K_{reg}(\lambda)$. The full row rank pencil $K_r(\lambda)$ is $n_r \times (n_r + v_r)$, where $n_r = \sum_{i=1}^{v_r} \varepsilon_i$, the full column rank pencil $K_l(\lambda)$ is $(n_l + v_l) \times n_l$, where $n_l = \sum_{j=1}^{v_l} \eta_j$, while the regular pencil $K_{reg}(\lambda)$ is $n_{reg} \times n_{reg}$, with $n_{reg} = \tilde{n}_f + \tilde{n}_\infty$, where $\tilde{n}_f$ is the number of eigenvalues of J_f and $\tilde{n}_\infty$ is the number of infinite eigenvalues of $I - \lambda \tilde{J}_\infty$ (or equivalently the number of null eigenvalues of $\tilde{J}_\infty$). The normal rank r of the pencil $M - \lambda N$ results as

$$r := \text{rank}(M - \lambda N) = n_r + \tilde{n}_f + \tilde{n}_\infty + n_l.$$

If $M - \lambda N$ is regular, then there are no left- and right-Kronecker structures, and the Kronecker

canonical form is simply the Weierstrass canonical form.

The reduction of matrix pencils to the Weierstrass or Kronecker canonical forms generally involves the use of non-orthogonal, possibly ill-conditioned, transformation matrices. Therefore, the computation of these forms must be avoided when devising numerically reliable algorithms for descriptor systems. Alternative condensed forms, as the (ordered) generalized real Schur form of a regular pencil or various Kronecker-like forms of singular pencils, can be determined by using exclusively perfectly conditioned orthogonal transformations and can be always used instead of the Weierstrass or Kronecker canonical forms, respectively, in addressing the computational issues of descriptor systems.

Advanced Descriptor Techniques

In this section we discuss a selection of problems involving rational matrices, whose solutions involve the use of advanced descriptor system manipulation techniques. These techniques are instrumental in addressing controller and filter synthesis problems in the most general setting by using numerically reliable algorithms for the reduction of linear matrix pencils to appropriate condensed forms.

Normal Rank

The *normal rank* of a $p \times m$ rational matrix $G(\lambda)$, which we denote by $\text{rank } G(\lambda)$, is the maximal number of linearly independent rows (or columns) over the field of rational functions $\mathbb{R}(\lambda)$. It can be shown that the normal rank of $G(\lambda)$ is the maximally possible rank of the complex matrix $G(\lambda)$ for all values of $\lambda \in \mathbb{C}$ such that $G(\lambda)$ has finite norm. For the calculation of the normal rank r of $G(\lambda)$ in terms of its descriptor realization $(A - \lambda E, B, C, D)$, we use the relation

$$r = \text{rank } S(\lambda) - n,$$

where $\text{rank } S(\lambda)$ is the normal rank of the system matrix pencil $S(\lambda)$ defined as

$$S(\lambda) := \begin{bmatrix} A - \lambda E & B \\ C & D \end{bmatrix} \quad (12)$$

and n is the order of the descriptor state-space realization. The normal rank r can be easily determined from the Kronecker form of the pencil $S(\lambda)$ as

$$r := n_r + n_{reg} + n_l - n,$$

where n_r , n_{reg} , and n_l defines the normal ranks of $K_r(\lambda)$, $K_{reg}(\lambda)$, and $K_l(\lambda)$, respectively, in the Kronecker form (10) of $S(\lambda)$.

Poles and Zeros

The poles of $G(\lambda)$ are related to $\Lambda(A - \lambda E)$, the eigenvalues of the *pole pencil* $A - \lambda E$ (also known as the generalized eigenvalues of the pair (A, E)). For a minimal realization $(A - \lambda E, B, C, D)$ of $G(\lambda)$, the finite poles of $G(\lambda)$ are the $n_{p,f}$ finite eigenvalues in the Weierstrass canonical form of the regular pencil $A - \lambda E$, while the number of infinite poles is given by $n_{p,\infty} = \sum_{i=1}^h (s_i^\infty - 1)$, where s_i^∞ is the multiplicity of the i -th infinite eigenvalue. The infinite eigenvalues of multiplicity ones are the so-called non-dynamic modes. The McMillan degree of $G(\lambda)$, denoted by $\delta(G(\lambda))$, is the total number of poles $n_p := n_{p,f} + n_{p,\infty}$ of $G(\lambda)$

$$\delta(G(\lambda)) := n_p$$

and satisfies $\delta(G(\lambda)) \leq n$. A *proper* $G(\lambda)$ has only finite poles.

A proper $G(\lambda)$ is *stable* if all its poles belong to the appropriate stable region $\mathbb{C}_s \subset \mathbb{C}$, where $\mathbb{C}_s$ is the open left half plane of $\mathbb{C}$, for a continuous-time system, and the interior of the unit circle centered in the origin, for a discrete-time system. $G(\lambda)$ is *unstable* if it has at least one pole (finite or infinite) outside of the stability domain $\mathbb{C}_s$.

The zeros of $G(\lambda)$ are those complex values of λ (including also infinity), where the rank of the system matrix pencil (12) drops below its normal

rank $n + r$. Therefore, the zeros can be defined on the basis of the eigenvalues of the regular part $K_{reg}(\lambda)$ of the Kronecker form (10) of $S(\lambda)$. The finite zeros of $G(\lambda)$ are the $n_{z,f}$ finite eigenvalues of the regular pencil $K_{reg}(\lambda)$, while $n_{z,\infty}$, the total number of infinite zeros, is the sum of multiplicities of infinite zeros, which are defined by the multiplicities of infinite eigenvalues of $K_{reg}(\lambda)$ minus one. The total number of zeros is $n_z := n_{z,f} + n_{z,\infty}$. A proper and stable $G(z)$ is *minimum-phase* if all its zeros are finite and stable.

The number of poles and zeros of $G(\lambda)$ satisfy the relation

$$n_p = n_z + n_l + n_r,$$

where n_r and n_l are the normal ranks of $K_r(\lambda)$ and $K_l(\lambda)$, respectively, in the Kronecker form (10) of $S(\lambda)$.

Rational Nullspace Bases

Let $G(\lambda)$ be a $p \times m$ rational matrix of normal rank r and let $(A - \lambda E, B, C, D)$ be a minimal descriptor realization of $G(\lambda)$. The set of $1 \times p$ rational (row) vectors $\{v(\lambda)\}$ satisfying $v(\lambda)G(\lambda) = 0$ is a linear space, called the *left nullspace* of $G(\lambda)$, and has dimension $p - r$. Analogously, the set of $m \times 1$ rational (column) vectors $\{w(\lambda)\}$ satisfying $G(\lambda)w(\lambda) = 0$ is a linear space, called the *right nullspace* of $G(\lambda)$, and has dimension $m - r$.

The $p - r$ rows of a $(p - r) \times p$ rational matrix $N_l(\lambda)$ satisfying $N_l(\lambda)G(\lambda) = 0$ is a basis of the left nullspace of $G(\lambda)$, provided $N_l(\lambda)$ has full row rank. Analogously, the $m - r$ columns of a $m \times (m - r)$ rational matrix $N_r(\lambda)$ satisfying $G(\lambda)N_r(\lambda) = 0$ is a basis of the right nullspace of $G(\lambda)$, provided $N_r(\lambda)$ has full column rank. The determination of a rational left nullspace basis $N_l(\lambda)$ of $G(\lambda)$ can be easily turned into the problem of determining a rational basis of the system matrix $S(\lambda)$. Let $M_l(\lambda)$ be a suitable rational matrix such that

$$Y_l(\lambda) := [M_l(\lambda) \ N_l(\lambda)] \quad (13)$$

is a left nullspace basis of the associated system matrix pencil $S(\lambda)$ (12). Thus, to determine $N_l(\lambda)$ we can determine first $Y_l(\lambda)$, a left nullspace basis of $S(\lambda)$, and then $N_l(\lambda)$ results as

$$N_l(\lambda) = Y_l(\lambda) \begin{bmatrix} 0 \\ I_p \end{bmatrix}.$$

By duality, if $Y_r(\lambda)$ is a right nullspace basis of $S(\lambda)$, then a right nullspace basis of $G(\lambda)$ is given by

$$N_r(\lambda) = [0 \ I_m] Y_r(\lambda).$$

The Kronecker canonical form (10) of the system pencil $S(\lambda)$ in (12) allows to easily determine left and right nullspace bases of $G(\lambda)$. Let $\bar{S}(\lambda) = US(\lambda)V$ be the Kronecker canonical form (10) of $S(\lambda)$, where U and V are the respective left and right transformation matrices. If $\bar{Y}_l(\lambda)$ is a left nullspace basis of $\bar{S}(\lambda)$, then

$$N_l(\lambda) = \bar{Y}_l(\lambda)U \begin{bmatrix} 0 \\ I_p \end{bmatrix}. \quad (14)$$

Similarly, if $\bar{Y}_r(\lambda)$ is a right nullspace basis of $\bar{S}(\lambda)$ then

$$N_r(\lambda) = [0 \ I_m] V \bar{Y}_r(\lambda). \quad (15)$$

We choose $\bar{Y}_l(\lambda)$ of the form

$$\bar{Y}_l(\lambda) = [0 \ 0 \ \bar{Y}_{l,3}(\lambda)], \quad (16)$$

where $\bar{Y}_{l,3}(\lambda)$ satisfies $\bar{Y}_{l,3}(\lambda)K_l(\lambda) = 0$. Similarly, we choose $\bar{Y}_r(\lambda)$ of the form

$$\bar{Y}_r(\lambda) = \begin{bmatrix} \bar{Y}_{r,1}(\lambda) \\ 0 \\ 0 \end{bmatrix}, \quad (17)$$

where $\bar{Y}_{r,1}(\lambda)$ satisfies $K_r(\lambda)\bar{Y}_{r,1}(\lambda) = 0$. Both $\bar{Y}_{l,3}(\lambda)$ and $\bar{Y}_{r,1}(\lambda)$ can be determined as polynomial or rational matrices and the resulting bases are polynomial or rational as well.

Additive Decompositions

Let $G(\lambda)$ be a rational TFM with a descriptor system realization $G(\lambda) = (A - \lambda E, B, C, D)$.

Consider a disjunct partition of the complex plane $\mathbb{C}$ as

$$\mathbb{C} = \mathbb{C}_g \cup \mathbb{C}_b, \quad \mathbb{C}_g \cap \mathbb{C}_b = \emptyset, \quad (18)$$

where both $\mathbb{C}_g$ and $\mathbb{C}_b$ are symmetrically located with respect to the real axis, and $\mathbb{C}_g$ has at least one point on the real axis. Since $\mathbb{C}_g$ and $\mathbb{C}_b$ are disjoint, each pole of $G(\lambda)$ lies either in $\mathbb{C}_g$ or in $\mathbb{C}_b$. Using a similarity transformation of the form (6), we can determine an equivalent representation of $G(\lambda)$ with partitioned system matrices of the form

$$G(\lambda) = \left[\begin{array}{c|c} UAV - \lambda UEV & UB \\ \hline CV & D \end{array} \right] \quad (19)$$

$$= \left[\begin{array}{cc|c} A_g - \lambda E_g & 0 & B_g \\ 0 & A_b - \lambda E_b & B_b \\ \hline C_g & C_b & D \end{array} \right],$$

where $\Lambda(A_g - \lambda E_g) \subset \mathbb{C}_g$ and $\Lambda(A_b - \lambda E_b) \subset \mathbb{C}_b$. It follows that $G(\lambda)$ can be additively decomposed as

$$G(\lambda) = G_g(\lambda) + G_b(\lambda), \quad (20)$$

where

$$G_g(\lambda) = \left[\begin{array}{c|c} A_g - \lambda E_g & B_g \\ \hline C_g & D \end{array} \right],$$

$$G_b(\lambda) = \left[\begin{array}{c|c} A_b - \lambda E_b & B_b \\ \hline C_b & 0 \end{array} \right],$$

and $G_g(\lambda)$ has only poles in $\mathbb{C}_g$, while $G_b(\lambda)$ has only poles in $\mathbb{C}_b$. The spectral separation in (19) is automatically provided by the Weierstrass canonical form of the pole pencil $A - \lambda E$, where the diagonal Jordan blocks are suitably permuted to correspond to the desired eigenvalue splitting. This approach automatically leads to partial fraction expansions of $G_g(\lambda)$ and $G_b(\lambda)$.

Coprime Factorizations

Consider a disjunct partition (18) of the complex plane $\mathbb{C}$, where both $\mathbb{C}_g$ and $\mathbb{C}_b$ are symmetrically located with respect to the real axis, and such that

$\mathbb{C}_g$ has at least one point on the real axis. Any rational matrix $G(\lambda)$ can be expressed in a left fractional form

$$G(\lambda) = M^{-1}(\lambda)N(\lambda), \quad (21)$$

or in a right fractional form

$$G(\lambda) = N(\lambda)M^{-1}(\lambda), \quad (22)$$

where both the denominator factor $M(\lambda)$ and the numerator factor $N(\lambda)$ have only poles in $\mathbb{C}_g$. These fractional factorizations over a “good” domain of poles $\mathbb{C}_g$ are important in various observer, fault detection filter, or controller synthesis methods, because they allow to achieve the placement of all poles of a TFM $G(\lambda)$ in the domain $\mathbb{C}_g$ simply, by a premultiplication or postmultiplication of $G(\lambda)$ with a suitable $M(\lambda)$.

Of special interest are the so-called coprime factorizations, where the factors satisfy additional coprimeness conditions. A fractional representation of the form (21) is a *left coprime factorization* (LCF) of $G(\lambda)$ with respect to $\mathbb{C}_g$, if there exist $U(\lambda)$ and $V(\lambda)$ with poles only in $\mathbb{C}_g$ which satisfy the *Bezout identity*

$$M(\lambda)U(\lambda) + N(\lambda)V(\lambda) = I.$$

A fractional representation of the form (22) is a *right coprime factorization* (RCF) of $G(\lambda)$ with respect to $\mathbb{C}_g$, if there exist $U(\lambda)$ and $V(\lambda)$ with poles only in $\mathbb{C}_g$ which satisfy

$$U(\lambda)M(\lambda) + V(\lambda)N(\lambda) = I.$$

For the computation of a right coprime factorization of $G(\lambda)$ with a minimal descriptor realization $(A - \lambda E, B, C, D)$, it is sufficient to determine a state-feedback matrix F such that all finite eigenvalues in $\Lambda(A + BF - \lambda E)$ belong to $\mathbb{C}_g$, and all infinite eigenvalues in $\Lambda(A + BF - \lambda E)$ are simple. The descriptor realizations of the factors are given by

$$\begin{bmatrix} N(\lambda) \\ M(\lambda) \end{bmatrix} = \begin{bmatrix} A + BF - \lambda E & B \\ C + DF & D \\ F & I_m \end{bmatrix}.$$

Similarly, to determine a left coprime factorization, it is sufficient to determine an output-injection matrix K such that all finite eigenvalues in $\Lambda(A + KC - \lambda E)$ belong to $\mathbb{C}_g$ and all infinite eigenvalues in $\Lambda(A + KC - \lambda E)$ are simple. The descriptor realizations of the factors are given by

$$\begin{bmatrix} N(\lambda) & M(\lambda) \end{bmatrix} = \left[\begin{array}{c|cc} A + KC - \lambda E & B + KD & K \\ \hline C & D & I_p \end{array} \right].$$

An important class of coprime factorizations is the class of coprime factorizations with minimum-degree denominators. The McMillan degree of $G(\lambda)$ satisfies $\delta(G(\lambda)) = n_g + n_b$, where n_g and n_b are the number of poles of $G(\lambda)$ in $\mathbb{C}_g$ and $\mathbb{C}_b$, respectively. The denominator factor $M(\lambda)$ has the minimum-degree property if $\delta(M(\lambda)) = n_b$. Special classes of coprime factorizations, as the coprime factorizations with inner denominators or the normalized coprime factorizations, have important applications in solving various analysis and synthesis problems.

Linear Rational Matrix Equations

The solution of model-matching problems encountered in the synthesis of controllers or filters involves the solution of the linear rational matrix equation

$$G(\lambda)X(\lambda) = F(\lambda), \quad (23)$$

with $G(\lambda)$ a $p \times m$ rational matrix and $F(\lambda)$ a $p \times q$ rational matrix. This equation has a solution provided the compatibility condition

$$\text{rank } G(\lambda) = \text{rank}[G(\lambda) \ F(\lambda)] \quad (24)$$

is fulfilled. Assume $G(\lambda)$ and $F(\lambda)$ have descriptor realizations of the form

$$\begin{aligned} G(\lambda) &= \left[\begin{array}{c|c} A - \lambda E & B_G \\ \hline C & D_G \end{array} \right], \\ F(\lambda) &= \left[\begin{array}{c|c} A - \lambda E & B_F \\ \hline C & D_F \end{array} \right], \end{aligned}$$

which share the system pair $(A - \lambda E, C)$. It is easy to observe that any solution $X(\lambda)$ of (23) is also part of the solution $Y(\lambda) = \begin{bmatrix} W(\lambda) \\ X(\lambda) \end{bmatrix}$ of the linear polynomial equation

$$S_G(\lambda)Y(\lambda) = \begin{bmatrix} B_F \\ D_F \end{bmatrix}, \quad (25)$$

where $S_G(\lambda)$ is the associated system matrix pencil

$$S_G(\lambda) = \begin{bmatrix} A - \lambda E & B_G \\ C & D_G \end{bmatrix}. \quad (26)$$

Therefore, an alternative to solving (23) is to solve (25) for $Y(\lambda)$ instead and compute $X(\lambda)$ as

$$X(\lambda) = [0 \ I_p]Y(\lambda). \quad (27)$$

The compatibility condition (24) becomes

$$\begin{aligned} & \text{rank} \begin{bmatrix} A - \lambda E & B_G \\ C & D_G \end{bmatrix} \\ &= \text{rank} \begin{bmatrix} A - \lambda E & B_G & B_F \\ C & D_G & D_F \end{bmatrix}. \end{aligned}$$

If $G(\lambda)$ is invertible, a descriptor system realization of $X(\lambda)$ can be explicitly obtained as

$$X(\lambda) = \left[\begin{array}{cc|c} A - \lambda E & B_G & B_F \\ C & D_G & D_F \\ \hline 0 & -I_p & 0 \end{array} \right]. \quad (28)$$

The general solution of (23) can be expressed as

$$X(\lambda) = X_0(\lambda) + X_r(\lambda)Y(\lambda),$$

where $X_0(\lambda)$ is any particular solution of (23), $X_r(\lambda)$ is a rational basis matrix for the right nullspace of $G(\lambda)$, and $Y(\lambda)$ is an arbitrary rational matrix with suitable dimensions. General methods to determine both $X_0(\lambda)$ and $X_r(\lambda)$ can be devised by using the Kronecker canonical form of the associated system matrix pencil $S_G(\lambda)$ in (26). It is also possible to choose $Y(\lambda)$ to obtain special solutions, as for example, with least McMillan degree.

Software Tools

Several basic requirements are desirable when implementing robust software tools for numerical computations:

- employing exclusively numerically stable or numerically reliable algorithms;
- ensuring high computational efficiency;
- enforcing robustness against numerical exceptions (overflows, underflows) and poorly scaled data;
- ensuring ease of use, high portability, and high reusability.

The above requirements have been enforced in the development of high-performance linear algebra software libraries, such as BLAS (Dongarra et al. 1990), a collection of basic linear algebra subroutines, and LAPACK (Anderson et al. 1999), a comprehensive linear algebra package based on BLAS. These requirements have been also adopted to implement SLICOT (Benner et al. 1999; Van Huffel et al. 2004), a subroutine library for control theory, based primarily on BLAS and LAPACK. The general-purpose library LAPACK contains over 1300 subroutines and covers most of the basic linear algebra computations for solving systems of linear equations and eigenvalue problems. The release 4.5 of the specialized library SLICOT (<http://www.slicot.org/>) is a free software distributed under the GNU General Public License (GPL). The substantially enriched current release 5.6 is free for academic and non-commercial use and contains over 500 subroutines. SLICOT covers the basic computational problems for the analysis and design of linear control systems, such as linear system analysis and synthesis, filtering, identification, solution of matrix equations, model reduction, and system transformations. Of special interest is the comprehensive collection of routines for handling descriptor systems and for solving generalized linear matrix equations, as well as, the routines for computing various Kronecker-like forms. The subroutine libraries BLAS, LAPACK, and SLICOT have been originally implemented in the general-purpose

language Fortran 77 and, therefore, provide a high level of reusability, which allows their easy incorporation in user-friendly software environments, for example, MATLAB. In the case of MATLAB, selected LAPACK routines underlie the linear algebra functionalities, while the incorporation of selected SLICOT routines was possible via suitable gateways, as the provided *mex*-function interface.

The Control System Toolbox (CST) of MATLAB supports both descriptor system state-space models and input-output representations with improper rational TFMs and provides a rich functionality covering the basic system operations and couplings, model conversions, as well as some advanced functionality such as pole and zero computations, minimal realizations, and the solution of generalized Lyapunov and Riccati equations. However, most of the functions of the CST can only handle descriptor systems with proper TFMs and important functionality is currently lacking for handling the more general descriptor systems with improper TFMs, notably for determining the complete pole and zero structures or for computing minimal order realizations, to mention only a few of existing limitations.

To facilitate the implementation of the synthesis procedures of fault detection and isolation filters described in the book (Varga 2017), a new collection of freely available *m*-files, called the Descriptor System Tools (DSTOOLS), has been implemented for MATLAB. DSTOOLS is primarily intended to provide an extended functionality for both MATLAB (e.g., with matrix pencil manipulation methods for the computation of Kronecker-like forms) and for the CST by providing functions for minimal realization of descriptor systems, computation of pole and zero structures, computation of nullspace and range space bases, additive decompositions, several factorizations of rational matrices (e.g., coprime, full rank, inner-outer), evaluation of ν -gap distance, exact and approximate solution of linear rational matrix equations, eigenvalue assignment and stabilization via state feedback, etc. The approach used to develop DSTOOLS exploits MATLAB's matrix and object manipulation

features by means of a flexible and functionally rich collection of *m*-files, intended for noncritical computations, while simultaneously enforcing highly efficient and numerically sound computations via *mex*-functions (calling Fortran routines from SLICOT), to solve critical numerical problems requiring the use of structure-exploiting algorithms. An important aspect of implementing DSTOOLS was to ensure that standard systems are fully supported, using specific algorithms. In the same vein, all algorithms are available for both continuous- and discrete-time systems.

A precursor of DSTOOLS was the Descriptor Systems Toolbox for MATLAB, a proprietary software of the German Aerospace Center (DLR), developed between 1996 and 2006 (for the status of this toolbox around 2000, see Varga 2000). Some descriptor system functionality covering the basic manipulation of rational matrices is also available in the free, open-source software Scilab <http://scilab.org> and Octave <http://www.gnu.org/software/octave/>.

Cross-References

- [Basic Numerical Methods and Software for Computer Aided Control Systems Design](#)
- [Fault Detection and Diagnosis: Computational Issues and Tools](#)
- [Interactive Environments and Software Tools for CACSD](#)
- [Matrix Equations in Control](#)

Recommended Reading

The theoretical aspects of descriptor systems are discussed in the two textbooks (Dai 1989; Duan 2010). For a thorough treatment of rational matrices in a system theoretical context two authoritative references are Kailath (1980) and Vidyasagar (2011). The book (Varga 2017) illustrates the use of descriptor system techniques to solve the synthesis problems of fault detection and isolation filters in the most general setting. Chapters 9 and 10 of this book describe in details the

presented descriptor system techniques for the manipulation of rational matrices and also give details on available numerically reliable algorithms. These algorithms form the basis of the implementation of the functions available in the DSTOOLS collection. A comprehensive documentation of DSTOOLS is available in arXiv (Varga 2018). An extended version of this entry, available in arXiv (Varga 2019), provides bibliographical references on the best computational methods and describes additional descriptor system techniques.

Bibliography

- Anderson E, Bai Z, Bishop J, Demmel J, Du Croz J, Greenbaum A, Hammarling S, McKenney A, Ostrouchov S, Sorensen D (1999) LAPACK user's guide, 3rd edn. SIAM, Philadelphia
- Benner P, Mehrmann V, Sima V, Van Huffel S, Varga A (1999) SLICOT – a subroutine library in systems and control theory. In: Datta BN (ed) Applied and computational control, signals and circuits, vol 1. Birkhäuser, pp 499–539
- Dai L (1989) Singular control systems. Lecture notes in control and information sciences, vol 118. Springer, New York
- Dongarra JJ, Croz JD, Hammarling S, Duff I (1990) A set of level 3 basic linear algebra subprograms. ACM Trans Math Softw 16:1–17
- Duan GR (2010) Analysis and design of descriptor linear systems. Advances in mechanics and mathematics, vol 23. Springer, New York
- Kailath T (1980) Linear systems. Prentice Hall, Englewood Cliffs
- Van Huffel S, Sima V, Varga A, Hammarling S, Delebecque F (2004) High-performance numerical software for control. IEEE Control Syst Mag 24:60–76
- Varga A (2000) A descriptor systems toolbox for Matlab. In: Proceedings of the IEEE international symposium on computer-aided control system design, Anchorage, pp 150–155
- Varga A (2017) Solving fault diagnosis problems – linear synthesis techniques. Studies in systems, decision and control, vol 84. Springer International Publishing
- Varga A (2018) Descriptor system tools (DSTOOLS) V0.71, user's guide. <https://arxiv.org/abs/1707.07140>
- Varga A (2019) Descriptor system techniques and software tools. <https://arxiv.org/abs/1902.00009>
- Vergheze G, Lévy B, Kailath T (1981) A generalized state-space for singular systems. IEEE Trans Autom Control 26:811–831
- Vidyasagar M (2011) Control system synthesis: a factorization approach. Morgan & Claypool, San Rafael

Deterministic Description of Biochemical Networks

Jörg Stelling and Hans-Michael Kaltenbach
ETH Zürich, Basel, Switzerland

Abstract

Mathematical models of living systems are often based on formal representations of the underlying reaction networks. Here, we present the basic concepts for the deterministic nonspatial treatment of such networks. We describe the most prominent approaches for steady-state and dynamic analysis using systems of ordinary differential equations.

Keywords

Michaelis-Menten kinetics · Reaction networks · Stoichiometry

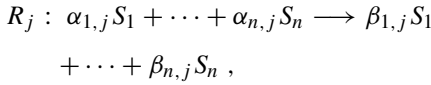
Introduction

A biochemical network describes the interconversion of biochemical species such as proteins or metabolites by chemical reactions. Such networks are ubiquitous in living cells, where they are involved in a variety of cellular functions such as conversion of metabolites into energy or building material of the cell, detection and processing of external and internal signals of nutrient availability or environmental stress, and regulation of genetic programs for development.

Reaction Networks

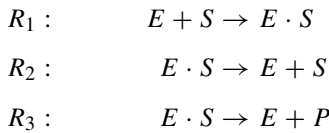
A biochemical network can be modeled as a dynamic system with the chemical concentration of each species taken as the states and dynamics described by the changes in species concentrations as they are converted by reactions. Assuming that species are homogeneously distributed in the reaction volume and that copy numbers are sufficiently high, we may ignore spatial and stochastic effects and derive a system of ordinary differential equations (ODEs) to model the dynamics.

Formally, a biochemical network is given by r reactions $R_1, \dots, R_r$ acting on n different chemical species $S_1, \dots, S_n$. Reaction R_j is given by



where $\alpha_{i,j}, \beta_{i,j} \in \mathbb{N}$ are called the *molecularities* of the species in the reaction. Their differences form the *stoichiometric matrix* $\mathbf{N} = (\beta_{i,j} - \alpha_{i,j})_{i=1\dots n, j=1\dots r}$, with $N_{i,j}$ describing the net effect of one turnover of R_j on the copy number of species S_i . The j th column is also called the *stoichiometry* of reaction R_j . The system can be opened to an un-modeled environment by introducing inflow reactions $\emptyset \rightarrow S_i$ and outflow reactions $S_i \rightarrow \emptyset$.

For example, consider the following reaction network:



Here, an enzyme E (a protein that acts as a catalyst for biochemical reactions) binds to a substrate species S , forming an intermediate complex $E \cdot S$ and subsequently converting S into a product P . Note that the enzyme-substrate binding is reversible, while the conversion to a product is irreversible. This network contains $r = 3$ reactions interconverting $n = 4$ chemical species $S_1 = S$, $S_2 = P$, $S_3 = E$, and $S_4 = E \cdot S$. The stoichiometric matrix is

$$\mathbf{N} = \begin{pmatrix} -1 & +1 & 0 \\ 0 & 0 & +1 \\ -1 & +1 & +1 \\ +1 & -1 & -1 \end{pmatrix}.$$

Dynamics

Let $\mathbf{x}(t) = (x_1(t), \dots, x_n(t))^T$ be the vector of concentrations, that is, $x_i(t)$ is the concentration of S_i at time t . Abbreviating this state vector as $\mathbf{x}$ by dropping the explicit dependence on time, its dynamics is governed by a system of n ordinary

differential equations:

$$\frac{d}{dt} \mathbf{x} = \mathbf{N} \cdot \mathbf{v}(\mathbf{x}, \mathbf{p}). \quad (1)$$

Here, the *reaction rate* vector $\mathbf{v}(\mathbf{x}, \mathbf{p}) = (v_1(\mathbf{x}, \mathbf{p}_1), \dots, v_r(\mathbf{x}, \mathbf{p}_r))^T$ gives the rate of conversion of each reaction per unit-time as a function of the current system state and of a set of parameters $\mathbf{p}$.

A typical reaction rate is given by the *mass-action rate law*

$$v_j(\mathbf{x}, \mathbf{p}_j) = k_j \cdot \prod_{i=1}^n x_i^{\alpha_{i,j}},$$

where the rate constant $k_j > 0$ is the only parameter and the rate is proportional to the concentration of each species participating as an educt (consumed component) in the respective reaction.

Equation (1) decomposes the system into a time-independent and linear part described solely by the topology and stoichiometry of the reaction network via $\mathbf{N}$ and a dynamic and typically nonlinear part given by the reaction rate laws $\mathbf{v}(\cdot, \cdot)$. One can define a directed graph of the network with one vertex per state and take $\mathbf{N}$ as the (weighted) incidence matrix. Reaction rates are then properties of the resulting edges. In essence, the equation describes the change of each species' concentration as the sum of the current reaction rates. Each rate is weighted by the molecularity of the species in the corresponding reaction; it is negative if the species is consumed by the reaction and positive if it is produced.

Using mass-action kinetics throughout, and using the species name instead of the numeric subscript, the reaction rates and parameters of the example network are given by

$$v_1(\mathbf{x}, \mathbf{p}_1) = k_1 \cdot x_E(t) \cdot x_S(t)$$

$$v_2(\mathbf{x}, \mathbf{p}_2) = k_2 \cdot x_{E \cdot S}(t)$$

$$v_3(\mathbf{x}, \mathbf{p}_3) = k_3 \cdot x_{E \cdot S}(t)$$

$$\mathbf{p} = (k_1, k_2, k_3)$$

The system equations are then

$$\begin{aligned}\frac{d}{dt}x_S &= -k_1 \cdot x_S \cdot x_E + k_2 \cdot x_{E \cdot S} \\ \frac{d}{dt}x_P &= k_3 \cdot x_{E \cdot S} \\ \frac{d}{dt}x_E &= -k_1 \cdot x_S \cdot x_E + k_2 \cdot x_{E \cdot S} + k_3 \cdot x_{E \cdot S} \\ \frac{d}{dt}x_{E \cdot S} &= k_1 \cdot x_S \cdot x_E - k_2 \cdot x_{E \cdot S} - k_3 \cdot x_{E \cdot S} .\end{aligned}$$

Steady-State Analysis

A reaction network is in *steady state* if the production and consumption of each species are balanced. Steady-state concentrations $\mathbf{x}^*$ then satisfy the equation

$$\mathbf{0} = \mathbf{N} \cdot \mathbf{v}(\mathbf{x}^*, \mathbf{p}) .$$

Computing steady-state concentrations requires explicit knowledge of reaction rates and their parameter values. For biochemical reaction networks, these are often very difficult to obtain. An alternative is the computation of *steady-state fluxes* $\mathbf{v}^*$, which only requires solving the system of homogeneous linear equations

$$\mathbf{0} = \mathbf{N} \cdot \mathbf{v} . \quad (2)$$

Lower and upper bounds v_i^l, v_i^u for each flux v_i can be given such that $v_i^l \leq v_i \leq v_i^u$; an example is an irreversible reaction R_i which implies $v_i^l = 0$. The set of all feasible solutions then forms a pointed, convex, polyhedral *flux cone* in $\mathbb{R}^r$. The rays spanning the flux cone correspond to *elementary flux modes (EFMs)* or *extreme pathways (EPs)*, minimal subnetworks that are already balanced. Each feasible steady-state flux can be written as a nonnegative combination

$$\mathbf{v}^* = \sum_i \lambda_i \cdot \mathbf{e}_i, \quad \lambda_i \geq 0$$

of EFMs $\mathbf{e}_1, \mathbf{e}_2, \dots$, where the λ_i are the corresponding weights.

Even if in steady state, living cells grow and divide. Growth of a cell is often described by

a combination of fluxes $\mathbf{b}^T \cdot \mathbf{v}$, the total production rate of relevant metabolites to form new biomass. The *biomass function* given by $\mathbf{b} \in \mathbb{R}^r$ is determined experimentally. The technique of *flux balance analysis (FBA)* then solves the linear program

$$\max_{\mathbf{v}} \mathbf{b}^T \cdot \mathbf{v}$$

subject to

$$\mathbf{0} = \mathbf{N} \cdot \mathbf{v}$$

$$v_i^l \leq v_i \leq v_i^u$$

to yield a feasible flux vector that balances the network while maximizing growth. Alternative objective functions have been proposed, for instance, for higher organisms that do not necessarily maximize the growth of each cell.

Quasi-Steady-State Analysis

In many reaction mechanisms, a *quasi-steady-state assumption (QSSA)* can be made, postulating that the concentration of some of the involved species does not change. This assumption is often justified if reaction rates differ hugely, leading to a time scale separation, or if some concentrations are very high, such that their change is negligible for the mechanism. A typical example is the derivation of *Michaelis-Menten kinetics*, which corresponds to our example network. There, we may assume that the concentration of the intermediate species $E \cdot S$ stays approximately constant on the time scale of the overall conversion of substrate into product and that the substrate, at least initially, is in much larger abundance than the enzyme. On the slower time scale, this leads to the *Michaelis-Menten rate law*:

$$v_P = \frac{d}{dt}x_P(t) = \frac{v_{\max} \cdot x_S(t)}{K_m + x_S(t)} ,$$

with a maximal rate $v_{\max} = k_3 \cdot x_E^{\text{tot}}$, where x_E^{tot} is the total amount of enzyme and the Michaelis-Menten constant $K_m = (k_2 + k_3)/k_1$ as a direct relation between substrate concentration and production rate. This approximation reduces the number of states by two. Both parameters of

the Michaelis-Menten rate law are also better suited for experimental determination: $v_{\max}$ is the highest rate achievable and K_m corresponds to the substrate concentration that yields a rate of $v_{\max}/2$.

Cooperativity and Ultra-sensitivity

In the Michaelis-Menten mechanism, the production rate gradually increases with increasing substrate concentration, until saturation (Fig. 1; $h = 1$). A different behavior is achieved if the enzyme has several binding sites for the substrate and these sites interact such that occupation of one site alters the affinity of the other sites positively or negatively, phenomena known as positive and negative cooperativity, respectively. With QSSA arguments as before, the fraction of enzymes completely occupied by substrate molecules at time t is given by

$$v = \frac{v_{\max} \cdot x_S(t)}{K^h + x_S^h(t)}$$

where $K > 0$ is a constant and $h > 0$ is the *Hill coefficient*. The Hill coefficient determines the shape of the response with increasing substrate concentration: a coefficient of $h > 1$ ($h < 1$) indicates positive (negative) cooperativity; $h = 1$ reduces to the Michaelis-Menten mechanism. With increasing coefficient h , the response changes from gradual to switch-like, such that the transition from low to high response becomes more rapid as indicated in Fig. 1. This phenomenon is also known as *ultra-sensitivity*.

Constrained Dynamics

Due to the particular structure of the system equation (1), the trajectories $\mathbf{x}(t)$ of the network with $\mathbf{x}_0 = \mathbf{x}(0)$ are confined to the *stoichiometric subspace*, the intersection of $\mathbf{x}_0 + \text{Im}\mathbf{N}$ with the positive orthant. *Conservation relations* that describe conservation of mass are thus found as solutions to

$$\mathbf{c}^T \cdot \mathbf{N} = \mathbf{0},$$

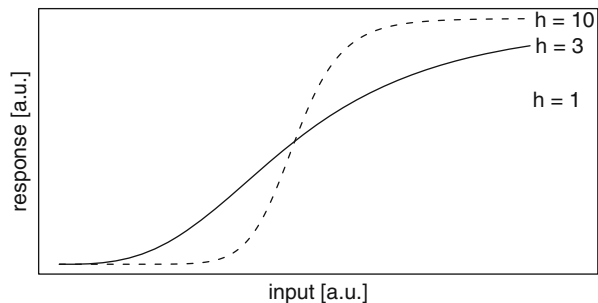
and two initial conditions $\mathbf{x}_0, \mathbf{x}'_0$ lead to the same stoichiometric subspace if $\mathbf{c}^T \cdot \mathbf{x}_0 = \mathbf{c}^T \cdot \mathbf{x}'_0$. This allows for the analysis of, for example, bistability using only the reaction network structure.

Summary and Future Directions

Reactions networks, even in simple cells, typically encompass thousands of components and reactions, resulting in potentially high-dimensional nonlinear dynamic systems. In contrast to engineered systems, biology is characterized by a high degree of uncertainty of both model structure and parameter values. Therefore, system identification is a central problem in this domain. Specifically, advanced methods for model topology and parameter identification as well as for uncertainty quantification need to be developed that take into account the very limited observability of biological systems. In addition, biological systems operate on multiple time, length, and concentration scales. For example, genetic regulation usually operates on the time scale of minutes and involves very few molecules, whereas metabolism is significantly faster and states are well approximated by species

Deterministic Description of Biochemical Networks,

Fig. 1 Responses for cooperative enzyme reaction with Hill coefficient $h = 1, 3, 10$, respectively. All other parameters are set to 1



concentrations. Corresponding systematic frameworks for multiscale modeling, however, are currently lacking.

Cross-References

- [Dynamic Graphs, Connectivity of](#)
- [Modeling of Dynamic Systems from First Principles](#)
- [Monotone Systems in Biology](#)
- [Robustness Analysis of Biological Models](#)
- [Spatial Description of Biochemical Networks](#)
- [Stochastic Description of Biochemical Networks](#)
- [Synthetic Biology](#)

Bibliography

- Craciun G, Tang Y, Feinberg M (2006) Understanding bistability in complex enzyme-driven reaction networks. *Proc Natl Acad Sci USA* 103(23):8697–8702
- Higham DJ (2008) Modeling and simulating chemical reactions. *SIAM Rev* 50(2):347–368
- LeDuc PR, Messner WC, Wikswo JP (2011) How do control-based approaches enter into biology? *Annu Rev Biomed Eng* 13:369–396
- Sontag E (2005) Molecular systems biology and control. *Eur J Control* 11(4–5):396–435
- Szallasi Z, Stelling J, Periwál V (eds) (2010) System modeling in cellular biology: from concepts to nuts and bolts. MIT, Cambridge
- Tyson JJ, Chen KC, Novak B (2003) Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. *Curr Opin Cell Biol* 15(2):221–231

Diagnosis of Discrete Event Systems

Stéphane Lafortune
Department of Electrical Engineering and
Computer Science, University of Michigan, Ann
Arbor, MI, USA

Abstract

We discuss the problem of event diagnosis in partially observed discrete event systems. The objective is to infer the past occurrence, if any,

of an unobservable event of interest based on the observed system behavior and the complete model of the system. Event diagnosis is performed by diagnosers that are synthesized from the system model and that observe the system behavior at run-time. Diagnosability analysis is the off-line task of determining which events of interest can be diagnosed at run-time by diagnosers.

Keywords

Diagnosability · Diagnoser · Fault diagnosis · Verifier

Introduction

In this entry, we consider discrete event systems that are partially observable and discuss the two related problems of event diagnosis and diagnosability analysis. Let the DES of interest be denoted by M with event set E . Since M is partially observable, its set of events E is the disjoint union of a set of *observable events*, denoted by E_o , with a set of *unobservable events*, denoted by E_{uo} : $E = E_o \cup E_{uo}$. At this point, we do not specify how M is represented; it could be an automaton or a Petri net. Let L_M be the set of all strings of events in E that the DES M can execute, i.e., the (untimed) language model of the system; cf. the related entries, ► [Models for Discrete Event Systems: An Overview](#), ► [Supervisory Control of Discrete-Event Systems](#), and ► [Modeling, Analysis, and Control with Petri Nets](#). The set E_{uo} captures the fact that the set of sensors attached to the DES M is limited and may not cover all the events of the system. Unobservable events can be internal system events that are not directly “seen” by the monitoring agent that observes the behavior of M for diagnosis purposes. They can also be fault events that are included in the system model but are not directly observable by a dedicated sensor. For the purpose of diagnosis, let us designate a specific unobservable event of interest and denote it by $d \in E_{uo}$. Event d could be a fault event or some other significant event that is unobservable.

Before we can state the problem of event diagnosis, we need to introduce some notation. E^* is the set of all strings of any length $n \in \mathbb{N}$ that can be formed by concatenating elements of E . The unique string of length $n = 0$ is denoted by ε and is the identity element of concatenation. As in article ► [Supervisory Control of Discrete-Event Systems](#), section “Supervisory Control Under Partial Observations,” we define the projection function $P : E^* \rightarrow E_o^*$ that “erases” the unobservable events in a string and replaces them by ε . The function P is naturally extended to a set of strings by applying it to each string in the set, resulting in a set of projected strings. The observed behavior of M is the language $P(L_M)$ over event set E_o .

The problem of *event diagnosis*, or simply *diagnosis*, is stated as follows: how to infer the past occurrence of event d when observing strings in $P(L_M)$ at run-time, i.e., during the operation of the system? This is model-based inferencing, i.e., the monitoring agent knows L_M and the partition $E = E_o \cup E_{uo}$, and it observes strings in $P(L_M)$. When there are multiple events of interest, d_1 to d_n , and these events are *fault* events, we have a problem of *fault diagnosis*. In this case, the objective is not only to determine that a fault has occurred (commonly referred to as “fault detection”) but also to identify which fault has occurred, namely, which event d_i (commonly referred to as “fault isolation and identification”). Fault diagnosis requires that L_M contains not only the nominal or fault-free behavior of the system but also its behavior after the occurrence of each fault event d_i of interest, i.e., its faulty behavior. Event d_i is typically a fault of a component that leads to degraded behavior on the part of the system. It is not a catastrophic failure that would cause the system to completely stop operating, as such a failure would be immediately observable. The decision on which fault events d_i , along with their associated faulty behaviors, to include in the complete model L_M is a design one that is based on practical considerations related to the diagnosis objectives.

A complementary problem to event diagnosis is that of *diagnosability analysis*. Diagnosability

analysis is the off-line task of determining, on the basis of L_M and of E_o and E_{uo} , if any and all occurrences of the given event of interest d will eventually be diagnosed by the monitoring agent that observes the system behavior.

Event diagnosis and diagnosability analysis arise in numerous applications of systems that are modeled as DES. We mention a few application areas where DES diagnosis theory has been employed. In heating, ventilation, and air-conditioning systems, components such as valves, pumps, and controllers can fail in degraded modes of operation, such as a valve gets stuck open or stuck closed or a pump or controller gets stuck on or stuck off. The available sensors may not directly observe these faults, as the sensing abilities are limited. Fault diagnosis techniques are essential, since the components of the system are often not easily accessible. In monitoring communication networks, faults of certain transmitters or receivers are not directly observable and must be inferred from the set of successful communications and the topology of the network. In document processing systems, faults of internal components can lead to jams in the paper path or a decrease in image quality, and while the paper jam or the image quality is itself observable, the underlying fault may not be as the number of internal sensors is limited.

Without loss of generality, we consider a single event of interest to diagnose, d . When there are multiple events to diagnose, the methodologies that we describe in the remaining of this entry can be applied to each event of interest d_i , $i = 1, \dots, n$, individually; in this case, the other events of interest d_j , $j \neq i$ are treated the same as the other unobservable events in the set E_{uo} in the process of model-based inferencing.

Problem Formulation

Event Diagnosis

We start with a general language-based formulation of the event diagnosis problem. The information available to the agent that monitors the system behavior and performs the task of event diagnosis is the language L_M and the set of

observable events E_o , along with the specific string $t \in P(L_M)$ that it observes at run-time. The actual string generated by the system is $s \in L_M$ where $P(s) = t$. However, as far as the monitoring agent is concerned, the actual string that has occurred could be any string in $P^{-1}(t) \cap L_M$, where P^{-1} is the inverse projection operation, i.e., $P^{-1}(t)$ is the set of all strings $s_t \in E^*$ such that $P(s_t) = t$. Let us denote this estimate set by $\mathcal{E}(t) = P^{-1}(t) \cap L_M$, where “ $\mathcal{E}$ ” stands for “estimate.” If a string $s \in L_M$ contains event d , we write that $d \in L_M$; otherwise, we write that $d \notin L_M$.

The event diagnosis problem is to synthesize a diagnostic engine that will automatically provide the following answers from the observed t and from the knowledge of L_M and E_o :

1. **Yes**, if and only if $d \in s$ for all $s \in \mathcal{E}(t)$.
2. **No**, if and only if $d \notin s$ for all $s \in \mathcal{E}(t)$.
3. **Maybe**, if and only if there exists $s_Y, s_N \in \mathcal{E}(t)$ such that $d \in s_Y$ and $d \notin s_N$.

As defined, $\mathcal{E}(t)$ is a string-based estimate. In section “[Diagnosis of Automata](#),” we discuss how to build a finite-state structure that will encode the desired answers for the above three cases when the DES M is modeled by a deterministic finite-state automaton. The resulting structure is called a *diagnoser automaton*.

Diagnosability Analysis

Diagnosability analysis consists in determining, a priori, if any and all occurrences of event d in L_M will eventually be diagnosed, in the sense that if event d occurs, then the diagnostic engine is guaranteed to eventually issue the decision “Yes.” For the sake of technical simplicity, we assume hereafter that L_M is a *live* language, i.e., any trace in L_M can always be extended by one more event. In this context, we would not want the diagnostic engine to issue the decision “Maybe” for an arbitrarily long number of event occurrences after event d occurs. When this outcome is possible, we say that event d is *not diagnosable* in language L_M .

The property of *diagnosability* of DES is defined as follows. In view of the liveness

assumption on language L_M , any string s'_Y that contains event d can always be extended to a longer string, meaning that it can be made “arbitrarily long” after the occurrence of d . That is, for any s'_Y in L_M and for any $n \in \mathbb{N}$, there exists $s_Y = s'_Y t \in L_M$ where the length of t is equal to n . Event d is *not* diagnosable in language L_M if there exists such a string s_Y together with a second string s_N that does not contain event d , and such that $P(s_Y) = P(s_N)$. This means that the monitoring agent is unable to distinguish between s_Y and s_N , yet, the number of events after an occurrence of d can be made arbitrarily large in s_Y , thereby preventing diagnosis of event d within a finite number of events after its occurrence. On the other hand, if no such pair of strings (s_Y, s_N) exists in L_M , then event d is diagnosable in L_M . (The mathematically precise definition of diagnosability is available in the literature cited at the end of this entry.)

Diagnosis of Automata

We recall the definition of a deterministic finite-state automaton, or simply automaton, from article ► [Models for Discrete Event Systems: An Overview](#), with the addition of a set of unobservable events as in section “Supervisory Control Under Partial Observations” in article ► [Supervisory Control of Discrete-Event Systems](#). The automaton, denoted by G , is a four-tuple $G = (X, E, f, x_0)$ where X is the finite set of states, E is the finite set of events partitioned into $E = E_o \cup E_{uo}$, x_0 is the initial state, and f is the deterministic partial transition function $f : X \times E \rightarrow X$ that is immediately extended to strings $f : X \times E^* \rightarrow X$. For a DES M represented by an automaton G , L_M is the language generated by automaton G , denoted by $L(G)$ and formally defined as the set of all strings for which the extended f is defined. It is an infinite set if the transition graph of G has one or more cycles. In view of the liveness assumption made on L_M in the preceding section, G has no reachable deadlocked state, i.e., for all $s \in E^*$ such that $f(x, s)$ is defined, then there exists $\sigma \in E$ such that $f(x, s\sigma)$ is also defined.

To synthesize a diagnoser automaton that correctly performs the diagnostic task formulated in the preceding section, we proceed as follows. First, we perform the parallel composition (denoted by $\parallel$) of G with the two-state *label automaton* A_{label} that is defined as follows. $A_{\text{label}} = (\{N, Y\}, \{d\}, f_{\text{label}}, N)$, where f_{label} has two transitions defined: (i) $f_{\text{label}}(N, d) = Y$ and (ii) $f_{\text{label}}(Y, d) = Y$. The purpose of A_{label} is to record the occurrence of event d , which causes a transition to state Y . By forming $G^{\text{labeled}} = G \parallel A_{\text{label}}$, we record in the states of G^{labeled} , which are of the form (x_G, x_A) , if the first element of the pair, state $x_G \in X$, was reached or not by executing event d at some point in the past: if d was executed, then $x_A = Y$, otherwise $x_A = N$. (We refer the reader to Chap. 2 in Cassandras and Lafortune (2008) for the formal definition of parallel composition of automata.) By construction, $L(G^{\text{labeled}}) = L(G)$.

The second step of the construction of the diagnoser automaton is to build the *observer* of G^{labeled} , denoted by $\text{Obs}(G^{\text{labeled}})$, with respect to the set of observable events E_o . (We refer the reader to Chap. 2 in Cassandras and Lafortune (2008) for the definition of the observer automaton and for its construction.) The construction of the observer involves the standard subset construction algorithm for nondeterministic automata in automata theory; here, the unobservable events are the source of nondeterminism, since they effectively correspond to ε -transitions. The diagnoser automaton of G with respect to E_o is defined as $\text{Diag}(G) = \text{Obs}(G \parallel A_{\text{label}})$. Its event set is E_o .

The states of $\text{Diag}(G)$ are *sets* of state pairs of the form (x_G, x_A) where x_A is either N or Y . Examination of the state of $\text{Diag}(G)$ reached by string $t \in P[L(G)]$ provides the answers to the event diagnosis problem. Let us denote that state by x_{Diag}^t . Then:

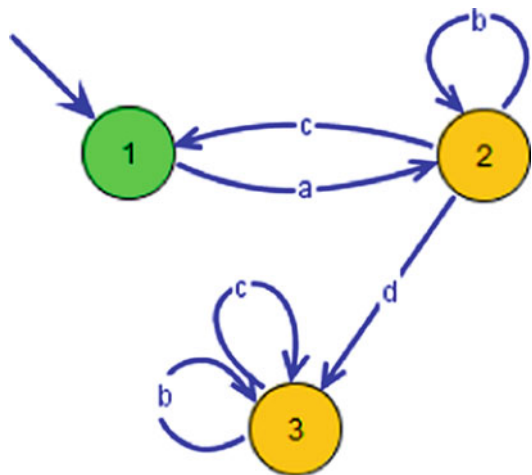
1. The diagnostic decision is **Yes** if *all* state pairs in x_{Diag}^t have their second component equal to Y ; we call such a state a “Yes-state” of $\text{Diag}(G)$.
2. The diagnostic decision is **No** if *all* state pairs in x_{Diag}^t have their second component equal

to N ; we call such a state a “No-state” of $\text{Diag}(G)$.

3. The diagnostic decision is **Maybe** if there is at least one state pair in x_{Diag}^t whose second component is equal to Y and at least one state pair in x_{Diag}^t whose second component is equal to N ; we call such a state a “Maybe-state” of $\text{Diag}(G)$.

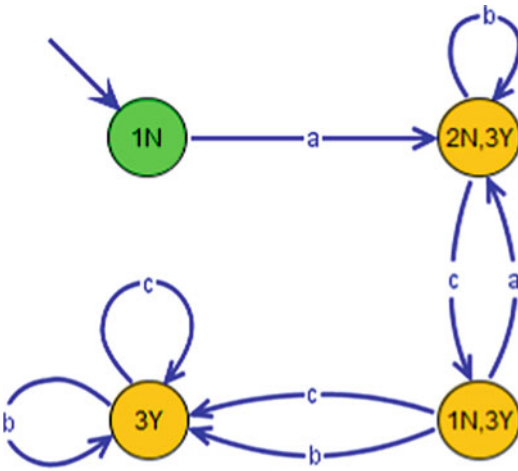
To perform run-time diagnosis, it therefore suffices to examine the current state of $\text{Diag}(G)$. Note that $\text{Diag}(G)$ can be computed off-line from G and stored in memory, so that run-time diagnosis requires only updating the new state of $\text{Diag}(G)$ on the basis of the most recent observed event (which is necessarily in E_o). If storing the entire structure of $\text{Diag}(G)$ is impractical, its current state can be computed on-the-fly on the basis of the most recent observed event and of the transition structure of G^{labeled} ; this involves one step of the subset construction algorithm.

As a simple example, consider the automaton G_1 shown in Fig. 1, where $E_{uo} = \{d\}$. The occurrence of event d changes the behavior of the system such that event c does not cause a return to the initial state 1 (identified by incoming arrow); instead, the system gets stuck in state 3 after d occurs.

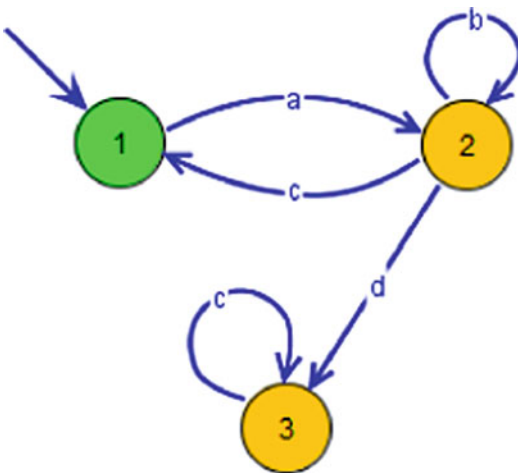


Diagnosis of Discrete Event Systems, Fig. 1
Automaton G_1

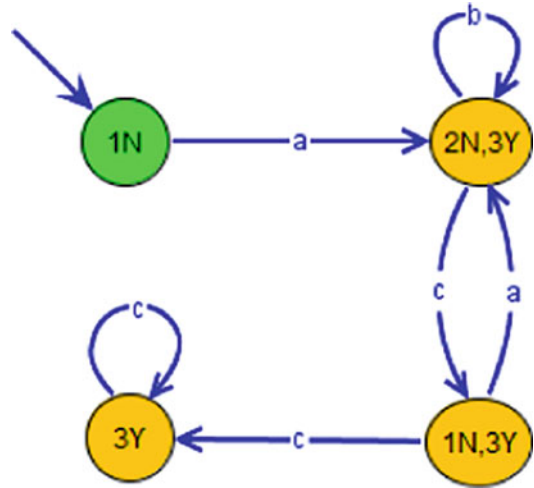
Its diagnoser is depicted in Fig. 2. It contains one Yes-state, state $\{(3, Y)\}$ (abbreviated as “3Y” in the figure), one No-state, and two Maybe-states (similarly abbreviated). Two consecutive occurrences of event c , or an occurrence of b right after c , both indicate that the system must be in state 3, i.e., that event d must have occurred; this is captured by the two transitions from Maybe-state $\{(1, N), (3, Y)\}$ to the Yes-state in $\text{Diag}(G_1)$. As a second example, consider the automaton G_2 shown in Fig. 3, where the self-loop b at state 3



Diagnosis of Discrete Event Systems, Fig. 2
Diagnoser automaton of G_1



Diagnosis of Discrete Event Systems, Fig. 3
Automaton G_2



Diagnosis of Discrete Event Systems, Fig. 4
Diagnoser automaton of G_2

in G_1 has been removed. Its diagnoser is shown in Fig. 4.

Diagnoser automata provide as much information as can be inferred, from the available observations and the automaton model of the system, regarding the past occurrence of unobservable event d . However, we may want to answer the question: Can the occurrence of event d always be diagnosed? This is the realm of diagnosability analysis discussed in the next section.

Diagnosability Analysis of Automata

As mentioned earlier, diagnosability analysis consists in determining, a priori, if any and all occurrences of event d in L_M will eventually be diagnosed. In the case of diagnoser automata, we do not want $\text{Diag}(G)$ to loop forever in a cycle of Maybe-states and never enter a Yes-state if event d has occurred, as happens in the diagnoser in Fig. 2 for the string adb^n when n gets arbitrarily large. In this case, $\text{Diag}(G_1)$ loops in Maybe-state $\{(2, N), (3, Y)\}$, and the occurrence of d goes undetected. This shows that event d is not diagnosable in G_1 ; the counterexample is provided by strings $s_Y = adb^n$ and $s_N = ab^n$.

For systems modeled as automata, diagnosability can be tested with quadratic time com-

plexity in the size of the state space of G by forming a so-called twin-automaton (also called “verifier”) where G is parallel composed with itself, but synchronization is only enforced on *observable* events, allowing arbitrary interleavings of unobservable events. The test for diagnosability reduces to detection of cycles that occur after event d in the twin-automaton. It can be verified that for automaton G_2 in our example, event d is diagnosable. Indeed, it is clear from the structure of G_2 that $\text{Diag}(G_2)$ in Fig. 4 will never loop in Maybe-state $\{(2, N), (3, Y)\}$ if event d occurs; rather, after d occurs, G_2 can only execute c events, and after two such events, $\text{Diag}(G_2)$ enters Yes-state $\{(3, Y)\}$.

Note that diagnosability will not hold in an automaton that contains a cycle of unobservable events after the occurrence of event d , although this is not the only instance where the property is violated, as we saw in our simple example.

Diagnosis and Diagnosability Analysis of Petri Nets

There are several approaches for diagnosis and diagnosability analysis of DES modeled by Petri nets, depending on the boundedness properties of the net and on what is observable about its behavior. Let N be the Petri net model of the system, which consists of a Petri net structure along with an initial marking of all places. If the transitions of N are labeled by events in a set E , some by observable events in E_o and some by unobservable events in E_{uo} , and if the contents of the Petri net places are not observed except for the initial marking of N , then we have a language-based diagnosis problem as considered so far in this entry, for language $L(N)$ and for $E = E_o \cup E_{uo}$, with event of interest $d \in E_{uo}$. In this case, if the set of reachable states of the net is bounded, then we can use the reachability graph as an equivalent automaton model of the same system and build a diagnoser automaton as described earlier. It is also possible to encode diagnoser states into the original structure of net N by keeping track of all possible net markings following the observation of an event in E_o , appending the appropriate label (“N” or “Y”)

to each marking in the state estimate. This is reminiscent of the on-the-fly construction of the current state of the diagnoser automaton discussed earlier, except that the possible system states are directly listed as Petri net markings on the structure of N . Regarding diagnosability analysis, it can be performed using the twin-automaton technique of the preceding section, from the reachability graph of N .

Another approach that is actively being pursued in current literature is to exploit the structure of the net model N for diagnosis and for diagnosability analysis, instead of working with the automaton model obtained from its reachability graph. In addition to potential computational gains from avoiding the explicit generation of the entire set of reachable states, this approach is motivated by the need to handle Petri nets whose sets of reachable states are infinite and in particular Petri nets that generate languages that are not regular and hence cannot be represented by finite-state automata. Moreover, in this approach, one can incorporate potential observability of token contents in the places of the Petri nets. We refer the interested reader to the relevant chapters in Campos et al. (2013) and Seatzu et al. (2013) for coverage of these topics.

Current and Future Directions

The basic methodologies described so far for diagnosis and diagnosability analysis have been extended in many different directions. We briefly discuss a few of these directions, which are active research areas. Detailed coverage of these topics is beyond the scope of this entry and is available in the references listed at the end.

Diagnosis of *timed models* of DES has been considered, for classes of timed automata and timed Petri nets, where the objective is to ensure that each occurrence of event d is detected within a bounded time delay. Diagnosis of *stochastic models* of DES has been considered, in particular stochastic automata, where the hard diagnosability constraints are relaxed and detection of each occurrence of event d must be guaranteed with some probability $1 - \epsilon$, for some small $\epsilon > 0$. Stochastic models also allow handling of

unreliable sensors or noisy environments where event observations may be corrupted with some probability, such as when an occurrence of event a is observed as event a 80 % of the time and as some other event a' 20 % of the time.

Decentralized diagnosis is concerned with DES that are observed by several monitoring agents $i = 1, \dots, n$, each with its own set of observable events $E_{o,i}$ and each having access to the entire set of system behaviors, L_M . The task is to design a set of individual diagnosers, one for each set $E_{o,i}$, such that the n diagnosers together diagnose all the occurrences of event d . In other words, for each occurrence of event d in any string of L_M , there exists at least one diagnoser that will detect it (i.e., answer “Yes”). The individual diagnosers may or may not communicate with each other at run-time or they may communicate with a coordinating diagnoser that will fuse their information; several decentralized diagnosis architectures have been studied and their properties characterized. The focus in these works is the decentralized nature of the information available about the strings in L_M , as captured by the individual observable event sets $E_{o,i}$, $i = 1, \dots, n$.

Distributed diagnosis is closely related to decentralized diagnosis, except that it normally refers to situations where each individual diagnoser uses only part of the entire system model. Let M be an automaton G obtained by parallel composition of subsystem models: $G = \parallel_{i=1,n} G_i$. In distributed diagnosis, one would want to design each individual diagnoser Diag_i on the basis of G_i alone or on the basis of G_i and of an *abstraction* of the rest of the system, $\parallel_{j=1,n; j \neq i} G_j$. Here, the emphasis is on the distributed nature of the system, as captured by the parallel composition operation. In the case where M is a Petri net N , the distributed nature of the system may be captured by individual net models N_i , $i = 1, \dots, n$, that are coupled by common places, i.e., *place-bordered* Petri nets.

Robust diagnosis generally refers to decentralized or distributed diagnosis, but where one or more of the individual diagnosers may fail. Thus, there must be built-in redundancy in the set of individual diagnosers so that they together may still detect every occurrence of event d even if one or more of them ceases to operate.

So far we have considered a fixed and static set of observable events, $E_o \subset E$, where every occurrence of each event in E_o is always observed by the monitoring agent. However, there are many instances where one would want the monitoring agent to dynamically activate or deactivate the observability properties of a subset of the events in E_o ; this arises in situations where event monitoring is “costly” in terms of energy, bandwidth, or security reasons. This is referred to as the case of *dynamic observations*, and the goal is to synthesize sensor activation policies that minimize a given cost function while preserving the diagnosability properties of the system.

Cross-References

- [Models for Discrete Event Systems: An Overview](#)
- [Modeling, Analysis, and Control with Petri Nets](#)
- [Supervisory Control of Discrete-Event Systems](#)
- [Modeling, Analysis, and Control with Petri Nets](#)

Recommended Reading

There is a very large amount of literature on diagnosis and diagnosability analysis of DES that has been published in control engineering, computer science, and artificial intelligence journals and conference proceedings. We mention a few recent books or survey articles that are a good starting point for readers interested in learning more about this active area of research. In the DES literature, the study of fault diagnosis and the formalization of diagnosability properties started in Lin (1994) and Sampath et al. (1995). Chapter 2 of the textbook Cassandras and Lafortune (2008) contains basic results about diagnoser automata and diagnosability analysis of DES, following the approach introduced in Sampath et al. (1995). The research monograph Lamperti and Zanella (2003) presents DES diagnostic methodologies developed in the artificial intelligence literature. The survey paper Zaytoon and Lafortune (2013) presents a detailed

overview of fault diagnosis research in the control engineering literature. The two edited books Campos et al. (2013) and Seatzu et al. (2013) contain chapters specifically devoted to diagnosis of automata and Petri nets, with an emphasis on automated manufacturing applications for the latter. Specifically, Chaps. 5, 14, 15, 17, and 19 in Campos et al. (2013) and Chaps. 22–25 in Seatzu et al. (2013) are recommended for further reading on several aspects of DES diagnosis. Zaytoon and Lafortune (2013) and the cited chapters in Campos et al. (2013) and Seatzu et al. (2013) contain extensive bibliographies.

Bibliography

- Campos J, Seatzu C, Xie X (eds) (2013) Formal methods in manufacturing. Series on industrial information technology. CRC/Taylor and Francis, Boca Raton, FL
- Cassandras CG, Lafortune S (2008) Introduction to discrete event systems, 2nd edn. Springer, New York, NY
- Lamperti G, Zanella M (2003) Diagnosis of active systems: principles and techniques. Kluwer, Dordrecht
- Lin F (1994) Diagnosability of discrete event systems and its applications. *Discret Event Dyn Syst Theory Appl* 4(2):197–212
- Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzi D (1995) Diagnosability of discrete event systems. *IEEE Trans Autom Control* 40(9):1555–1575
- Seatzu C, Silva M, van Schuppen J (eds) (2013) Control of discrete-event systems. Automata and Petri net perspectives. Lecture notes in control and information sciences, vol 433. Springer, London
- Zaytoon J, Lafortune S (2013) Overview of fault diagnosis methods for discrete event systems. *Annu Rev Control* 37(2):308–320

Differential Geometric Methods in Nonlinear Control

Arthur J. Krener
Department of Applied Mathematics, Naval
Postgraduate School, Monterey, CA, USA

Abstract

In the early 1970s, concepts from differential geometry were introduced to study nonlinear

control systems. The leading researchers in this effort were Roger Brockett, Robert Hermann, Henry Hermes, Alberto Isidori, Velimir Jurdjevic, Arthur Krener, Claude Lobry, and Hector Sussmann. These concepts revolutionized our knowledge of the analytic properties of control systems, e.g., controllability, observability, minimality, and decoupling. With these concepts, a theory of nonlinear control systems emerged that generalized the linear theory. This theory of nonlinear systems is largely parallel to the linear theory, but of course it is considerably more complicated.

Keywords

Codistribution · Distribution · Frobenius theorem · Involutive distribution · Lie jet

Introduction

This is a brief survey of the influence of differential geometric concepts on the development of nonlinear systems theory. Section “[A Primer on Differential Geometry](#)” reviews some concepts and theorems of differential geometry. Nonlinear controllability and nonlinear observability are discussed in sections “[Controllability of Nonlinear Systems](#)” and “[Observability for Nonlinear Systems](#)”. Section “[Minimal Realizations](#)” discusses minimal realizations of nonlinear systems, and section “[Disturbance Decoupling](#)” discusses the disturbance decoupling problem.

A Primer on Differential Geometry

Perhaps a better title might be “A Primer on Differential Topology” since we will not treat Riemannian or other metrics. A n -dimensional manifold $\mathcal{M}$ is a topological space that is locally homeomorphic to a subset of $\mathbb{R}^n$. For simplicity, we shall restrict our attention to smooth (C^∞) manifolds and smooth objects on them. Around each point $p \in \mathcal{M}$, there is at least one coordinate chart that is a neighborhood $\mathcal{N}_p \subset \mathcal{M}$ and a

homeomorphism $x : \mathcal{N}_p \rightarrow \mathcal{U}$ where $\mathcal{U}$ is an open subset of $\mathbb{R}^n$. When two coordinate charts overlap, the change of coordinates should be smooth. For simplicity, we restrict our attention to differential geometric objects described in local coordinates.

In local coordinates, a vector field is just an ODE of the form

$$\dot{x} = f(x) \quad (1)$$

where $f(x)$ is a smooth $\mathbb{R}^{n \times 1}$ valued function of x . In a different coordinate chart with local coordinates z , this vector field would be represented by a different formula:

$$\dot{z} = g(z)$$

If the charts overlap, then on the overlap they are related:

$$f(x(z)) = \frac{\partial x}{\partial z}(z)g(z), \quad g(z(x)) = \frac{\partial z}{\partial x}(x)f(x)$$

Since $f(x)$ is smooth, it generates a smooth flow $\phi(t, x^0)$ where for each t , the mapping $x \mapsto \phi(t, x^0)$ is a local diffeomorphism and for each x^0 the mapping $t \mapsto \phi(t, x^0)$ is a solution of the ODE (1) satisfying the initial condition $\phi(0, x^0) = x^0$. We assume that all the flows are complete, i.e., defined for all $t \in \mathbb{R}$, $x \in \mathcal{M}$. The flows are one parameter groups, i.e., $\phi(t, \phi(s, x^0)) = \phi(t+s, x^0) = \phi(s, \phi(t, x^0))$.

If $f(x^0) = b$, a constant vector, then locally the flow looks like translation, $\phi(t, x^1) = x^1 + tb$. If $f(x^0) \neq 0$, then we can always choose local coordinates z so that in these coordinates the vector field is constant. Without loss of generality, we can assume that $x^0 = 0$ and that the first component of f is $f_1(0) \neq 0$. Define the local change of coordinates: $x(z) = \phi(z_1, x^1(z))$ where $x^1(z) = (0, z_2, \dots, z_n)'$. It is not hard to see that this is a local diffeomorphism and that, in z coordinates, the vector field is the first unit vector.

If $f(x^0) = 0$ let $F = \frac{\partial f}{\partial x}(x^0)$, then if all the eigenvalues of F are off the imaginary axis, then the integral curves of $f(x)$ and

$$\dot{z} = Fz \quad (2)$$

are locally topologically equivalent (Arnol'd 1983). This is the Grobman-Hartman theorem. There exists a local homeomorphism $z = h(x)$ that carries $x(t)$ trajectories into $z(s)$ trajectories in some neighborhood of $x^0 = 0$. This homeomorphism need not preserve time $t \neq s$, but it does preserve the direction of time. Whether these flows are locally diffeomorphic is a more difficult question that was explored by Poincaré. See the section on feedback linearization (Krener 2013).

If all the eigenvalues of F are in the open left half plane, then the linear dynamics (2) is globally asymptotically stable around $z^0 = 0$, i.e., if the flow of (2) is $\psi(t, z)$, then $\psi(t, z^1) \rightarrow 0$ as $t \rightarrow \infty$. Then it can be shown that the nonlinear dynamics is locally asymptotically stable, $\phi(t, x^1) \rightarrow x^0$ as $t \rightarrow \infty$ for all x^1 in some neighborhood of x^0 .

One forms, $\omega(x)$, are dual to vector fields. The simplest example of a one form (also called a covector field) is the differential $dh(x)$ of a scalar-valued smooth function $h(x)$. This is the $\mathbb{R}^{1 \times n}$ covector field

$$\omega(x) = \left[\frac{\partial h}{\partial x_1}(x) \dots \frac{\partial h}{\partial x_n}(x) \right]$$

Sometimes this is written as

$$\omega(x) = \sum_1^n \frac{\partial h}{\partial x_i}(x) dx_i$$

The most general smooth one form is of the form

$$\begin{aligned} \omega(x) &= [\omega^1(x) \dots \omega^n(x)] \\ &= \sum_{i=1}^n \omega^i(x) dx_i \end{aligned}$$

where the $\omega^i(x)$ are smooth functions. The duality between one forms and vector fields is the bilinear pairing

$$\begin{aligned} \langle \omega(x), f(x) \rangle &= \omega(f)(x) = \omega(x)f(x) \\ &= \sum_{i=1}^n \omega^i(x) f_i(x) \end{aligned}$$

Just as a vector field can be thought of as a first-order ODE, a one form can be thought of as a first-order PDE. Given $\omega(x)$, find $h(x)$ such that $dh(x) = \omega(x)$. A one form $\omega(x)$ is said to be exact if there exists such an $h(x)$. Of course if there is one solution, then there are many all differing by a constant of integration which we can take as the value of h at some point x^0 .

Unlike smooth first-order ODEs, smooth first-order PDEs do not always have a solution. There are integrability conditions and topological conditions that must be satisfied. Suppose $dh(x) = \omega(x)$, then $\frac{\partial h}{\partial x_i}(x) = \omega^i(x)$ so

$$\frac{\partial \omega^i}{\partial x_j}(x) = \frac{\partial^2 h}{\partial x_j \partial x_i}(x) = \frac{\partial^2 h}{\partial x_i \partial x_j}(x) = \frac{\partial \omega^j}{\partial x_i}(x)$$

Therefore, for the PDE to have a solution, the integrability conditions

$$\frac{\partial \omega^i}{\partial x_j}(x) - \frac{\partial \omega^j}{\partial x_i}(x) = 0$$

must be satisfied. The exterior derivative of a one form is a skew-symmetric matrix field

$$d\omega(x) = \sum_{i < j} \left(\frac{\partial \omega^i}{\partial x_j}(x) - \frac{\partial \omega^j}{\partial x_i}(x) \right) dx_i \wedge dx_j$$

A one form $\omega(x)$ is said to be closed if $d\omega(x) = 0$. This is locally sufficient for there to exist an $h(x)$ such that $dh(x) = \omega(x)$.

Every exact form is closed but not every closed form is exact. A counter example on $\mathbb{R}^2$ is

$$\omega(x) = [-x_2 \ x_1]$$

This is closed but not exact. The line integral of this convector field around any circle centered at the origin is 2π . If it were exact, the line integral would have been zero because the curve ends where it begins.

The Lie derivative of a scalar-valued function $h(x)$ by a vector field $f(x)$ is denoted to be the scalar-valued function

$$L_f(h)(x) = \frac{\partial h}{\partial x}(x)f(x) = \langle dh(x), f(x) \rangle$$

This can be iterated

$$L_f^k(h)(x) = \frac{\partial L_f^{k-1}h}{\partial x}(x)f(x)$$

If $h(x) \in \mathbb{R}^{p \times 1}$, then $L_f(h)(x) \in \mathbb{R}^{p \times 1}$.

The Lie bracket of two vector fields $f^1(x)$ and $f^2(x)$ is another vector field

$$[f^1, f^2](x) = \frac{\partial f^2}{\partial x}(x)f^1(x) - \frac{\partial f^1}{\partial x}(x)f^2(x)$$

Clearly, the Lie bracket is skew symmetric, $[f^1, f^2](x) = -[f^2, f^1](x)$. It also satisfies the Jacobi identity

$$[f^1, [f^2, f^3]](x) + [f^2, [f^3, f^1]](x) + [f^3, [f^1, f^2]](x) = 0$$

Repeated Lie brackets are often expressed inductively as

$$ad_f^0(g)(x) = g(x)$$

$$ad_f^k(g)(x) = [f, ad_f^{k-1}(g)](x)$$

The geometric interpretation of the Lie bracket $[f, g](x)$ is the infinitesimal commutator of their flows $\phi(t, x)$ and $\psi(t, x)$, i.e.,

$$\psi(t, \phi(t, x)) - \phi(t, \psi(t, x)) = [f, g](x)t^2 + O(t)^3$$

Another interpretation of the Lie bracket is given by the Lie series expansion

$$g(\phi(t, x)) = \sum_{k=0}^{\infty} (-1)^k ad_f^k(g)(x) \frac{t^k}{k!}$$

This is a Taylor series expansion which is convergent for small $|t|$ if f, g are real analytic vector fields. Another Lie series is

$$h(\phi(t, x)) = \sum_{k=0}^{\infty} L_f^k(h)(x) \frac{t^k}{k!}$$

Given a smooth mapping $z = \theta(x)$ from an open subset of $\mathbb{R}^n$ to an open subset of $\mathbb{R}^m$ and vector fields $f(x)$ and $g(z)$ on these open subsets, we say $f(x)$ is θ -related to $g(z)$ if

$$g(\theta(x)) = \frac{\partial \theta}{\partial x}(x) f(x)$$

It is not hard to see that if $f^1(x), f^2(x)$ are θ -related to $g^1(z), g^2(z)$, then $[f^1, f^2](x)$ is θ -related to $[g^1, g^2](z)$. For this reason, we say that the Lie bracket is an intrinsic differentiation. The other intrinsic differentiation is the exterior derivative operation d .

The Lie derivative of a one form $\omega(x)$ by a vector field $f(x)$ is given by

$$L_f(\omega)(x) = \sum_{i,j} \left(\frac{\partial \omega^i}{\partial x_j}(x) f_j(x) + \omega_j(x) \frac{\partial f_j}{\partial x_i}(x) \right) dx_i$$

It is not hard to see that

$$L_f(\langle \omega, g \rangle)(x) = \langle L_f(\omega), g \rangle(x) + \langle \omega, [f, g] \rangle(x)$$

and

$$L_f(dh)(x) = d(L_f(h))(x) \quad (3)$$

Control systems involve multiple vector fields. A distribution $\mathcal{D}$ is a set of vector fields on $\mathcal{M}$ that is closed under addition of vector fields and under multiplication by scalar functions. A distribution defines at each $x \in \mathcal{M}$ a subspace of the tangent space

$$D(x) = \{f(x) : f \in \mathcal{D}\}$$

These subspaces form a subbundle D of the tangent bundle. If the subspaces are all of the same dimension, then the distribution is said to be nonsingular. We will restrict our attention to nonsingular distributions.

A codistribution (or Pfaffian system) $\mathcal{E}$ is a set of one forms on $\mathcal{M}$ that is closed under

addition and multiplication by scalar functions. A codistribution defines at each $x \in \mathcal{M}$ a subspace of the cotangent space

$$\{\omega(x) : \omega \in \mathcal{E}\}$$

These subspaces form a subbundle E of the cotangent bundle. If the subspaces are all of the same dimension, then the codistribution is said to be nonsingular. Again, we will restrict our attention to nonsingular codistributions.

Every distribution $\mathcal{D}$ defines a dual codistribution

$$\mathcal{D}^* = \{\omega(x) : \omega(x)f(x) = 0, \text{ for all } f(x) \in \mathcal{D}\}$$

and vice versa

$$\mathcal{E}_* = \{f(x) : \omega(x)f(x) = 0, \text{ for all } \omega(x) \in \mathcal{E}\}$$

A k dimensional distribution $\mathcal{D}$ (or its dual codistribution $\mathcal{D}^*$) can be thought of as a system of PDEs on $\mathcal{M}$. Find $n - k$ independent functions $h_1(x), \dots, h_{n-k}(x)$ such that

$$dh_i(x)f(x) = 0 \text{ for all } f(x) \in \mathcal{D}$$

The functions $h_1(x), \dots, h_{n-k}(x)$ are said to be independent if $dh_1(x), \dots, dh_{n-k}(x)$ are linearly independent at every $x \in \mathcal{M}$. In other words, $dh_1(x), \dots, dh_{n-k}(x)$ span $\mathcal{D}^*$ over the space of smooth functions.

The Frobenius theorem gives the integrability conditions for these functions to exist locally. The distribution $\mathcal{D}$ must be involutive, i.e., closed under the Lie bracket,

$$[\mathcal{D}, \mathcal{D}] = \{[f, g] : f, g \in \mathcal{D}\} \subset \mathcal{D}$$

When the functions exist, their joint level sets $\{x : h_i(x) = c_i, i = 1, \dots, n - k\}$ are the leaves of a local foliation. Through each x^0 in a convex local coordinate chart $\mathcal{N}$, there exists locally a k -dimensional submanifold $\{x \in \mathcal{N} : h_i(x) = h_i(x^0)\}$. At each x^1 in this submanifold, its tangent space is $D(x)$. Whether these $h_i(x)$ exist globally to define a global foliation, a partition of $\mathcal{M}$ into smooth submanifolds, is

a delicate question. Consider a distribution on $\mathbb{R}^2$ generated by a constant vector field $f(x) = b$ of irrational slope, b_2/b_1 is irrational. Construct the torus T^2 as the quotient of $\mathbb{R}^2$ by the integer lattice Z^2 . The distribution passes to the quotient and since it is one dimensional, it is clearly involutive. The leaves of the quotient distribution are curves that wind around the torus indefinitely, and each curve is dense in T^2 . Therefore, any smooth function $h(x)$ that is constant on such a leaf is constant on all of T^2 . Hence, the local foliation does not extend to a global foliation.

Another delicate question is whether the quotient space of $\mathcal{M}$ by a foliation induced by an involutive distribution is a smooth manifold (Sussmann 1975). This is always true locally, but it may not hold globally. Think of the foliation of T^2 discussed above.

Given $k \leq n$ vector fields $f^1(x), \dots, f^k(x)$ that are linearly independent at each x and that commute, $[f^i, f^j](x) = 0$, there exists a local change of coordinates $z = z(x)$ so that in the new coordinates the vector fields are the first k unit vectors.

The involutive closure $\bar{\mathcal{D}}$ of $\mathcal{D}$ is the smallest involutive distribution containing $\mathcal{D}$. As with all distributions, we always assume implicitly that is nonsingular. A point x^1 is $\mathcal{D}$ -accessible from x^0 if there exists a continuous and piecewise smooth curve joining x^0 to x^1 whose left and right tangent vectors are always in $\mathcal{D}$. Obviously $\mathcal{D}$ -accessibility is an equivalence relation. Chow's theorem (1939) asserts that its equivalence classes are the leaves of the foliation induced by $\bar{\mathcal{D}}$. Chow's theorem goes a step further. Suppose $f^1(x), \dots, f^k(x)$ span $\mathcal{D}(x)$ at each $x \in \mathcal{M}$ then given any two points x^0, x^1 in a leaf of $\bar{\mathcal{D}}$, there is a continuous and piecewise smooth curve joining x^0 to x^1 whose left and right tangent vectors are always one of the $f^i(x)$.

Controllability of Nonlinear Systems

An initialized nonlinear system that is affine in the control is of the form

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ &= f(x) + \sum_{j=1}^m g^j(x)u_j \\ y &= h(x) \\ x(0) &= x^0\end{aligned}\quad (4)$$

where the state x are local coordinates on an n -dimensional manifold $\mathcal{M}$, the control u is restricted to lie in some set $\mathcal{U} \subset \mathbb{R}^m$, and the output y takes values $\mathbb{R}^p$. We shall only consider such systems.

A particular case is a linear system of the form

$$\begin{aligned}\dot{x} &= Fx + Gu \\ y &= Hx \\ x(0) &= x^0\end{aligned}\quad (5)$$

where $\mathcal{M} = \mathbb{R}^n$ and $\mathcal{U} = \mathbb{R}^m$.

The set $\mathcal{A}_t(x^0)$ of points accessible at time $t \geq 0$ from x^0 is the set of all $x^1 \in \mathcal{M}$ such that there exists a bounded, measurable control trajectory $u(s) \in \mathcal{U}$, $0 \leq s \leq t$, so that the solution of (4) satisfies $x(t) = x^1$. We define $\mathcal{A}(x^0)$ as the union of $\mathcal{A}_t(x^0)$ for all $t \geq 0$. The system (4) is said to be controllable at time $t > 0$ if $\mathcal{A}_t(x^0) = \mathcal{M}$ and controllable in forward time if $\mathcal{A}(x^0) = \mathcal{M}$.

For linear systems, controllability is a rather straightforward matter, but for nonlinear systems it is more subtle with numerous variations. The variation of constants formula gives the solution of the linear system (5) as

$$x(t) = e^{Ft}x^0 + \int_0^t e^{F(t-s)}Gu(s)ds$$

so $\mathcal{A}_t(x^0)$ is an affine subspace of $\mathbb{R}^n$ for any $t > 0$. It is not hard to see that the columns of

$$\begin{bmatrix} G & \dots & F^{n-1}G \end{bmatrix}\quad (6)$$

are tangent to this affine subspace, so if this matrix is of rank n , then $\mathcal{A}_t(x^0) = \mathbb{R}^n$ for any $t > 0$. This is the so-called controllability rank condition for linear systems.

Turning to the nonlinear system (4), let $\mathcal{D}$ be the distribution spanned by the vector fields $f(x), g^1(x), \dots, g^m(x)$, and let $\bar{\mathcal{D}}$ be its involutive closure. It is clear that $\mathcal{A}(x^0)$ is contained in

the leaf of $\bar{\mathcal{D}}$ through x^0 . Krener (1971) showed that $\mathcal{A}(x^0)$ has nonempty interior in this leaf. More precisely, $\mathcal{A}(x^0)$ is between an open set and its closure in the relative topology of the leaf.

Let $\mathcal{D}_0$ be the smallest distribution containing the vector fields $g^1(x), \dots, g^m(x)$ and invariant under bracketing by $f(x)$, i.e.,

$$[f, \mathcal{D}_0] \subset \mathcal{D}_0$$

and let $\bar{\mathcal{D}}_0$ be its involutive closure. Sussmann and Jurdjevic (1972) showed that $\mathcal{A}_t(x^0)$ is in the leaf of $\bar{\mathcal{D}}_0$ through x^0 , and it is between an open set and its closure in the topology of this leaf.

For linear systems (5) where $f(x) = Fx$ and $g^j(x) = G^j$, the j th column of G , it is not hard to see that

$$ad_f^k(g^j)(x) = (-1)^k F^k G^j$$

so the nonlinear generalization of the controllability rank condition is that at each x the dimension of $\bar{\mathcal{D}}_0(x)$ is n . This guarantees that $\mathcal{A}_t(x^0)$ is between an open set and its closure in the topology of $\mathcal{M}$.

The condition that

$$\text{dimension } \bar{\mathcal{D}}(x) = n \quad (7)$$

is referred to as the nonlinear controllability rank condition. This guarantees that $\mathcal{A}(x^0)$ is between an open set and its closure in the topology of $\mathcal{M}$.

There are stronger interpretations of controllability for nonlinear systems. One is short time local controllability (STLC). The definition of this is that the set of accessible points from x^0 in any small $t > 0$ with state trajectories restricted to an arbitrarily small neighborhood of x^0 should contain x^0 in its interior. Hermes (1994) and others have done work on this.

Observability for Nonlinear Systems

Two possible initial states x^0, x^1 for the nonlinear system are distinguishable if there exists a control $u(\cdot)$ such that the corresponding

outputs $y^0(t), y^1(t)$ are not equal. They are short time distinguishable if there is an $u(\cdot)$ such that $y^0(t) \neq y^1(t)$ for all small $t > 0$. They are locally short time distinguishable if in addition the corresponding state trajectories do not leave an arbitrarily small open set containing x^0, x^1 . The open set need not be connected. A nonlinear system is (short time, locally short time) observable if every pair of initial states is (short time, locally short time) distinguishable. Finally, a nonlinear system is (short time, locally short time) locally observable if every x^0 has a neighborhood such that every other point x^1 in the neighborhood is (short time, locally short time) distinguishable from x^0 .

For a linear system (5), all these definitions coalesce into a single concept of observability which can be checked by the observability rank condition which is that the rank of

$$\begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n-1} \end{bmatrix} \quad (8)$$

equals n .

The corresponding concept for nonlinear systems involves $\mathcal{E}$, the smallest codistribution containing $dh_1(x), \dots, dh_p(x)$ that is invariant under repeated Lie differentiation by the vector fields $f(x), g^1(x), \dots, g^m(x)$. Let

$$E(x) = \{\omega(x) : \omega \in \mathcal{E}\}$$

The nonlinear observability rank condition is

$$\text{dimension } E(x) = n \quad (9)$$

for all $x \in \mathcal{M}$. This condition guarantees that the nonlinear system is locally short time, locally observable. It follows from (3) that $\mathcal{E}$ is spanned by a set of exact one form, so its dual distribution $\mathcal{E}^*$ is involutive.

For a linear system (5), the input–output mapping from $x(0) = x^i, i = 0, 1$ is

$$y^i(t) = He^{Ft}x^i + \int_0^t He^{F(t-s)}Gu(s)ds$$

The difference is

$$y1(t) - y^0(t) = He^{Ft}(x^1 - x^0)$$

So if one input $u(\cdot)$ distinguishes x^0 from x^1 , then so does every input.

For nonlinear systems, this is not necessarily true. That prompted Gauthier et al. (1992) to introduce a stronger concept of observability for nonlinear systems. For simplicity, we describe it for scalar input and scalar output systems. A nonlinear system is uniformly observable for any input if there exist local coordinates so that it is of the form

$$\begin{aligned} y &= x_1 + h(u) \\ \dot{x}_1 &= x_2 + f_1(x_1, u) \\ &\vdots \\ \dot{x}_{n-1} &= x_n + f_{n-1}(x_1, \dots, x_{n-2}, u) \\ \dot{x}_n &= f_n(x, u) \end{aligned}$$

Clearly if we know $u(\cdot)$, $y(\cdot)$, then by repeated differentiation of $y(\cdot)$, we can reconstruct $x(\cdot)$. It has been shown that for nonlinear systems that are uniformly observable for any input, the extended Kalman filter is a locally convergent observer (Krener 2002a), and the minimum energy estimator is globally convergent (Krener 2002b).

Minimal Realizations

The initialized nonlinear system (4) can be viewed as defining an input–output mapping from input trajectories $u(\cdot)$ to output trajectories $y(\cdot)$. Is it a minimal realization of this mapping, does there exist an initialized nonlinear system on a smaller dimensional state space that realizes the same input–output mapping?

Kalman showed that (5) initialized at $x^0 = 0$ is minimal iff the controllability rank condition

and the observability rank condition hold. He also showed how to reduce a linear system to a minimal one.

If the controllability rank condition does not hold, then the span of (6) dimension is $k < n$. This subspace contains the columns of G and is invariant under multiplication by F . In fact, it is the maximal subspace with these properties. So the linear system can be restricted to this k -dimensional subspace, and it realizes the same input–output mapping from $x^0 = 0$. The restricted system satisfies the controllability rank condition.

If the observability rank condition does not hold, then the kernel of (8) is a subspace of $\mathbb{R}^{n \times 1}$ of dimension $n - l > 0$. This subspace is in the kernel of H and is invariant under multiplication by F . In fact, it is the maximal subspace with these properties. Therefore, there is a quotient linear system on the $\mathbb{R}^{n \times 1}$ mod, the kernel of (8) which has the same input–output mapping. The quotient is of dimension $l < n$, and it realizes the same input–output mapping. The quotient system satisfies the observability rank condition.

By employing these two steps in either order, we pass to a minimal realization of the input–output map of (5) from $x^0 = 0$. Kalman also showed that two linear minimal realizations differ by a linear change of state coordinates.

An initialized nonlinear system is a realization of minimal dimension of its input–output mapping if the nonlinear controllability rank condition (7) and the nonlinear observability rank condition (9) hold.

If the nonlinear controllability rank condition (7) fails to hold because the dimension of $D(x)$ is $k < n$, then by replacing the state space $\mathcal{M}$ with the k dimensional leaf through x^0 of the foliation induced by $\mathcal{D}$, we obtain a smaller state space on which the nonlinear controllability rank condition (7) holds. The input–output mapping is unchanged by this restriction.

Suppose the nonlinear observability rank condition (9) fails to hold because the dimension of $E(x)$ is $l < n$. Then consider a convex neighborhood $\mathcal{N}$ of x^0 . The distribution $\mathcal{E}^*$ induces a local foliation of $\mathcal{N}$ into leaves of dimension $n - l > 0$. The nonlinear system leaves this

local foliation invariant in the following sense. Suppose x^0 and x^1 are on the same leaf then if $x^i(t)$ is the trajectory starting at x^i , then $x^0(t)$ and $x^1(t)$ are on the same leaf as long as the trajectories remain in $\mathcal{N}$. Furthermore, $h(x)$ is constant on leaves so $y^0(t) = y^1(t)$. Hence, there exists locally a nonlinear system whose state space is the leaf space. On this leaf space, the nonlinear observability rank condition holds (9), and the projected system has the same input–output map as the original locally around x^0 . If the leaf space of the foliation induced by $\mathcal{E}^*$ admits the structure of a manifold, then the reduced system can be defined globally on it. Sussmann (1973) and Sussmann (1977) studied minimal realizations of analytic nonlinear systems.

The state space of two minimal nonlinear systems need not be diffeomorphic. Consider the system

$$\dot{x} = u, \quad y = \sin x$$

where x, u are scalars. We can take the state space to be either $\mathcal{M} = \mathbb{R}$ or $\mathcal{M} = S^1$, and we will realize the same input–output mapping. These two state spaces are certainly not diffeomorphic but one is a covering space of the other.

Lie Jet and Approximations

Consider two initialized nonlinear controlled dynamics

$$\begin{aligned} \dot{x} &= f^0(x) + \sum_{j=1}^m f^j(x)u_j \\ x(0) &= x^0 \end{aligned} \quad (10)$$

$$\begin{aligned} \dot{z} &= g^0(z) + \sum_{j=1}^m g^j(z)u_j \\ z(0) &= z^0 \end{aligned} \quad (11)$$

Suppose that (10) satisfies the nonlinear controllability rank condition (7). Further, suppose that there is a smooth mapping $\Phi(x) = z$ and constants $M > 0, \epsilon > 0$ such that for any $\|u(t)\| < 1$, the corresponding trajectories $x(t)$ and $z(t)$ satisfy

$$\|\Phi(x(t)) - z(t)\| < Mt^{k+1} \quad (12)$$

for $0 \leq t < \epsilon$.

Then it is not hard to show that the linear map $L = \frac{\partial \Phi}{\partial x}(x^0)$ takes brackets up to order k of the vector fields f^j evaluated at x^0 into the corresponding brackets of the vector fields g^j evaluated at z^0 ,

$$\begin{aligned} L[f^{j_l}[\dots[f^{j_2}, f^{j_1}]\dots]](x^0) \\ = [g^{j_l}[\dots[g^{j_2}, g^{j_1}]\dots]](z^0) \end{aligned} \quad (13)$$

for $1 \leq l \leq k$.

On the other hand, if there is a linear map L such that (13) holds for $1 \leq l \leq k$, then there exists a smooth mapping $\Phi(x) = z$ and constants $M > 0, \epsilon > 0$ such that for any $\|u(t)\| < 1$, the corresponding trajectories $x(t)$ and $z(t)$ satisfy (12).

The k -Lie jet of (10) at x^0 is the tree of brackets $[f^{j_l}[\dots[f^{j_2}, f^{j_1}]\dots]](x^0)$ for $1 \leq l \leq k$. In some sense these are the coordinate-free Taylor series coefficients of (10) at x^0 .

The dynamics (10) is free-nilpotent of degree k if all these brackets are as linearly independent as possible consistent with skew symmetry and the Jacobi identity and all higher degree brackets are zero. If it is free to degree k , the controlled dynamics (10) can be used to approximate any other controlled dynamics (11) to degree k . Because it is nilpotent, integrating (10) reduces to repeated quadratures. If all brackets with two or more f^i , $1 \leq i \leq m$ are zero at x^0 , then (10) is linear in appropriate coordinates. If all brackets with three or more f^i , $1 \leq i \leq m$ are zero at x^0 , then (10) is quadratic in appropriate coordinates. The references Krener and Schaettler (1988) and Krener (2010a, b) discuss the structure of the reachable sets for such systems.

Disturbance Decoupling

Consider a control system affected by a disturbance input $w(t)$

$$\begin{aligned}\dot{x} &= f(x) + g(x)u + b(x)w \\ y &= h(x)\end{aligned}\quad (14)$$

The disturbance decoupling problem is to find a feedback $u = \kappa(x)$, so that in the closed-loop system, the output $y(t)$ is not affected by the disturbance $w(t)$. Wonham and Morse (1970) solved this problem for a linear system

$$\begin{aligned}\dot{x} &= Fx + Gu + Bw \\ y &= Hx\end{aligned}\quad (15)$$

To do so, they introduced the concept of an F, G invariant subspace. A subspace $\mathcal{V} \subset \mathbb{R}^n$ is F, G invariant if

$$F\mathcal{V} \subset \mathcal{V} + \mathcal{G} \quad (16)$$

where $\mathcal{G}$ is the span of the columns of G . It is easy to see that $\mathcal{V}$ is F, G invariant iff there exists a $K \in \mathbb{R}^{m \times n}$ such that

$$(F + GK)\mathcal{V} \subset \mathcal{V} \quad (17)$$

The feedback gain K is called a friend of $\mathcal{V}$.

It is easy from (16) that if $\mathcal{V}^i$ is F, G invariant for $i = 1, 2$, then $\mathcal{V}^1 + \mathcal{V}^2$ is also. So there exists a maximal F, G invariant subspace $\mathcal{V}^{\max}$ in the kernel of H . Wonham and Morse showed that the linear disturbance decoupling problem is solvable iff $\mathcal{B} \subset \mathcal{V}^{\max}$ where $\mathcal{B}$ is the span of the columns of B .

Isidori et al. (1981a) and independently Hirschorn (1981) solved the nonlinear disturbance decoupling problem. A distribution $\mathcal{D}$ is locally f, g invariant if

$$\begin{aligned}[f, \mathcal{D}] &\subset \mathcal{D} + \Gamma \\ [g^j, \mathcal{D}] &\subset \mathcal{D} + \Gamma\end{aligned}\quad (18)$$

for $j = 1, \dots, m$ where Γ is the distribution spanned by the columns of g . In Isidori et al. (1981b) it is shown that if $\mathcal{D}$ is locally f, g invariant, then so is its involutive closure.

A distribution $\mathcal{D}$ is f, g invariant if there exists $\alpha(x) \in \mathbb{R}^{m \times 1}$ and invertible $\beta(x) \in \mathbb{R}^{m \times m}$ such that

$$\begin{aligned}[f + g\alpha, \mathcal{D}] &\subset \mathcal{D} \\ \left[\sum_j g^j \beta_j^k, \mathcal{D}\right] &\subset \mathcal{D}\end{aligned}\quad (19)$$

for $k = 1, \dots, m$. It is not hard to see that a f, g invariant is locally f, g invariant. It is shown in Isidori et al. (1981b) that if $\mathcal{D}$ is a locally f, g invariant distribution, then locally there exists $\alpha(x)$ and $\beta(x)$ so that (19) holds. Furthermore, if the state space is simply connected, then $\alpha(x)$ and $\beta(x)$ exist globally, but the matrix field $\beta(x)$ may fail to be invertible at some x .

From (18), it is clear that if $\mathcal{D}^i$ is locally f, g invariant for $i = 1, 2$, then so is $\mathcal{D}^1 + \mathcal{D}^2$. Hence, there exists a maximal locally f, g invariant distribution $\mathcal{D}^{\max}$ in the kernel of dh . Moreover, this distribution is involutive. The disturbance decoupling problem is locally solvable iff columns of $b(x)$ are contained in $\mathcal{D}^{\max}$. If $\mathcal{M}$ is simply connected, then the disturbance decoupling problem is globally solvable iff columns of $b(x)$ are contained in $\mathcal{D}^{\max}$.

Conclusion

We have briefly described the role that differential geometric concepts played in the development of controllability, observability, minimality, approximation, and decoupling of nonlinear systems.

Cross-References

- [Feedback Linearization of Nonlinear Systems](#)
- [Lie Algebraic Methods in Nonlinear Control](#)
- [Nonlinear Zero Dynamics](#)

Bibliography

- Arnol'd VI (1983) Geometrical methods in the theory of ordinary differential equations. Springer, Berlin
- Brockett RW (1972) Systems theory on group manifolds and coset spaces. SIAM J Control 10: 265–284
- Chow WL (1939) Über Systeme von Linearen Partiellen Differentialgleichungen Erster Ordnung. Math Ann 117:98–105
- Gauthier JP, Hammouri H, Othman S (1992) A simple observer for nonlinear systems with applications to bioreactors. IEEE Trans Autom Control 37:875–880

- Griffith EW, Kumar KSP (1971) On the observability of nonlinear systems. I. *J Math Anal Appl* 35:135–147
- Haynes GW, Hermes H (1970) Non-linear controllability via Lie theory *SIAM J Control* 8: 450–460
- Hermann R, Krener AJ (1977) Nonlinear controllability and observability. *IEEE Trans Autom Control* 22:728–740
- Hermes H (1994) Large and small time local controllability. In: *Proceedings of the 33rd IEEE conference on decision and control*, vol 2, pp 1280–1281
- Hirschorn RM (1981) (A,B)-invariant distributions and the disturbance decoupling of nonlinear systems. *SIAM J Control Optim* 19:1–19
- Isidori A, Krener AJ, Gori Giorgi C, Monaco S (1981a) Nonlinear decoupling via feedback: a differential geometric approach. *IEEE Trans Autom Control* 26:331–345
- Isidori A, Krener AJ, Gori Giorgi C, Monaco S (1981b) Locally (f,g) invariant distributions. *Syst Control Lett* 1:12–15
- Kostyukovskii YML (1968a) Observability of nonlinear controlled systems. *Autom Remote Control* 9:1384–1396
- Kostyukovskii YML (1968b) Simple conditions for observability of nonlinear controlled systems. *Autom Remote Control* 10:1575–1584–1396
- Kou SR, Elliot DL, Tarn TJ (1973) Observability of nonlinear systems. *Inf Control* 22: 89–99
- Krener AJ (1971) A generalization of the Pontryagin maximal principle and the bang-bang principle. PhD dissertation, University of California, Berkeley
- Krener AJ (1974) A generalization of Chow's theorem and the bang-bang theorem to nonlinear control problems. *SIAM J Control* 12:43–52
- Krener AJ (1975) Local approximation of control systems. *J Differ Equ* 19:125–133
- Krener AJ (2002a) The convergence of the extended Kalman filter. In: Rantzer A, Byrnes CI (eds) *Directions in mathematical systems theory and optimization*. Springer, Berlin, pp 173–182. Corrected version available at arXiv:math.OA/0212255 v. 1
- Krener AJ (2002b) The convergence of the minimum energy estimator. I. In: Kang W, Xiao M, Borges C (eds) *New trends in nonlinear dynamics and control, and their applications*. Springer, Heidelberg, pp 187–208
- Krener AJ (2010a) The accessible sets of linear free nilpotent control systems. In: *Proceeding of NOLCOS 2010*, Bologna
- Krener AJ (2010b) The accessible sets of quadratic free nilpotent control systems. *Commun Inf Syst* 11:35–46
- Krener AJ (2013) Feedback linearization of nonlinear systems. Baillieul J, Samad T (eds) *Encyclopedia of systems and control*. Springer
- Krener AJ, Schaettler H (1988) The structure of small time reachable sets in low dimensions. *SIAM J Control Optim* 27:120–147
- Lobry C (1970) Controllabilite des Systemes Non Lineaires. *SIAM J Control* 8:573–605

- Sussmann HJ (1973) Minimal realizations of nonlinear systems. In: Mayne DQ, Brockett RW (eds) *Geometric methods in systems theory*. D. Ridel, Dordrecht
- Sussmann HJ (1975) A generalization of the closed subgroup theorem to quotients of arbitrary manifolds. *J Differ Geom* 10:151–166
- Sussmann HJ (1977) Existence and uniqueness of minimal realizations of nonlinear systems. *Math Syst Theory* 10:263–284
- Sussmann HJ, Jurdjevic VJ (1972) Controllability of nonlinear systems. *J Differ Equ* 12:95–116
- Wonham WM, Morse AS (1970) Decoupling an pole assignment in linear multivariable systems: a geometric approach. *SIAM J Control* 8:1–18

Disaster Response Robot

Satoshi Tadokoro

Tohoku University, Sendai, Japan

Abstract

Disaster response robots are robotic systems used for preventing outspread of disaster damage under emergent situations. Robots for natural disasters (water disaster, volcano eruption, earthquakes, flood, landslides, and fire) and man-made disasters (explosive ordnance disposal, CBRNE disasters, Fukushima-Daiichi nuclear power plant accident) are introduced. Technical challenges are described on the basis of generalized data flow.

Keywords

Rescue robot · Response robot

Introduction

Disaster response robots are robotic systems used for preventing outspread of disaster damage under emergent situations, such as for search and rescue, recovery construction, etc.

Disaster changes its state as time passes. It enters by unforeseen occurrence from prevention phase to emergency response phase, recovery phase, and revival phase. Although a disaster

response robot usually means a system for disaster response and recovery in a narrow sense, a system used in every phase of disaster can be called a disaster response robot in a broad sense.

When parties of firefighters and military respond to disaster, robots are one of the technical equipment. The purposes of robots are (1) to perform tasks impossible/difficult by human and conventional equipment, (2) to reduce responders' risk for preventing secondary damage, and (3) to improve rapidness/efficiency of tasks, by using remote/automatic robot equipment.



Disaster Response Robot, Fig. 1 SARbot. (By courtesy of SeaBotix, Inc.) <http://www.seabotix.com/products/sarbot.htm>

Response Robots for Natural Disasters

Water Disaster

Underwater robots (ROV, remotely operated vehicle) are deployed to responder organizations in preparation for water damage such as at tsunami, flood, cataract, and accidents in the sea and rivers. They are equipped with cameras and sonars and remotely controlled by crews via tether from land or shipboard within several tens meters area for victim search and damage investigation. At Great Eastern Japan Earthquake in 2011, the Self-Defense Force and volunteers of the International Rescue System Institute (IRS) and Center for Robot-Assisted Search and Rescue (CRASAR) used various types of ROVs such as SARbot shown in Fig. 1 for victim search and debris investigation in port.

Volcano Eruption

In order to reduce risk of monitoring and recovery construction at volcano eruption, application of robotics and remote systems is highly expected. Various types of UAVs (unmanned aerial vehicles) such as small-size robot helicopters and airplanes have been used for this purpose. Unmanned construction system consists of tele-operated robot backhoes, trucks, and bulldozers with wireless relaying cars and camera vehicles as shown in Fig. 2 and is remotely controlled



Disaster Response Robot, Fig. 2 Unmanned construction system. (By courtesy of Society for Unmanned Construction Systems) http://www.kenmukyou.gr.jp/f_souti.htm

from an operator vehicle. It has been used since the 1990 for remote civil engineering works from a distance of a few kilometers.

Structural Collapse by Earthquakes, Landslides, etc.

Small-sized UGVs (unmanned ground vehicles) were developed for victim search and monitoring in confined space of collapsed buildings and underground structures. VGTV-Xtreme shown in Fig. 3 is a tracked vehicle remotely operated via a tether. It was used for victim search at mine accidents and the 9.11 terror. Active Scope Camera shown in Fig. 4 is a serpentine robot like a fiberscope and was used for forensic investigation of structural collapse accidents.

Fire

Large-scale fires of chemical plants and forests sometimes have a high risk, and firefighters cannot approach near them. Remote-controlled robots with firefighting nozzles for water and chemical extinguishing agents are deployed. Large-size robots can discharge large volume of the fluid with water cannon, and small-size ones

have better mobility. Dragon Firefighter (Fig. 5) is a flying hose robot for extinguishing fire in confined space to minimize the risk of human firefighters at the operation inside.

Flood and Landslide

Unmanned aerial vehicles are effective tools for gathering overview information of flood and landslide where human responders cannot access or need time to cover wide damaged area. Captured 2D/3D images as shown in Fig. 6 are used for identifying damaged area, finding victims, estimating the amount of damage, and supporting recovery.

Response Robots for Man-Made Disasters

Explosive Ordnance Disposal (EOD)

Detection and disposal of explosive ordnance is one of the most dangerous tasks. TALON, PackBot, and Telemex are widely used in military and explosive ordnance disposal teams worldwide. Telemex has an arm with seven

Disaster Response Robot, Fig. 3

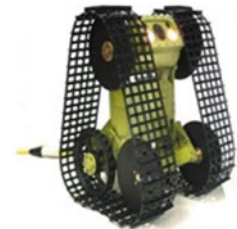
VGTV-Xtreme. (By courtesy of Recce Robotics)<http://www.recce-robotics.com/vgtv.html>



VG TV X-treme
in Lowered Position



and many other
configurations



VG TV X-treme
in Raised Position

Disaster Response Robot, Fig. 4

Active Scope Camera. (By courtesy of International Rescue System Institute)





Disaster Response Robot, Fig. 5 Dragon Firefighter. (By courtesy of Tohoku University)



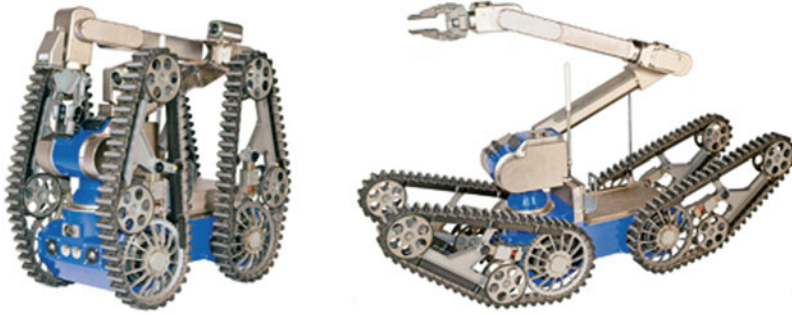
Disaster Response Robot, Fig. 6 High-resolution ortho-image captured by UAV. (By courtesy of ImPACT-TRC)

degrees of freedom on a tracked vehicle with four sub-tracks as shown in Fig. 7. It can observe narrow spaces like overhead lockers of airplanes and bottom of automobiles by cameras, manipulate objects by the arm, and deactivate explosives by a disrupter.

CBRNE Disasters

CBRNE (chemical, biological, radiological, nuclear, and explosive) disasters have a high

risk and can cause large-scale damage because human cannot detect contamination by the hazmat (hazardous materials). Application of robotic systems is highly expected for this disaster. PackBot has sensors for toxic industrial chemicals (TIC), blood agents, blister agents, volatile organic compounds (VOCs), radiation, etc. as options and can measure the hazmat in dangerous confined spaces (Fig. 8). Quince was developed for research into technical issues of



Disaster Response Robot, Fig. 7 Telex. (By courtesy of Cobham Mission Equipment) <http://www.cobham.com/about-cobham/mission-systems/about-us/>

[mission-equipment/unmanned-systems/products-and-services/remote-controlled-robotic-solutions/telex-explosive-ordnance-\(eod\)-robot.aspx](http://www.cobham.com/about-cobham/mission-systems/about-us/mission-equipment/unmanned-systems/products-and-services/remote-controlled-robotic-solutions/telex-explosive-ordnance-(eod)-robot.aspx)



Disaster Response Robot, Fig. 8 PackBot. (By courtesy of iRobot) <http://www.irobot.com/us/learn/defense/packbot/Specifications.aspx>

Fukushima-Daiichi Nuclear Power Plant Accident

At Fukushima-Daiichi nuclear power plant accident caused by tsunami in 2011, various disaster response robots were applied. They contributed to the cool shutdown and decommissioning of the plant. For example, PackBot and Quince gave essential data for task planning by shooting images and radiation measurement in nuclear reactor buildings there. Unmanned construction system removed debris outdoors that were contaminated by radiological materials and reduced the radiation rate there significantly.

Groupe INTRA in France and KHG in Germany are organizations for responding nuclear plant accidents. They are equipped with robots and remote-controlled construction machines for radiation measurement, decontamination, and constructions in emergency. In Japan, the Assist Center for Nuclear Emergencies was established after the Fukushima accident.

Many types of remote systems (robots), such as PMORPH (Fig. 10), have been applied for preventing exposure to radiation at the decommissioning work of the plant toward removal of fuel debris.



Disaster Response Robot, Fig. 9 Quince. (By courtesy of International Rescue System Institute)

UGVs at CBRNE disasters and has high mobility on rough terrain (Fig. 9).

Summary and Future Directions

Data flow of disaster response robots is generally described by a feedback system as shown in Fig. 11. Robots change the states of objects and

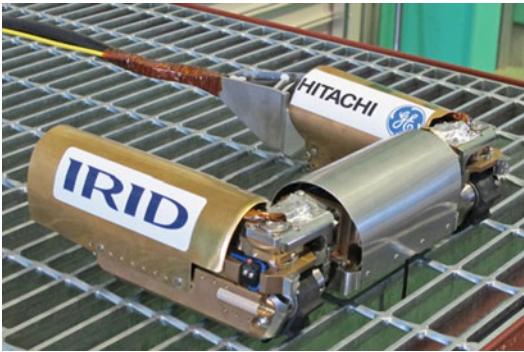
environment by movement and task execution. Sensors measure and recognize them, and their feedback enables the robots autonomous motion and work. The sensed data are shown to operators via communication, data processing, and a human interface. The operators give commands of motion and work to the system via the human interface. The system recognizes and transmits them to the robot.

Each functional block has its own technical challenges to be fulfilled under extreme environments of disaster space in response to the objectives and the conditions. They should be solved technically in order to improve the robot performance. They include insufficient mobility in disaster spaces (steps, gaps, slippage, narrow space, obstacles, etc.), deficient workability (dexterity, accuracy, speed, force, work space, etc.), poor sensors and sensor data processing (image, recognition, etc.), lack of reliability and

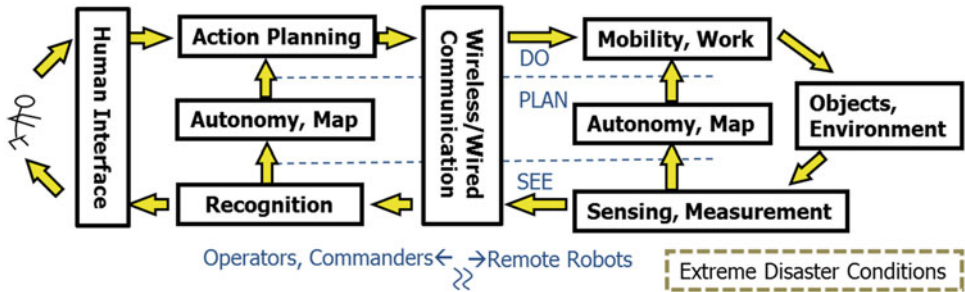
performance of autonomy (robot intelligence, multiagent collaboration, etc.), issues of wireless and wired communication (instability, delay, capacity, tether handling, etc.), operators' limitations (situation awareness, decision ability, fatigue, mistake, etc.), basic performances (explosion proof, weight, durability, portability, etc.), and system integration that combines the components into the solution. Mission critical planning and execution including human factors, training, role sharing, logistics, etc. have to be considered at the same time.

Research into systems and control is expected to solve the abovementioned challenges of components and systems. For example, intelligent control is essential for mobility and workability under extreme conditions, control of feedback systems including long delay and dynamic instability, control of human-in-loop systems, and system integration of heterogeneous systems which are important research topics of systems and control.

In the research field of disaster robotics, various competitions of practical robots have been held, e.g., RoboCupRescue targeting CBRNE disasters, ELROB and euRathlon for field activities, MAGIC for multi-robot autonomy, DARPA Robotics Challenge for humanoid in nuclear disasters, and World Robot Summit Disaster Robotics Category for plant and tunnel disasters. They challenge to solve the abovementioned technical issues under different themes with providing practical test bed of advanced technologies.



Disaster Response Robot, Fig. 10 PMORPH for investigation in the pressure containment vessels of Fukushima-Daiichi nuclear power plant. (By courtesy of Hitachi)



Disaster Response Robot, Fig. 11 Data flow of remotely controlled disaster response robots

Cross-References

- [Robot Motion Control](#)
- [Robot Teleoperation](#)
- [Underwater Robots](#)
- [Wheeled Robots](#)

Bibliography

- ELROB (2013). www.elrob.org
 Group INTRA (2013). www.groupe-intra.com
 KHG (2013). www.khgmbh.de
 Murphy R (2014) Disaster robotics. MIT Press
 PMORPH (2019). social-innovation.hitachi/en-au/case-studies/pmorph.interview/
 RoboCup (2013). www.robocup.org
 Siciliano B, Khatib O (eds) (2008) Springer handbook of robotics, 1st edn. Springer
 Siciliano B, Khatib O (eds) (2014) Springer handbook of robotics, 2nd edn. Springer
 Tadokoro S (ed) (2010) Rescue robotics, DDT project on robots and systems for urban search and rescue. Springer, New York
 Tadokoro S (ed) (2019) Disaster robotics, results from the ImPACT tough robotics challenge. Springer, Cham
 Tadokoro S, Seki S, Asama H (2013) Priority issues of disaster robotics in Japan. In: Proceedings of IEEE region 10 humanitarian technology conference, Sendai, 27–29 Aug 2013
 Tadokoro S et al. (2019) The World Robot Summit Disaster Robotics Category - Achievements of the 2018 Preliminary Competition, Advanced Robotics, 33(17):854–875

Discrete Event Systems and Hybrid Systems, Connections Between

Alessandro Giua
 DIEE, University of Cagliari, Cagliari, Italy
 LSIS, Aix-en-Provence, France

Abstract

The causes of the complex behavior typical of hybrid systems are multifarious and are commonly explained in the literature using paradigms that are mainly focused on the connections between time-driven and hybrid

systems. In this entry, we recall some of these paradigms and further explore the connections between discrete event and hybrid systems from other perspectives. In particular, the role of abstraction in passing from a hybrid model to a discrete event one and vice versa is discussed.

Keywords

Hybrid system · Logical discrete event system · Timed discrete event system

Introduction

Hybrid systems combine the dynamics of both *time-driven systems* and *discrete event systems*.

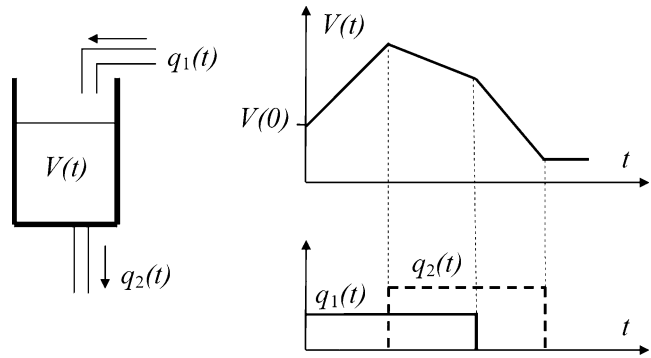
The evolution of a *time-driven system* can be described by a differential equation (in continuous time) or by a difference equation (in discrete time). An example of such a system is the tank shown in Fig. 1 whose behavior, assuming the tank is not full, is ruled in continuous time t by the differential equation

$$\frac{d}{dt}V(t) = q_1(t) - q_2(t)$$

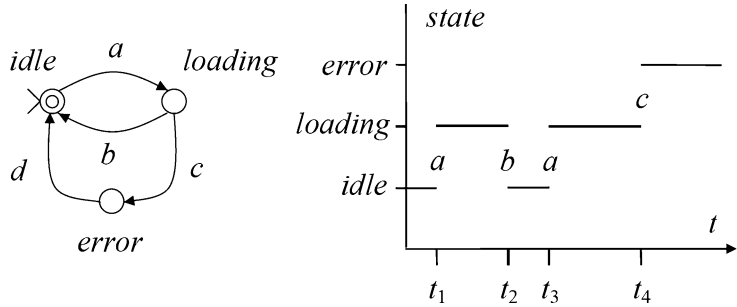
where V is the volume of liquid and q_1 and q_2 are, respectively, the input and output flow.

A *discrete event system* (Lafortune and Cassandras 2007; Seatzu et al. 2012) evolves in accordance with the abrupt occurrence, at possibly unknown irregular intervals, of physical events. Its states may have logical or symbolic, rather than numerical, values that change in response to events which may also be described in nonnumerical terms. An example of such a system is a robot that loads parts on a conveyor, whose behavior is described by the automaton in Fig. 2. The robot can be “idle,” “loading” a part, or in an “error” state when a part is incorrectly positioned. The events that drive its evolution are a (grasp a part), b (part correctly loaded), c (part incorrectly positioned), and d (part repositioned). In a *logical* discrete event system (► [Supervisory Control of Discrete-Event Systems](#)), the timing of event occurrences are ignored, while in a *timed*

Discrete Event Systems and Hybrid Systems, Connections Between,
Fig. 1 A tank



Discrete Event Systems and Hybrid Systems, Connections Between,
Fig. 2 A machine with failures



discrete event system (► [Models for Discrete Event Systems: An Overview](#)), they are described by means of a suitable timing structure.

In a *hybrid system* (► [Hybrid Dynamical Systems, Feedback Control of](#)), time-driven and event-driven evolutions are simultaneously present and mutually dependent. As an example, consider a room where a thermostat maintains the temperature $x(t)$ between $x_a = 20^\circ\text{C}$ and $x_b = 22^\circ\text{C}$ by turning a heat pump on and off. Due to the exchange with the external environment at temperature $x_e \ll x(t)$, when the pump is off, the room temperature derivative is

$$\frac{d}{dt}x(t) = -k[x(t) - x_e]$$

where k is a suitable coefficient, while when the pump is on, the room temperature derivative is

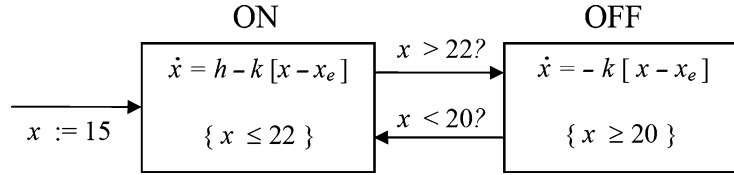
$$\frac{d}{dt}x(t) = h(t) - k[x(t) - x_e]$$

where the positive term $h(t)$ is due to the heat pump. The hybrid automaton that describes this system is shown in Fig. 3.

The causes of the complex behavior typical of hybrid systems are multifarious, and among the paradigms commonly used in the literature to describe them, we mention three.

- *Logically controlled systems.* Often, a physical system with a time-driven evolution is controlled in a feedback loop by means of a controller that implements discrete computations and event-based logic. This is the case of the thermostat mentioned above. Classes of systems that can be described by this paradigm are *embedded systems* or, when the feedback loop is closed through a communication network, *cyber-physical systems*.
- *State-dependent mode of operation.* A time-driven system can have different modes of evolution depending on its current state. As an example, consider a bouncing ball. While the ball is above the ground (vertical position $h > 0$) its behavior is that of a falling body subject to a constant gravitational force. However, when the ball collides with the ground (vertical position $h = 0$), its behavior is that of a (partially) elastic body that bounces up.

Discrete Event Systems and Hybrid Systems, Connections Between, Fig. 3 Hybrid automaton of the thermostat



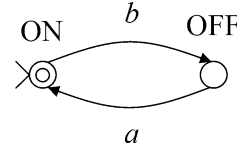
Classes of systems that can be described by this paradigm are *piecewise affine systems* and *linear complementarity system*.

- *Variable structure systems*. Some systems may change their structure assuming different configuration, each characterized by a different behavior. As an example, consider a multicell voltage converter composed by a cascade of elementary commutation cells: controlling some switches, it is possible to insert or remove cells so as to produce a desired output voltage signal. Classes of systems that can be described by this paradigm are *switched systems*.

While these are certainly appropriate and meaningful paradigms, they are mainly focused on the connections between time-driven and hybrid systems. In the rest of this entry, we will discuss the connections between discrete event and hybrid systems from other different perspectives. The focus is strictly on modeling, thus approaches for analysis or control will not be discussed.

From Hybrid Systems to Discrete Event System by Modeling Abstraction

A *system* is a physical object, while a *model* is a (more or less accurate) mathematical description of its behavior that captures those features that are deemed mostly significant. In the previous pages, we have introduced different classes of systems, such as “time-driven systems,” “discrete event systems,” and “hybrid systems,” but properly speaking, this taxonomy pertains to the models because the terms “time driven,” “discrete event,” or “hybrid” should be used to classify the mathematical description and not the physical object.



Discrete Event Systems and Hybrid Systems, Connections Between, Fig. 4 Logical discrete event model of the thermostat

According to this view, a discrete event model is often perceived as a high-level description of a physical system where the time-driven dynamics are ignored or, at best, approximated by a timing structure. This procedure to derive a simpler model in a way that preserves the properties being analyzed while hiding the details that are of no interest is called *abstraction* (Alur et al. 2000).

Consider, as an example, the thermostat in Fig. 3. In such a system, the time-driven evolution determines a change in the temperature, which in turn – reaching a threshold – triggers the occurrence of an event that changes the discrete state. Assume one does not care about the exact form this triggering mechanism takes and is only interested in determining if the heat pump is turned on or off. In such a case, we can completely *abstract* the time-driven evolution obtaining a logical discrete event model such as the automaton in Fig. 4, where label *a* denotes the event *the temperature drops below 20°C* and label *b* denotes the event *the temperature raises over 22°C*.

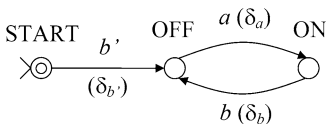
For some purposes, e.g., to determine the utilization rate of the heat pump and thus its operating cost, the model in Fig. 4 is inadequate. In such a case, one can consider a less coarse abstraction of the hybrid model in Fig. 3 obtaining a timed discrete event model such as the automaton in Fig. 4. Here, to each event is associated a firing delay: as an example, δ_a represents the time it takes – when the pump is

off – to cool down until the lower temperature threshold is reached and event a occurs. The delay may be a deterministic value or even a random one to take into account the uncertainty due to non-modeled time-varying parameters such as the temperature of the external environment. Note that a new state (START) and a new event b' have now been introduced to capture the transient phase in which the room temperature, from the initial value $x(0) = 15^\circ\text{C}$, reaches the higher temperature threshold: in fact, event b' has a delay greater than the delay of event b .

Timed Discrete Event Systems Are Hybrid Systems

Properly speaking, all timed discrete event systems may also be seen as hybrid systems if one considers the dynamics of the timers – that specify the event occurrence – as elementary time-driven evolutions. In fact, the simplest model of hybrid systems is the *timed automaton* introduced by Alur and Dill (1994) whose main feature is the fact that each continuous variable $x(t)$ has a constant derivative $\dot{x}(t) = 1$ and thus can only describe the passage of time. Incidentally, we note that the term “timed automaton” is also used in the area of discrete event systems (Lafortune and Cassandras 2007) to denote an automaton in which a timing structure is associated to the events: such an example was shown in Fig. 5. To avoid any confusion, in the following, we denote the former model *Alur-Dill automaton* and the latter model *timed DES automaton*.

In Fig. 6 is shown an Alur-Dill automaton that describes the thermostat, where the time-driven dynamics have been abstracted and only the timing of event occurrence is modeled as in



Discrete Event Systems and Hybrid Systems, Connections Between, Fig. 5 Timed discrete event model of the thermostat

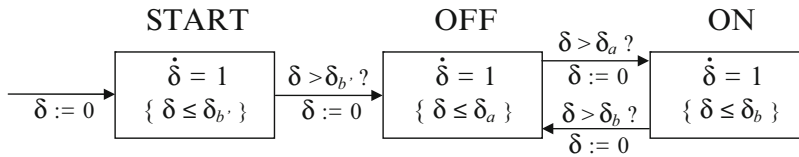
the timed DES automaton in Fig. 5. The only continuous variable is the value of a timer δ : when it goes beyond a certain threshold (e.g., $\delta > \delta_a$), an event occurs (e.g., event a) changing the discrete state (e.g., from OFF to ON) and resetting the timer to zero.

It is rather obvious that the behavior of the Alur-Dill automaton in Fig. 6 is equivalent to the behavior of timed DES automaton in Fig. 5. In the former model, the notion of time is encoded by means of an explicit continuous variable δ . In the latter model, the notion of time is implicitly encoded by the timer that during an evolution will be associated to each event. In both cases, however, the overall state of the systems is described by a pair $(\ell(t), x(t))$ where the first element ℓ takes value in a discrete set $\{START, ON, OFF\}$ and the second element is a vector (in this particular case with a single component) of timer valuations.

It should be pointed out that an Alur-Dill automaton may have a more complex structure than that shown in Fig. 6: as an example, the guard associated to a transition, i.e., the values of the timer that enable it, can be an arbitrary rectangular set. However, the same is also true for a timed discrete event system: several policies can be used to define the time intervals enabling an event (enabling policy) or to specify when a timer is reset (memory policy) (Ajmone Marsan et al. 1995). Furthermore, timed discrete event system can have arbitrary stochastic timing structures (e.g., semi-Markovian processes, Markov chains, and queuing networks (Lafortune and Cassandras 2007)), not to mention the possibility of having an infinite discrete state space (e.g., timed Petri nets (Ajmone Marsan et al. 1995; David and Alla 2004)). As a result, we can say that timed DES automata are far more general than Alur-Dill automata and represent a meaningful subclass of hybrid systems.

From Discrete Event System to Hybrid Systems by Fluidization

The computational complexity involved in the analysis and optimization of real-scale problems



Discrete Event Systems and Hybrid Systems, Connections Between, Fig. 6 Alur-Dill automaton of the thermostat

often becomes intractable with discrete event models due to the very large number of reachable states, and a technique that has shown to be effective in reducing this complexity is called *fluidization* (► [Applications of Discrete Event Systems](#)). It should be noted that the derivation of a fluid (i.e., hybrid) model from a discrete event one is yet an example of abstraction albeit going in opposite direction with respect to the examples discussed in the section “[From Hybrid Systems to Discrete Event System by Modeling Abstraction](#)” above.

The main drive that motivated the fluidization approach derives from the observation that some discrete event systems are “heavily populated” in the sense that there are many identical items in some component (e.g., clients in a queue). Fluidization consists in replacing the integer counter of the number of items by a real number and in approximating the “fast” discrete event dynamics that describe how the counter changes by a continuous dynamics. This approach has been successfully used to study the performance optimization of fluid-queuing networks (Cassandras and Lygeros 2006) or Petri net models (Balduzzi et al. 2000; David and Alla 2004; Silva and Recalde 2004) with applications in domains such as manufacturing systems and communication networks. We also remark that in general, different fluid approximations are necessary to describe the same system, depending on its discrete state, e.g., in the manufacturing domain, machines working or down, buffers full or empty, and so on. Thus, the resulting model can be better described as a hybrid model, where different time-driven dynamics are associated to different discrete states.

There are many advantages in using fluid approximations. First, there is the possibility of considerable increase in computational efficiency

because the simulation of a fluid model can often be performed much faster than that of its discrete event counterpart. Second, fluid approximations provide an aggregate formulation to deal with complex systems, thus reducing the dimension of the state space. Third, the resulting simple structures often allow explicit computation of performance measures. Finally, some design parameters in fluid models are continuous; hence, it is possible to use gradient information to speed up optimization and to perform sensitivity analysis (Balduzzi et al. 2000); in many cases, it has also been shown that fluid approximations do not introduce significant errors when carrying out performance analysis via simulation (► [Perturbation Analysis of Discrete Event Systems](#)).

Cross-References

- [Applications of Discrete Event Systems](#)
- [Hybrid Dynamical Systems, Feedback Control of](#)
- [Models for Discrete Event Systems: An Overview](#)
- [Perturbation Analysis of Discrete Event Systems](#)
- [Supervisory Control of Discrete-Event Systems](#)

Bibliography

- Ajmoné Marsan M, Balbo G, Conte G, Donatelli S, Franceschinis G (1995) Modelling with generalized stochastic Petri nets. Wiley, Chichester/New York
- Alur R, Dill DL (1994) A theory of timed automata. Theor Comput Sci 126:183–235
- Alur R, Henzinger TA, Lafferriere G, Pappas GJ (2000) Discrete abstractions of hybrid systems. Proc IEEE 88(7):971–984

- Balduzzi F, Giua A, Menga G (2000) First-order hybrid Petri nets: a model for optimization and control. *IEEE Trans Robot Autom* 16:382–399
- Cassandras CG, Lygeros J (eds) (2006) Stochastic hybrid systems: recent developments and research. CRC Press, New York
- David R, Alla H (2004) Discrete, continuous and hybrid Petri nets. Springer, Berlin
- Lafortune S, Cassandras CG (2007) Introduction to discrete event systems, 2nd edn. Springer, Boston
- Seatzu C, Silva M, van Schuppen JH (eds) (2012) Control of discrete event systems. Automata and Petri net perspectives. Volume 433 of lecture notes in control and information science. Springer, London
- Silva M, Recalde L (2004) On fluidification of Petri net models: from discrete to hybrid and continuous models. *Annu Rev Control* 28(2):253–266

Discrete Optimal Control

David Martin De Diego
 Instituto de Ciencias Matemáticas
 (CSIC-UAM-UC3M-UCM), Madrid, Spain

Abstract

Discrete optimal control is a branch of mathematics which studies optimization procedures for controlled discrete-time models – that is, the optimization of a performance index associated with a discrete-time control system. This entry gives an introduction to the topic. The formulation of a general discrete optimal control problem is described, and applications to mechanical systems are discussed.

Keywords

Discrete mechanics · Discrete-time models · Symplectic integrators · Variational integrators

Synonyms

DOC

Definition

Discrete optimal control is a branch of mathematics which studies optimization procedures for controlled discrete-time models, that is, the optimization of a performance index associated to a discrete-time control system.

Motivation

Optimal control theory is a mathematical discipline with innumerable applications in both science and engineering. Discrete optimal control is concerned with control optimization for discrete-time models. Recently, in discrete optimal control theory, a great interest has appeared in developing numerical methods to optimally control real mechanical systems, as for instance, autonomous robotic vehicles in natural environments such as robotic arms, spacecrafts, or underwater vehicles.

During the last years, a huge effort has been made for the comprehension of the fundamental geometric structures appearing in dynamical systems, including control systems and optimal control systems. This new geometric understanding of those systems has made possible the construction of suitable numerical techniques for integration. A collection of ad hoc numerical methods are available for both dynamical and control systems. These methods have grown up accordingly with the needs in research coming from different fields such as physics and engineering. However, a new breed of ideas in numerical analysis has started recently. They incorporate the geometry of the systems into the analysis and that allows faster and more accurate algorithms and with less spurious effects than the traditional ones. All this gives birth to a new field called Geometric Integration (Hairer et al. 2002). For instance, numerical integrators for Hamiltonian systems should preserve the symplectic structure underlying the geometry of the system. If so, they are called symplectic integrators.

Another approach used by more and more authors is based on the theory of discrete mechanics and variational integrators to obtain geometric integrators preserving some of the

geometry of the original system (Marsden and West 2001; Hussein et al. 2006; Wendlandt and Marsden 1997a, b) (see also the section “Discrete Mechanics”). These geometric integrators are easily adapted and applied to a wide range of mechanical systems: forced or dissipative systems, holonomically constrained systems, explicitly time-dependent systems, reduced systems with frictional contact, nonholonomic dynamics, and multisymplectic field theories, among others.

As before, in optimal control theory, it is necessary to distinguish two kinds of numerical methods: the so-called direct and indirect methods. If we use direct methods, we first discretize the state and control variables, control equations, and cost functional, and then we solve a nonlinear optimization problem with constraints given by the discrete control equations, additional constraints, and boundary conditions (Bock and Plitt 1984; Bonnans and Laurent-Varin 2006; Hager 2001; Pytlak 1999). In this case, we typically need to solve a system of the type (see the section “Formulation of a General Discrete Optimal Control Problem”)

$$\begin{cases} \text{minimize } F(X) \\ \text{with } \Psi(X) = 0 \\ \Phi(X) \geq 0 \end{cases} \quad X = (q^0, \dots, q^N, u_1, \dots, u_N)$$

On the other hand, indirect methods consist of solving numerically the boundary value problem obtained from the equations after applying Pontryagin’s Maximum Principle.

The combination of direct methods and discrete mechanics allows to obtain numerical control algorithms which are geometric structure preserving and exhibit a good long-time behavior (Bloch et al. 2013; Junge and Ober-Blöbaum 2005; Junge et al. 2006; Kobilarov 2008; Jiménez et al. 2013; Leyendecker et al. 2007; Ober-Blöbaum 2008; Ober-Blöbaum et al. 2011). Furthermore, it is possible to adapt many of the techniques used for continuous control mechanical systems to the design of quantitative and qualitative accurate numerical methods for optimal control methods (reduction by

symmetries, preservation of geometric structures, Lie group methods, etc.).

Formulation of a General Discrete Optimal Control Problem

Let M be an n -dimensional manifold, x denote the state variables in M for an agent’s environment, and $u \in U \subset \mathbb{R}^m$ be the control or action that the agent chooses to accomplish a task or objective. Let $f_d(x, u) \in M$ be the resulting state after applying the control u to the state x . For instance, x may be the configuration of a vehicle at time t and u its fuel consumption, and then $f_d(x, u)$ is the new configuration of the vehicle at time $t + h$, with $t, h > 0$. Of course, we want to minimize the fuel consumption. Hence, the optimal control problem consists of finding the cheapest way to move the system from a given initial position to a final state. The problem can be mathematically described as follows: find a sequence of controls $(u_0, u_1, \dots, u_{N-1})$ and a sequence of states $(x_0, x_1, \dots, x_N)$ such that

$$x_{k+1} = f_d(k, x_k, u_k), \quad (1)$$

where $x_k \in M, u_k \in U$, and the total cost

$$C_d = \sum_{k=0}^{N-1} C_d(k, x_k, u_k) + \phi_d(N, x_N) \quad (2)$$

is minimized where ϕ_d is a function of the final time and state at the final time (the terminal payoff) and C_d is a function depending on the discrete time, the state, and the control at each intermediate discrete time k (the running payoff).

To solve the discrete optimal control problem determined by Eqs. (1) and (2), it is possible to use the classical Lagrangian multiplier approach. In this case, we consider the control equations (1) as constraint equations associating a Lagrange multiplier to each constraint. Assume for simplicity that $M = \mathbb{R}^n$. Then, we construct the augmented cost function

$$\tilde{C}_d = \sum_{k=0}^{N-1} \left[p_{k+1} (x_{k+1} - f_d(k, x_k, u_k)) - C_d(k, x_k, u_k) \right] - \Phi_d(N, x_N) \quad (3)$$

where $p_k \in \mathbb{R}^n$, $k = 1, \dots, N$, are considered as the Lagrange multipliers. The notation xy is used for the scalar (inner) product $x \cdot y$ of two vectors in $\mathbb{R}^n$.

From the *pseudo-Hamiltonian function*

$$H_d(k, x_k, p_{k+1}, u_k) = p_{k+1} f_d(k, x_k, u_k) - C_d(k, x_k, u_k),$$

we deduce the necessary conditions for a constrained minimum:

$$\begin{aligned} x_{k+1} &= \frac{\partial H_d}{\partial p}(k, x_k, p_{k+1}, u_k) \\ &= f_d(k, x_k, u_k) \end{aligned} \quad (4)$$

$$\begin{aligned} p_k &= \frac{\partial H_d}{\partial q}(k, x_k, p_{k+1}, u_k) \\ &= p_{k+1} \frac{\partial f_d}{\partial q}(k, x_k, u_k) \\ &\quad - \frac{\partial C_d}{\partial q}(k, x_k, u_k) \end{aligned} \quad (5)$$

$$\begin{aligned} 0 &= \frac{\partial H_d}{\partial u}(k, x_k, p_{k+1}, u_k) \\ &= p_{k+1} \frac{\partial f_d}{\partial u}(k, x_k, u_k) \\ &\quad - \frac{\partial C_d}{\partial u}(k, x_k, u_k) \end{aligned} \quad (6)$$

where $0 \leq k \leq N-1$. Moreover, we have some boundary conditions

$$x_0 \text{ is given and } p_N = -\frac{\partial \Phi_d}{\partial q}(N, x_N) \quad (7)$$

The variable p_k is called the costate of the system and Eq. (5) is called the adjoint equation. Observe that the recursion of x_k given by Eq. (4) develops forward in the discrete time, but the recursion of the costate variable is backward in the discrete time.

In the sequel, it is assumed the following regularity condition:

$$\det \left(\frac{\partial^2 H_d}{\partial u^a \partial u^b} \right) \neq 0$$

where $1 \leq a, b \leq m$, and $(u^a) \in U \subseteq \mathbb{R}^m$. Applying the implicit function theorem, we obtain from Eq. (6) that locally $u_k = g(k, x_k, p_{k+1})$. Defining the function

$$\begin{aligned} \tilde{H}_d : \quad \mathbb{Z} \times \mathbb{R}^{2n} &\longrightarrow \mathbb{R} \\ (k, q_k, p_{k+1}) &\longmapsto H_d(k, q_k, p_{k+1}, u_k) \end{aligned}$$

Equations (4) and (5) are rewritten as the following discrete Hamiltonian system:

$$x_{k+1} = \frac{\partial \tilde{H}_d}{\partial p}(k, x_k, p_{k+1}) \quad (8)$$

$$p_k = \frac{\partial \tilde{H}_d}{\partial q}(k, x_k, p_{k+1}) \quad (9)$$

The expression of the solutions of the optimal control problem as a discrete Hamiltonian system (under some regularity properties) is important since it indicates that the discrete evolution is preserving symplecticity. A simple proof of this fact is the following (de León et al. (2007)). Construct the following function

$$\begin{aligned} G_k(x_k, x_{k+1}, p_{k+1}) &= \tilde{H}_d(k, x_k, p_{k+1}) \\ &\quad - p_{k+1} x_{k+1}, \end{aligned}$$

with $0 \leq k \leq N-1$. For each fixed k :

$$\begin{aligned} dG_k &= \frac{\partial \tilde{H}_d}{\partial q}(k, x_k, p_{k+1}) dx_k \\ &\quad + \frac{\partial \tilde{H}_d}{\partial p}(k, x_k, p_{k+1}) dp_{k+1} \\ &\quad - p_{k+1} dx_{k+1} - x_{k+1} dp_{k+1}. \end{aligned}$$

Thus, along solutions of Eqs. (8) and (9), we have that $dG_k|_{\text{solutions}} = p_k dx_k - p_{k+1} dx_{k+1}$ which implies $dx_k \wedge dp_k = dx_{k+1} \wedge dp_{k+1}$.

In the next section, we will study the case of discrete optimal control of mechanical systems.

First, we will need an introduction to discrete mechanics and variational integrators.

Discrete Mechanics

Let Q be an n -dimensional differentiable manifold with local coordinates (q^i) , $1 \leq i \leq n$. We denote by TQ its tangent bundle with induced coordinates $(q^i, \dot{q}^i)$. Let $L: TQ \rightarrow \mathbb{R}$ be a Lagrangian function; the associated Euler–Lagrange equations are given by

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}^i} \right) - \frac{\partial L}{\partial q^i} = 0, \quad 1 \leq i \leq n. \quad (10)$$

These equations are a system of implicit second-order differential equations. Assume that the Lagrangian is **regular**, that is, the matrix $\left(\frac{\partial^2 L}{\partial \dot{q}^i \partial \dot{q}^j} \right)$ is non-singular. It is well known that the origin of these equations is variational (see Marsden and West 2001). Variational integrators retain this variational character and also some of the key geometric properties of the continuous system, such as symplecticity and momentum conservation (see Hairer et al. 2002 and references therein). In the following, we summarize the main features of this type of numerical integrators (Marsden and West 2001). A **discrete Lagrangian** is a map $L_d: Q \times Q \rightarrow \mathbb{R}$, which may be considered as an approximation of the integral action defined by a continuous Lagrangian $L: TQ \rightarrow \mathbb{R}$: $L_d(q_0, q_1) \approx \int_0^h L(q(t), \dot{q}(t)) dt$ where $q(t)$ is a solution of the Euler–Lagrange equations for L with $q(0) = q_0$, $q(h) = q_1$, and $h > 0$ being enough small.

Remark 1 The Cartesian product $Q \times Q$ is equipped with an interesting differential structure, called Lie groupoid, which allows the extension of variational calculus to more general settings (see Marrero et al. 2006, 2010 for more details).

Define the **action sum** $S_d: Q^{N+1} \rightarrow \mathbb{R}$, corresponding to the Lagrangian L_d by $S_d = \sum_{k=1}^N L_d(q_{k-1}, q_k)$, where $q_k \in Q$ for $0 \leq k \leq N$ and N is the number of steps. The discrete

variational principle states that the solutions of the discrete system determined by L_d must extremize the action sum given fixed endpoints q_0 and q_N . By extremizing S_d over q_k , $1 \leq k \leq N-1$, we obtain the system of difference equations

$$D_1 L_d(q_k, q_{k+1}) + D_2 L_d(q_{k-1}, q_k) = 0, \quad (11)$$

or, in coordinates,

$$\frac{\partial L_d}{\partial x^i}(q_k, q_{k+1}) + \frac{\partial L_d}{\partial y^i}(q_{k-1}, q_k) = 0,$$

where $1 \leq i \leq n$, $1 \leq k \leq N-1$, and x, y denote the n -first and n -second variables of the function L_d , respectively.

These equations are usually called the **discrete Euler–Lagrange equations**. Under some regularity hypotheses (the matrix $D_{12} L_d(q_k, q_{k+1})$ is regular), it is possible to define a (local) discrete flow $\Upsilon_{L_d}: Q \times Q \rightarrow Q \times Q$, by $\Upsilon_{L_d}(q_{k-1}, q_k) = (q_k, q_{k+1})$ from (11). Define the discrete Legendre transformations associated to L_d as

$$\begin{aligned} \mathbb{F}^- L_d: Q \times Q &\rightarrow T^*Q \\ (q_0, q_1) &\mapsto (q_0, -D_1 L_d(q_0, q_1)), \\ \mathbb{F}^+ L_d: Q \times Q &\rightarrow T^*Q \\ (q_0, q_1) &\mapsto (q_1, D_2 L_d(q_0, q_1)), \end{aligned}$$

and the discrete Poincaré–Cartan 2-form $\omega_d = (\mathbb{F}^+ L_d)^* \omega_Q = (\mathbb{F}^- L_d)^* \omega_Q$, where ω_Q is the canonical symplectic form on T^*Q . The discrete algorithm determined by Υ_{L_d} preserves the symplectic form ω_d , i.e., $\Upsilon_{L_d}^* \omega_d = \omega_d$. Moreover, if the discrete Lagrangian is invariant under the diagonal action of a Lie group G , then the discrete momentum map $J_d: Q \times Q \rightarrow \mathfrak{g}^*$ defined by

$$\begin{aligned} \langle J_d(q_k, q_{k+1}), \xi \rangle &= \langle D_2 L_d(q_k, q_{k+1}), \\ &\quad \xi_Q(q_{k+1}) \rangle \end{aligned}$$

is preserved by the discrete flow. Therefore, these integrators are symplectic-momentum preserving. Here, ξ_Q denotes the fundamental vector field determined by $\xi \in \mathfrak{g}$, where $\mathfrak{g}$ is the Lie

algebra of G . As stated in Marsden and West (2001), discrete mechanics is inspired by discrete formulations of optimal control problems (see Cadzow 1970; Jordan and Polak 1964; Hwang and Fan 1967).

Discrete Optimal Control of Mechanical Systems

Consider a mechanical system whose configuration space is an n -dimensional differentiable manifold Q and whose dynamics is determined by a Lagrangian $L : TQ \rightarrow \mathbb{R}$. The control forces are modeled as a mapping $f : TQ \times U \rightarrow T^*Q$, where $f(v_q, u) \in T_q^*Q$, $v_q \in T_qQ$ and $u \in U$, being U the control space. Observe that this last definition also covers configuration and velocity-dependent forces such as dissipation or friction (see Ober-Blobaum et al. 2011).

The motion of the mechanical system is described by applying the principle of **Lagrange–d’Alembert**, which requires that the solutions $q(t) \in Q$ must satisfy

$$\begin{aligned} \delta \int_0^T L(q(t), \dot{q}(t)) dt \\ + \int_0^T f(q(t), \dot{q}(t), u(t)) \delta q(t) dt = 0, \end{aligned} \quad (12)$$

where $(q, \dot{q})$ are the local coordinates of TQ and where we consider arbitrary variations $\delta q(t) \in T_{q(t)}Q$ with $\delta q(0) = 0$ and $\delta q(T) = 0$ (since we are prescribing fixed initial and final conditions $(q(0), \dot{q}(0))$ and $(q(T), \dot{q}(T))$).

As we consider an optimal control problem, the forces f must be chosen, if they exist, as the ones that extremize the **cost functional**:

$$\begin{aligned} \int_0^T C(q(t), \dot{q}(t), u(t)) dt \\ + \Phi(q(T), \dot{q}(T), u(T)), \end{aligned} \quad (13)$$

where $C : TQ \times U \rightarrow \mathbb{R}$.

The optimal equations of motion can now be derived using Pontryagin’s Maximum Principle.

In general, it is not possible to explicitly integrate these equations. Then, it is necessary to apply a numerical method. In this work, using discrete variational techniques, we first discretize the Lagrange–d’Alembert principle and then the cost functional. We obtain a numerical method that preserves some geometric features of the original continuous system as described in the sequel.

Discretization of the Lagrangian and Control Forces

To discretize this problem, we replace the tangent space TQ by the Cartesian product $Q \times Q$ and the continuous curves by sequences $q_0, q_1, \dots, q_N$ (we are using N steps, with time step h fixed, in such a way $t_k = kh$ and $Nh = T$). The discrete Lagrangian $L_d : Q \times Q \rightarrow \mathbb{R}$ is constructed as an approximation of the action integral in a single time step (see Marsden and West 2001), that is,

$$L_d(q_k, q_{k+1}) \approx \int_{kh}^{(k+1)h} L(q(t), \dot{q}(t)) dt.$$

We choose the following discretization for the external forces: $f_d^\pm : Q \times Q \times U \rightarrow T^*Q$, where $U \subset \mathbb{R}^m$, $m \leq n$, such that

$$\begin{aligned} f_d^-(q_k, q_{k+1}, u_k) &\in T_{q_k}^*Q, \\ f_d^+(q_k, q_{k+1}, u_k) &\in T_{q_{k+1}}^*Q. \end{aligned}$$

f_d^+ and f_d^- are right and left discrete forces (see Ober-Blobaum et al. 2011).

Discrete Lagrange–d’Alembert Principle

Given such forces, we define the discrete Lagrange–d’Alembert principle, which seeks sequences $\{q_k\}_{k=0}^N$ that satisfy

$$\begin{aligned} \delta \sum_{k=0}^{N-1} L_d(q_k, q_{k+1}) \\ + \sum_{k=0}^{N-1} \left(f_d^-(q_k, q_{k+1}, u_k) \delta q_k \right. \\ \left. + f_d^+(q_k, q_{k+1}, u_k) \delta q_{k+1} \right) = 0, \end{aligned}$$

for arbitrary variations $\{\delta q_k\}_{k=0}^N$ with $\delta q_0 = \delta q_N = 0$. After some straightforward manipulations, we arrive to the forced discrete Euler–Lagrange equations

$$\begin{aligned} D_2 L_d(q_{k-1}, q_k) + D_1 L_d(q_k, q_{k+1}) \\ + f_d^+(q_{k-1}, q_k, u_{k-1}) \\ + f_d^-(q_k, q_{k+1}, u_k) = 0, \end{aligned} \quad (14)$$

with $k = 1, \dots, N-1$.

Boundary Conditions

For simplicity, we assume that the boundary conditions of the continuous optimal control problem are given by $q(0) = x_0$, $\dot{q}(0) = v_0$, $q(T) = x_T$, $\dot{q}(T) = v_T$. To incorporate these conditions to the discrete setting, we use both the continuous and discrete Legendre transformations. From Marsden and West (2001), given a forced system, we can define the discrete momenta

$$\begin{aligned} \mu_k &= -D_1 L_d(q_k, q_{k+1}) - f_d^-(q_k, q_{k+1}, u_k), \\ \mu_{k+1} &= D_2 L_d(q_k, q_{k+1}) + f_d^+(q_k, q_{k+1}, u_k). \end{aligned}$$

From the continuous Lagrangian, we have the momenta

$$\begin{aligned} p_0 &= \mathbb{F}L(x_0, v_0) = (x_0, \frac{\partial L}{\partial v}(x_0, v_0)) \\ p_T &= \mathbb{F}L(x_T, v_T) = \left(x_T, \frac{\partial L}{\partial v}(x_T, v_T)\right) \end{aligned}$$

Therefore, the natural choice of boundary conditions is

$$\begin{aligned} x_0 &= q_0, \quad x_T = q_N \\ \mathbb{F}L(x_0, v_0) &= -D_1 L_d(q_0, q_1) - f_d^-(q_0, q_1, u_0) \\ \mathbb{F}L(x_T, v_T) &= D_2 L_d(q_{N-1}, q_N) \\ &\quad + f_d^+(q_{N-1}, q_N, u_{N-1}) \end{aligned}$$

that we add to the discrete optimal control problem.

Discrete Cost Function

We can also approximate the cost functional (13) in a single time step h by

$$\begin{aligned} C_d(q_k, q_{k+1}, u_k) \\ \approx \int_{kh}^{(k+1)h} C(q(t), \dot{q}(t), u(t)) dt, \end{aligned}$$

yielding the **discrete cost functional**:

$$\sum_{k=0}^{N-1} C_d(q_k, q_{k+1}, u_k) + \Phi_d(q_{N-1}, q_N, u_{N-1}).$$

Discrete Optimal Control Problem

With all these elements, we have the following discrete optimal control problem:

$$\begin{aligned} \min \sum_{k=0}^{N-1} C_d(q_k, q_{k+1}, u_k) \\ + \Phi_d(q_{N-1}, q_N, u_{N-1}) \end{aligned}$$

subject to

$$\begin{aligned} D_2 L_d(q_{k-1}, q_k) + D_1 L_d(q_k, q_{k+1}) \\ + f_d^+(q_{k-1}, q_k, u_{k-1}) + f_d^-(q_k, q_{k+1}, u_k) = 0, \\ x_0 = q_0, \quad x_T = q_N \\ \frac{\partial L}{\partial v}(x_0, v_0) = -D_1 L_d(q_0, q_1) - f_d^-(q_0, q_1, u_0) \\ \frac{\partial L}{\partial v}(x_T, v_T) = D_2 L_d(q_{N-1}, q_N) \\ + f_d^+(q_{N-1}, q_N, u_{N-1}) \end{aligned}$$

with $k = 1, \dots, N-1$ (see Jiménez et al. 2013; Jiménez and Martín de Diego 2010 for a modification of these equations admitting piecewise controls).

The system now is a constrained nonlinear optimization problem, that is, it corresponds to the minimization of a function subject to algebraic constraints. The necessary conditions for optimality are derived applying nonlinear programming optimization. For the concrete implementation, it is possible to use sequential quadratic programming (SQP) methods to numerically solve the nonlinear optimization problem (Ober-Blöbaum et al. 2011).

Optimal Control Systems with Symmetries

In many interesting cases, the continuous optimal control of a mechanical system is defined on a Lie group and the Lagrangian, cost function, control forces are invariant under the group action. The goal is again the same as in the previous section, that is, to move the system from its current state to a desired state in an optimal way. In this particular case, it is possible to adapt the contents of the section “[Discrete Mechanics](#)” in a similar way to the continuous case (from the standard Euler–Lagrange equations to the Euler–Poincaré equations) and to produce the so-called Lie group variational integrators. These methods preserve the Lie group structure avoiding the use of local charts, projections, or constraints. Based on these methods, for the case of controlled mechanical systems, we produce the discrete Euler–Poincaré equations with controls and the discrete cost function. Consequently, it is possible to deduce necessary optimality conditions for this class of invariant systems (see Bloch et al. 2009; Bou-Rabee and Marsden 2009; Kobilarov and Marsden 2011; Hussein et al. 2006).

Cross-References

- [Differential Geometric Methods in Nonlinear Control](#)
- [Numerical Methods for Nonlinear Optimal Control Problems](#)
- [Optimal Control and Mechanics](#)
- [Optimal Control and Pontryagin’s Maximum Principle](#)

Bibliography

- Bock HG, Plitt KJ (1984) A multiple shooting algorithm for direct solution of optimal control problems. In: 9th IFAC world congress, Budapest. Pergamon Press, pp 242–247
- Bloch AM, Hussein I, Leok M, Sanyal AK (2009) Geometric structure-preserving optimal control of a rigid body. *J Dyn Control Syst* 15(3):307–330
- Bloch AM, Crouch PE, Nordkvist N (2013) Continuous and discrete embedded optimal control problems and their application to the analysis of Clebsch optimal control problems and mechanical systems. *J Geom Mech* 5(1):1–38
- Bonnans JF, Laurent-Varin J (2006) Computation of order conditions for symplectic partitioned Runge–Kutta schemes with application to optimal control. *Numer Math* 103:1–10
- Bou-Rabee N, Marsden JE (2009) Hamilton–Pontryagin integrators on Lie groups: introduction and structure-preserving properties. *Found Comput Math* 9(2):197–219
- Cadzow JA (1970) Discrete calculus of variations. *Int J Control* 11:393–407
- de León M, Martín de Diego D, Santamaría-Merino A (2007) Discrete variational integrators and optimal control theory. *Adv Comput Math* 26(1–3):251–268
- Hager WW (2001) Numerical analysis in optimal control. In: *International series of numerical mathematics*, vol 139. Birkhäuser Verlag, Basel, pp 83–93
- Hairer E, Lubich C, Wanner G (2002) *Geometric numerical integration, structure-preserving algorithms for ordinary differential equations*. Springer series in computational mathematics, vol 31. Springer, Berlin
- Hussein I, Leok M, Sanyal A, Bloch A (2006) A discrete variational integrator for optimal control problems on $SO(3)$. In: *Proceedings of the 45th IEEE conference on decision and control*, San Diego, pp 6636–6641
- Hwang CL, Fan LT (1967) A discrete version of Pontryagin’s maximum principle. *Oper Res* 15:139–146
- Jordan BW, Polak E (1964) Theory of a class of discrete optimal control systems. *J Electron Control* 17:697–711
- Jiménez F, Martín de Diego D (2010) A geometric approach to Discrete mechanics for optimal control theory. In: *Proceedings of the IEEE conference on decision and control*, Atlanta, pp 5426–5431
- Jiménez F, Kobilarov M, Martín de Diego D (2013) Discrete variational optimal control. *J Nonlinear Sci* 23(3):393–426
- Junge O, Ober-Blöbaum S (2005) Optimal reconfiguration of formation flying satellites. In: *IEEE conference on decision and control and European control conference ECC*, Seville
- Junge O, Marsden JE, Ober-Blöbaum S (2006) Optimal reconfiguration of formation flying spacecraft- a decentralized approach. In: *IEEE conference on decision and control and European control conference ECC*, San Diego, pp 5210–5215
- Kobilarov M (2008) *Discrete geometric motion control of autonomous vehicles*. Thesis, Computer Science, University of Southern California
- Kobilarov M, Marsden JE (2011) Discrete geometric optimal control on Lie groups. *IEEE Trans Robot* 27(4):641–655
- Leok M (2004) *Foundations of computational geometric mechanics, control and dynamical systems*. Thesis, California Institute of Technology. Available in <http://www.math.lsa.umich.edu/~mleok>
- Leyendecker S, Ober-Blöbaum S, Marsden JE, Ortiz M (2007) Discrete mechanics and optimal control for constrained multibody dynamics. In: *6th international*

- conference on multibody systems, nonlinear dynamics, and control, ASME international design engineering technical conferences, Las Vegas
- Marrero JC, Martín de Diego D, Martínez E (2006) Discrete Lagrangian and Hamiltonian mechanics on Lie groupoids. *Nonlinearity* 19:1313–1348. Corrigendum: *Nonlinearity* 19
- Marrero JC, Martín de Diego D, Stern A (2010) Lagrangian submanifolds and discrete constrained mechanics on Lie groupoids. Preprint, To appear in *DCDS-A* 35-1, January 2015.
- Marsden JE, West M (2001) Discrete mechanics and variational integrators. *Acta Numer* 10: 357–514
- Marsden JE, Pekarsky S, Shkoller S (1999a) Discrete Euler-Poincaré and Lie-Poisson equations. *Nonlinearity* 12:1647–1662
- Marsden JE, Pekarsky S, Shkoller S (1999b) Symmetry reduction of discrete Lagrangian mechanics on Lie groups. *J Geom Phys* 36(1–2):140–151
- Ober-Blöbaum S (2008) Discrete mechanics and optimal control. Ph.D. Thesis, University of Paderborn
- Ober-Blöbaum S, Junge O, Marsden JE (2011) Discrete mechanics and optimal control: an analysis. *ESAIM Control Optim Calc Var* 17(2):322–352
- Pytlak R (1999) Numerical methods for optimal control problems with state constraints. *Lecture Notes in Mathematics*, 1707. Springer-Verlag, Berlin, xvi+215 pp.
- Wendlandt JM, Marsden JE (1997a) Mechanical integrators derived from a discrete variational principle. *Phys D* 106:2232–2246
- Wendlandt JM, Marsden JE (1997b) Mechanical systems with symmetry, variational principles and integration algorithms. In: Alber M, Hu B, Rosenthal J (eds) *Current and future directions in applied mathematics*, (Notre Dame, IN, 1996), Birkhäuser Boston, Boston, MA, pp 219–261

Distributed Estimation in Networks

Shinkyu Park¹ and Nuno C. Martins²

¹Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, USA

²Department of Electrical and Computer Engineering, Institute for Systems Research, University of Maryland, College Park, MD, USA

Abstract

The entry overviews theory and application of distributed estimation in networks. Distributed estimation research focuses on designing a

network of agents that estimate the state of a large-scale dynamical system, where information exchange between the agents plays a critical role. We discuss algorithms for distributed estimation that define how the agents exchange information and update their state estimates using exchanged information and sensor measurements on the system's state. We review key theoretical results on the algorithm design that find exact conditions under which the state estimate at each agent converges to the system's state and analyze the effect of disturbances in system models, sensing, and communication on estimation performance. To emphasize the importance of the research, we explain key applications of distributed estimation approaches in engineering problems such as distributed control design for maneuvering a vehicle platoon and sensor network design for monitoring wild animal groups.

Keywords

Estimation over networks · Linear time-invariant system · Stability · Robust estimation

Introduction

Distributed estimation research investigates an important problem of designing a network of agents that estimate the state of a large-scale dynamical system. Each agent is equipped with a sensor to collect measurements on the system's state and exchanges information with other agents that are within its communication range. The main goal of the problem is to design a distributed algorithm that defines what information to be exchanged between agents and how each agent estimates the system's state based on exchanged information and sensor measurements.

Unlike the centralized estimator design problem, each agent accesses measurements on a small portion of the system's state; hence without communication with other agents for information

exchange, it cannot fully estimate the system's state. On the other hand, information exchange between agents is constrained by the agents' communication capability, and the state estimation through simple exchange of the measurements, which requires all-to-all communication, would not be feasible.

To address the new challenges, the research in general focuses on the following three components: (1) designing a class of parameterized distributed algorithms that define what information to be exchanged between agents and how each agent updates a state estimate using its own sensor measurements and information from its neighbors, (2) finding conditions under which there is a parameter selection for the algorithm that results in convergence of every agent's estimate to the system's state, and (3) optimizing the parameter selection to attenuate the effect of disturbances in system models, sensing, and communication on estimation performance. In this article, we review some of well-established results in literature related to each component.

As immediate applications, distributed estimation algorithms have been applied in monitoring and controlling large-scale dynamical systems. We discuss two use cases wherein the algorithms are used to design a distributed control system for maneuvering a vehicle platoon and a network of sensing devices for monitoring wild animal groups.

Concise Description of Distributed Estimation

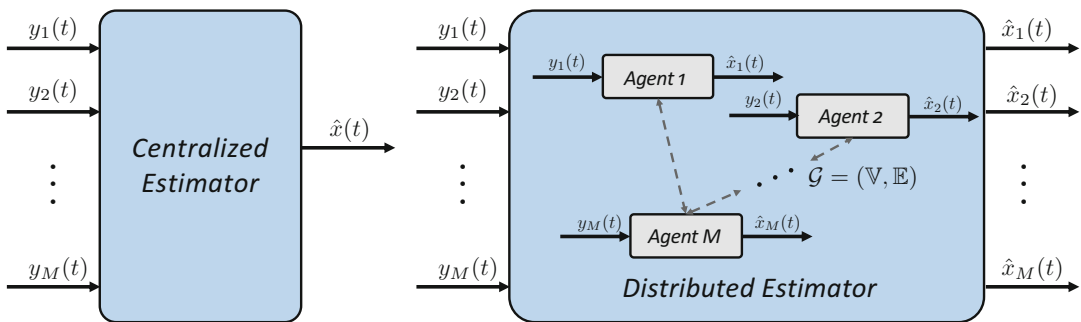
Consider the distributed estimation problem of designing a network of M agents to infer the state of a dynamical system to be monitored and potentially to be controlled. In this section, we review models for the dynamical system and inter-agent communication and an estimation rule that specifies how each agent infers the system's state in the network. Figure 1 illustrates an overview of the distributed estimation and a comparison with the centralized estimation.

We begin by describing the system model using a state-space representation in the continuous-time domain (Some of the distributed estimation approaches reviewed in this entry are designed for discrete-time domain models. However, since techniques developed there are also applicable to the continuous-time domain models (1), for brevity, we only adopt the continuous-time representation.):

$$\dot{x}(t) = Ax(t) + Bu(t) + w_x(t), \quad x(0) = x_0 \quad (1a)$$

$$y_i(t) = C_i x(t) + w_{y_i}(t), \quad i \in \{1, \dots, M\} \quad (1b)$$

The n -dimensional real-valued vector $x(t)$ represents the state of the system, and the p -dimensional real-valued vector $u(t)$ denotes the input to the system. For each i in $\{1, \dots, M\}$, the r_i -dimensional real-valued vector $y_i(t)$ denotes



Distributed Estimation in Networks, Fig. 1 A comparison between the centralized estimation and distributed estimation. Notably, in the distributed estimation, each agent i in the network accesses the measurement $y_i(t)$

on the system's state, communicates with its neighboring agents defined by a given graph $\mathcal{G}$, and computes the state estimate $\hat{x}_i(t)$

the i -th (sensor) measurement that is available to Agent i . The variables $w_x(t)$ and $w_{y_i}(t)$ are disturbance terms representing error in modeling the system dynamics using (1a) and sensor noise associated with the measurement $y_i(t)$ as in (1b), respectively.

The main motivation that prompted the study of distributed estimation is that monitoring and controlling large-scale dynamical systems require multiple networked agents to collect sensor measurements and to share information within the network to estimate the full state of the system, which is of a substantial dimension and cannot be estimated by a single agent. The following are a few examples of such large-scale systems that can be represented by the model (1) (In addition to the examples mentioned in this entry, it is worth mentioning that the model (1) would be used to describe parameter estimation problems in networks (Kar and Moura 2013), where a parameter to be estimated can be defined as the state $x(t)$ of (1) with the selection of $A = I$, $B = 0$, and $w_x(t) = 0$, $t \geq 0$.):

1. Vehicle platoon: To improve transportation efficiency, Levine and Athans (1966) propose the idea of vehicle platooning. A vehicle platoon consists of M vehicles; each needs to be controlled to maintain relative positions with its neighboring vehicles. To model the vehicle platoon using (1), we define $x(t) = (x_1(t), \dots, x_M(t))$ where $x_i(t) = (p_i(t) - p_{i-1}(t), v_i(t))$, with $p_0(t) = 0$, that is, the position of Vehicle i relative to Vehicle $i - 1$ and the velocity of Vehicle i . The sensor measurement $y_i(t)$ in this application is a noisy measurement of $x_i(t)$. The following equations describe how the position $p_i(t)$ and velocity $v_i(t)$ change in response to the control $u_i(t)$ applied to each vehicle and how the measurement $y_i(t)$ is defined: for each i in $\{1, \dots, M\}$,

$$\dot{p}_i(t) = v_i(t) \quad (2a)$$

$$\dot{v}_i(t) = u_i(t) + w_{v_i}(t) \quad (2b)$$

$$y_i(t) = \begin{pmatrix} p_i(t) - p_{i-1}(t) \\ v_i(t) \end{pmatrix} + w_{y_i}(t) \quad (2c)$$

2. Collective mobile agent model: Consider a group of mobile agents, each is described by its position $p_i(t)$ and moving velocity $v_i(t)$ in which the velocity vector $v_i(t)$ of Agent i depends on those of other agents in the same group. To express this model using (1), we define $x(t) = (x_1(t), \dots, x_M(t))$ with $x_i(t) = (p_i(t), v_i(t))$ and $y_i(t)$ as a noisy measurement of $x_i(t)$. In particular, the state-space representations for $p_i(t)$, $v_i(t)$, and $y_i(t)$ are defined as follows: for each i in $\{1, \dots, M\}$,

$$\dot{p}_i(t) = v_i(t) \quad (3a)$$

$$\dot{v}_i(t) = \sum_{j=1}^M l_{ij} v_j(t) + w_{v_i}(t) \quad (3b)$$

$$y_i(t) = \begin{pmatrix} p_i(t) \\ v_i(t) \end{pmatrix} + w_{y_i}(t) \quad (3c)$$

where l_{ij} are constants satisfying $\sum_{j=1}^M l_{ij} = 0$ and describing the velocity coupling among the agents. (A discrete-time counterpart of the model has been used to describe the group movement of terrestrial animals (Park et al. 2019).

3. Coupled Kuramoto oscillators: The Kuramoto oscillators are described by a nonlinear dynamic model and are used to study synchronization problems in power grids (Skardal and Arenas 2015). Its linearized model can be expressed by (1) with the state $x(t)$ defined by $x(t) = (\theta_1(t), \dots, \theta_M(t))$, where $\theta_i(t)$ is the phase of the i -th oscillator, and the measurement $y_i(t)$ is defined as $y_i(t) = \theta_i(t)$: for each i in $\{1, \dots, M\}$,

$$\dot{\theta}_i(t) = K \sum_{j=1}^M L_{ij} \theta_j(t) + \omega_i \quad (4a)$$

$$y_i(t) = \theta_i(t) \quad (4b)$$

where K and L_{ij} are a global coupling strength and an entry of a network Laplacian matrix L , respectively. The constant ω_i

represents the natural frequency of the i -th oscillator.

The physical limitations on the agents' communication capability define the set of neighbors each agent can exchange information with and impose a graph structure on the network, which we represent by a directed graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$: The vertex set $\mathbb{V}$ represents all the agents in the network, and each edge $e = (i, j)$ in the edge set $\mathbb{E}$ denotes the feasibility of information transfer from Agent i to Agent j . The directed graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ essentially specifies the feasibility of information exchange between agents and imposes a communication constraint on the algorithm design for distributed estimation.

The following definitions from graph theory are useful for our discussions. For each i in $\mathbb{V}$, the neighborhood set $\mathbb{N}_i$ is defined as the set of agents that can transmit information to Agent i , including Agent i itself. The graph $\mathcal{G}$ is said to be *strongly connected* if there is a directed path between every pair of agents, and a strongly connected subgraph of $\mathcal{G}$ is said to be a *source component* if there is no incoming edge from any agent outside the subgraph.

With the system model (1) and the graphical representation $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ of the network, we now describe a distributed estimation algorithm that specifies how each agent computes its state estimate $\hat{x}_i(t)$. One of the technical challenges in the algorithm design is mainly due to the constraint that communication between the agents is restricted by $\mathcal{G}$ and, hence, designing a distributed estimation algorithm is substantially different from the centralized estimation problem.

Two critical aspects of the distributed estimation are (1) sharing and fusing state estimates among agents that are within same neighborhood and (2) updating the state estimate using the measurements on the system's state. In literature, the following class of parameterized linear estimation rules is proposed to implement the above two aspects in a single framework:

$$\begin{aligned} \dot{\hat{x}}_i(t) = & A\hat{x}_i(t) + Bu(t) \\ & + \underbrace{\sum_{j \in \mathbb{N}_i} \mathbf{H}_{ij} (\hat{x}_j(t) - \hat{x}_i(t))}_{\text{Estimate Fusion}} \\ & + \underbrace{\mathbf{K}_i (y_i(t) - C_i \hat{x}_i(t))}_{\text{Measurement Update}} + \mathbf{P}_i \underbrace{z_i(t)}_{\text{Augmented State}} \end{aligned} \quad (5a)$$

$$\dot{z}_i(t) = \mathbf{Q}_i (y_i(t) - C_i \hat{x}_i(t)) + \mathbf{S}_i z_i(t) \quad (5b)$$

We note that the third term in (5a) contributes to reaching agreement in the state estimates between neighboring agents and the fourth term is associated with the measurement update. The matrices $\mathbf{H}_{ij}$, $\mathbf{K}_i$, $\mathbf{P}_i$, $\mathbf{Q}_i$, $\mathbf{S}_i$ are the parameters of the estimation rule that need to be found in such a way that the estimate $\hat{x}_i(t)$ of every agent converges to the system's state $x(t)$ exponentially fast given that the disturbance terms $w_x(t)$ and $w_{y_i}(t)$ in (1) are both zero: for all i in $\{1, \dots, M\}$,

$$\|\hat{x}_i(t) - x(t)\|_2 \leq \alpha \exp(-\lambda t), \quad \forall t \geq 0 \quad (6)$$

where the positive variable λ is referred to as the *convergence rate*. We refer to such convergence of the state estimate at every agent as *state omniscience* (Park and Martins 2017).

The estimation rule (5) was motivated by the pioneering work (Olfati-Saber and Jalalkamali 2012) on distributed Kalman filtering (DKF) that proposed a new solution to the distributed estimation problem based on Kalman filtering and a consensus algorithm (Olfati-Saber et al. 2007). The major difference from the DKF approach is the introduction of the augmented states $z_i(t)$. The idea of having the augmented states was suggested in Park and Martins (2017) and Wang and Morse (2018) to establish a duality between the distributed estimation and decentralized control. In consequence, such duality allows to select suitable parameters of (5) that attain the state omniscience using decentralized controller synthesis methods. In the next section, we discuss more details on the parameter selection.

Parameter Selection for State Omniscience

One of the important design objectives is to select the parameters of the estimation rule (5) that result in the state omniscience. To achieve this objective, we first identify conditions on the system model (1) and the graph $\mathcal{G}$ that ensure the existence of such omniscience-achieving parameters, and then we devise a method to compute the parameters. We explain these two points based on recent results from Park and Martins (2017), Wang and Morse (2018), and Mitra and Sundaram (2018).

To explain this, let us recall a central property of (1) so-called *detectability*: the system model (1) is said to be detectable if the state $x(t)$ can be asymptotically estimated from $(y_1(\tau), \dots, y_M(\tau))$, $0 \leq \tau \leq t$ which are the measurements available to the entire network. As one can infer from the definition, the detectability is the minimum requirement for the system's state to be estimated using any type of estimation algorithms.

Park and Martins (2017) describe the necessary and sufficient conditions for the existence of an omniscience-achieving parameter selection in terms of the detectability of (1) and connectivity of the graph $\mathcal{G}$. In fact, when the graph is strongly connected, the detectability condition is the only requirement to ensure the existence of an omniscience-achieving parameter selection. In other cases, when the graph has multiple source components, a subsystem model of (1) associated with every source component needs to satisfy the detectability requirement.

Once the conditions are met, we can proceed with finding omniscience-achieving parameters for (5). In what follows, we discuss two representative approaches where, for the sake of the discussion, we assume that the graph $\mathcal{G}$ is strongly connected. The first approach is based on the results from Park and Martins (2017) and Wang and Morse (2018) in which the key idea is based on the observation that when (5) is re-written in terms of the estimation error $\tilde{x}_i(t) = \hat{x}_i(t) - x(t)$ and the augmented states $z_i(t)$, the parameter

selection problem can be equivalently viewed as a decentralized controller synthesis problem. By adopting well-established methods for decentralized control design (Corfmat and Morse 1976), one can find omniscience-achieving parameters for (5) for which the augmented state $z_1(k)$ of Agent 1 is required to be of the dimension $M - 1$ (the number of agents in the network less 1) and all other agents do not need to have the augmented states in their estimation rules. Hence, the average dimension of the estimation rule (5) becomes less than $n + 1$, which only increases linearly with respect to the order n of the system model. Furthermore, the method proposed in Wang and Morse (2018) explains how to select the parameters that attain a pre-selected convergence rate λ in (6).

The second approach is based on a canonical decomposition method from Mitra and Sundaram (2018). The parameter selection under this method is decomposed into the following two steps: Each agent i identifies a portion of the state $x(t)$ – called an *observable sub-state* of $x(t)$ – that can be estimated only from its own measurements $y_i(\tau)$, $0 \leq \tau \leq t$ and locally estimates the sub-state (without communication with other agents), using a centralized estimation approach such as the Luenberger observer. Then, through a consensus algorithm, the agent estimates the rest of $x(k)$. The method would have a limitation that each agent estimates its observable sub-state only with its own measurements, which would deteriorate the estimation quality when the measurements are subject to significant sensor noise $w_{y_i}(t)$. However, it has an advantage that all the agents in the network do not need to have the augmented states in their estimation rules, which reduces complexity in selecting the parameters for the estimation rule.

The above two approaches describe methods to select omniscience-achieving parameters for (5) when they exist. In addition to the parameter selection for the state omniscience, another critical design objective is in optimizing the parameter selection to improve disturbance attenuation performance of (5). In the next section, we review key contributions in the distributed estimation

study that describe methods to assess and optimize the performance.

Optimizing Disturbance Attenuation Performance

We discuss methods to analyze the effect of disturbances on performance of (5) and to refine

the parameter selection to attenuate such effect. Primary sources of the disturbances we consider are the modeling error $w_x(t)$ and sensor noise $w_{y_i}(t)$, and also we discuss cases in which the measurement matrix C_i and the neighborhood set $\mathbb{N}_i$ are subject to random switching. We begin by explaining the following two performance indices needed for our discussions:

$$\begin{aligned} H_\infty \text{ Estimation Performance} &= \frac{\text{Estimation Error}}{\text{Total Disturbance}} \\ &= \frac{(1/M) \sum_{i=1}^M \|\tilde{x}_i\|_{L_2}^2}{\|\tilde{x}(0)\|_2^2 + \|w_x\|_{L_2}^2 + (1/M) \sum_{i=1}^M \|w_{y_i}\|_{L_2}^2} \end{aligned} \quad (7)$$

$$\begin{aligned} H_\infty \text{ Consensus Performance} &= \frac{\text{Disagreement in State Estimates}}{\text{Total Disturbance}} \\ &= \frac{(1/M) \sum_{i=1}^M \sum_{j \in \mathbb{N}_i} \|\hat{x}_i - \hat{x}_j\|_{L_2}^2}{\|\tilde{x}(0)\|_2^2 + \|w_x\|_{L_2}^2 + (1/M) \sum_{i=1}^M \|w_{y_i}\|_{L_2}^2} \end{aligned} \quad (8)$$

where $\|\cdot\|_2$ is the 2-norm and for a given real-valued signal $a(t)$, $t \geq 0$, the norm $\|a\|_{L_2}$ is defined as $\|a\|_{L_2}^2 = \int_{t=0}^{\infty} \|a(t)\|_2^2 dt$. Essentially, the performance indices (7) and (8) describe the disturbance attenuation performance of (5) in attaining the state omniscience and reaching agreement (consensus) in state estimates, respectively, across the entire network, which are known as H_∞ robustness in estimation and consensus (Ugrinovskii 2011).

A practical consideration in the distributed estimation research is to devise computation methods for assessing and optimizing the performance indices (7) and (8). Theory on linear matrix inequality (LMI) has been widely adopted to assess disturbance attenuation performance of control and estimation algorithms. The theory uses matrix inequalities to characterize disturbance attenuation performance and also to design computational approaches to optimize the performance. In what follows, we introduce

some of key LMI-based approaches proposed in the distributed estimation literature.

Ugrinovskii (2011) proposes a LMI-based method to assess and optimize the performance indices (7) and (8) for the estimation rule (5). Notably, the method is used to establish a strong link between the H_∞ consensus performance and the parameter selection of $\mathbf{H}_{ij}$ in (5). The work of Ugrinovskii (2013) and Shen et al. (2010) investigates more challenging and practical scenarios in which the measurement matrix C_i and the neighborhood set $\mathbb{N}_i$ are subject to random switching. Such types of randomly switching measurement and communication models arise in engineering applications where sensors would fail to collect measurements and wireless communication becomes unreliable and suffers from random packet drops resulting in failure of information exchange between agents. Under this setting, LMI-based approaches are proposed in Ugrinovskii (2013) and Shen et al. (2010) to

analyze the effect of the sensor and communication failure on the H_∞ estimation and consensus performance, which also can be used to refine the parameter selection to attenuate the effect.

All these LMI-based approaches can be used not only to analyze and improve the disturbance attenuation performance but also as tools to design measurement models and communication topologies that attain desired performance.

Applications to Distributed Control

The theory of distributed estimation not only serves as principles in designing estimation algorithms in networks but also is applied to design control algorithms for large-scale dynamical systems in engineering applications. The work of Muehlebach and Trimpe (2018) investigates a problem of designing a communication-efficient distributed control algorithm for stabilizing a large-scale dynamical system. To achieve this, an event-triggered distributed estimation algorithm is proposed in which, based on devised event triggers, agents in a network selectively collect sensor measurements on the system's state and share the measurements across the entire network, which would be used to estimate and eventually stabilize the state of the system. Simulations with the proposed distributed control method demonstrate the possibility of using the method to safely maneuver a vehicle platoon while achieving communication efficiency.

The distributed estimation approach was adopted in developing animal-borne networked sensing devices (Park et al. 2019). The devices are designed to collect data on animal group behavior through registering their locations using GPS modules and recording animal point-of-view videos using on-board cameras. Since the video recording is an extremely power- and memory-intensive operation, prudent use of such data collection capability is needed to keep the devices running over an extended period of time. In this scenario, the distributed estimation is used to allow each animal-borne device to estimate the positions and velocities of members in an

animal group instrumented with the devices. The estimates on the group movement are then used to judiciously control the triggering of video recording to document group interactions among instrumented animals.

Summary and Future Directions

This entry overviews theory and application of distributed estimation in networks. The theory describes systematic ways to design a network of multiple agents that estimate the state of dynamical systems through sensing and communication. We reviewed key approaches that investigate the design of distributed estimation algorithms and study the conditions under which the algorithms attain the state omniscience. In addition, we discussed the disturbance attenuation performance of the algorithms and reviewed important LMI-based methods that characterize the performance and can be used to optimize the performance through parameter refinement.

Application is an important component of the distributed estimation research for which we have explained a few examples where the distributed estimation approaches are applied to control large-scale dynamical systems. Further investigations on algorithm design and applications to system control problems are important future directions which would include (1) co-designing distributed estimation and control algorithms for multi-agent decision-making over non-trivial communication graphs, which would change over time, and (2) investigating efficient inter-agent communication schemes for the distributed estimation that find the optimal trade-off between power consumption due to the communication and the estimation performance.

Cross-References

- [Averaging Algorithms and Consensus](#)
- [Consensus of Complex Multi-agent Systems](#)
- [Controllability of Network Systems](#)
- [Estimation and Control over Networks](#)
- [Flocking in Networked Systems](#)

- Graphs for Modeling Networked Interactions
- Information-Based Multi-Agent Systems
- Networked Control Systems: Architecture and Stability Issues
- Networked Control Systems: Estimation and Control Over Lossy Networks
- Networked Systems
- Oscillator Synchronization
- Vehicular Chains

Bibliography

- Corfmat J, Morse A (1976) Decentralized control of linear multivariable systems. *Automatica* 12(5):479–495
- Kar S, Moura JMF (2013) Consensus + innovations distributed inference over networks: cooperation and sensing in networked systems. *IEEE Signal Process Mag* 30(3):99–109
- Levine W, Athans M (1966) On the optimal error regulation of a string of moving vehicles. *IEEE Trans Autom Control* 11(3):355–361
- Mitra A, Sundaram S (2018) Distributed Observers for LTI Systems. *IEEE Trans Autom Control* 63(11):3689–3704
- Muehlebach M, Trimpe S (2018) Distributed event-based state estimation for networked systems: an LMI approach. *IEEE Trans Autom Control* 63(1):269–276
- Olfati-Saber R, Jalalkamali P (2012) Coupled distributed estimation and control for mobile sensor networks. *IEEE Trans Autom Control* 57(10):2609–2614
- Olfati-Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. *Proc IEEE* 95(1):215–233
- Park S, Martins NC (2017) Design of distributed LTI observers for state omniscience. *IEEE Trans Autom Control* 62(2):561–576
- Park S, Aschenbach KH, Ahmed M, Scott WL, Leonard NE, Abernathy K, Marshall G, Shepard M, Martins NC (2019) Animal-borne wireless network: remote imaging of community ecology. *J Field Robot* 1–25. <https://doi.org/10.1002/rob.21891>
- Shen B, Wang Z, Hung Y (2010) Distributed H_∞ -consensus filtering in sensor networks with multiple missing measurements: the finite-horizon case. *Automatica* 46(10):1682–1688
- Skardal PS, Arenas A (2015) Control of coupled oscillator networks with application to microgrid technologies. *Sci Adv* 1(7)
- Ugrinovskii V (2011) Distributed robust filtering with H_∞ consensus of estimates. *Automatica* 47(1):1–13
- Ugrinovskii V (2013) Distributed robust estimation over randomly switching networks using H_∞ consensus. *Automatica* 49(1):160–168
- Wang L, Morse AS (2018) A distributed observer for a time-invariant linear system. *IEEE Trans Autom Control* 63(7):2123–2130

Distributed Model Predictive Control

Gabriele Pannocchia
University of Pisa, Pisa, Italy

Abstract

Distributed model predictive control refers to a class of predictive control architectures in which a number of local controllers manipulate a subset of inputs to control a subset of outputs (states) composing the overall system. Different levels of communication and (non)cooperation exist, although in general the most compelling properties can be established only for cooperative schemes, those in which all local controllers optimize local inputs to minimize the same plantwide objective function. Starting from state-feedback algorithms for constrained linear systems, extensions are discussed to cover output feedback, reference target tracking, and nonlinear systems. An outlook of future directions is finally presented.

Keywords

Constrained large-scale systems · Cooperative control systems · Interacting dynamical systems

Introduction and Motivations

Large-scale systems (e.g., industrial processing plants, power generation networks, etc.) usually comprise several interconnected units which may exchange material, energy, and information streams. The overall effectiveness and profitability of such large-scale systems depend strongly on the level of local effectiveness and profitability of each unit but also on the level of interactions among the different units. An overall optimization goal can be achieved by adopting a single *centralized* model predictive control (MPC) system (Rawlings et al. 2017) in which *all* control input trajectories are optimized simultaneously to minimize a *common* objective.

This choice is often avoided for several reasons. When the overall number of inputs and states is very large, a single optimization problem may require computational resources (CPU time, memory, etc.) that are not available and/or compatible with the system's dynamics. Even if these limitations do not hold, it is often the case that organizational reasons require the use of smaller, local controllers, which are easier to coordinate and maintain.

Thus, industrial control systems are often *decentralized*, i.e., the overall system is divided into (possibly mildly coupled) subsystems, and a local controller is designed for each unit disregarding the interactions from/to other subsystems. Depending on the extent of dynamic coupling, it is well-known that the performance of such decentralized systems may be poor and stability properties may be even lost. *Distributed* predictive control architectures arise to meet performance specifications (stability at minimum) similar to centralized predictive control systems, still retaining the modularity and local character of the optimization problems solved by each controller.

Definitions and Architectures for Constrained Linear Systems

Subsystem Dynamics, Constraints, and Objectives

We start the description of distributed MPC algorithms by considering an overall discrete-time linear time-invariant system in the form:

$$x^+ = Ax + Bu, \quad y = Cx \quad (1)$$

in which $x \in \mathbb{R}^n$ and $x^+ \in \mathbb{R}^n$ are, respectively, the system state at a given time and, at a successor time, $u \in \mathbb{R}^m$ is the input and $y \in \mathbb{R}^p$ is the output.

We consider that the overall system (1) is divided into M subsystems, $\mathbb{S}_i$, defined by (dis-joint) sets of inputs and outputs (states), and each $\mathbb{S}_i$ is regulated by a *local* MPC. For each $\mathbb{S}_i$, we denote by $y_i \in \mathbb{R}^{p_i}$ its output, by $x_i \in \mathbb{R}^{n_i}$ its state, and by $u_i \in \mathbb{R}^{m_i}$ the control input

computed by the i th MPC. Due to interactions among subsystems, the local output y_i (and state x_i) is affected by control inputs computed by (some) other MPCs. Hence, the dynamics of $\mathbb{S}_i$ can be written as

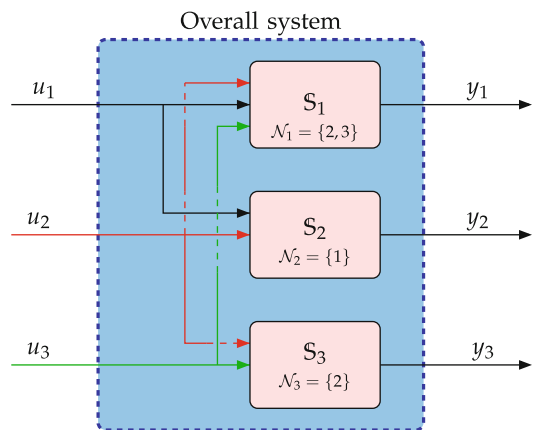
$$x_i^+ = A_i x_i + B_i u_i + \sum_{j \in \mathcal{N}_i} B_{ij} u_j, \quad y_i = C_i x_i \quad (2)$$

in which $\mathcal{N}_i$ denotes the indices of *neighbors* of $\mathbb{S}_i$, i.e., the subsystems whose inputs have an influence on the states of $\mathbb{S}_i$. To clarify the notation, we depict in Fig. 1 the case of three subsystems, with neighbors $\mathcal{N}_1 = \{2, 3\}$, $\mathcal{N}_2 = \{1\}$, and $\mathcal{N}_3 = \{2\}$.

Without loss of generality, we assume that each pair (A_i, B_i) is stabilizable. Moreover, the state of each subsystem x_i is assumed known (to the i th MPC) at each decision time. For each subsystem $\mathbb{S}_i$, inputs are required to fulfill (hard) constraints:

$$u_i \in \mathbb{U}_i, \quad i = 1, \dots, M \quad (3)$$

in which $\mathbb{U}_i$ are polyhedrons containing the origin in their interior. Moreover, we consider a quadratic stage cost function $\ell_i(x, u) \triangleq \frac{1}{2}(x' Q_i x + u' R_i u)$ and a terminal cost function $V_{fi}(x) \triangleq \frac{1}{2}x' P_i x$, with $Q_i \in \mathbb{R}^{n_i \times n_i}$, $R_i \in \mathbb{R}^{m_i \times m_i}$, and $P_i \in \mathbb{R}^{n_i \times n_i}$ positive definite.



Distributed Model Predictive Control, Fig. 1
Interconnected systems and neighbors definition

Without loss of generality, let $x_i(0)$ be the state of S_i at the current decision time. Consequently, the finite-horizon cost function associated with S_i is given by:

$$V_i(x_i(0), \mathbf{u}_i, \{\mathbf{u}_j\}_{j \in \mathcal{N}_i}) \triangleq \sum_{k=0}^{N-1} \ell_i(x_i(k), u_i(k)) + V_{fi}(x_i(N)) \quad (4)$$

in which $\mathbf{u}_i = (u_i(0), u_i(1), \dots, u_i(N-1))$ is a finite-horizon sequence of control inputs of S_i and $\mathbf{u}_j$ is similarly defined as a sequence of control inputs of each neighbor $j \in \mathcal{N}_i$. Notice that $V_i(\cdot)$ is a function of neighbors' input sequences, $\{\mathbf{u}_j\}_{j \in \mathcal{N}_i}$, due to the dynamics (2).

Decentralized, Noncooperative, and Cooperative Predictive Control Architectures

Several levels of communications and (non)cooperation can exist among the controllers, as depicted in Fig. 2 for the case of two subsystems.

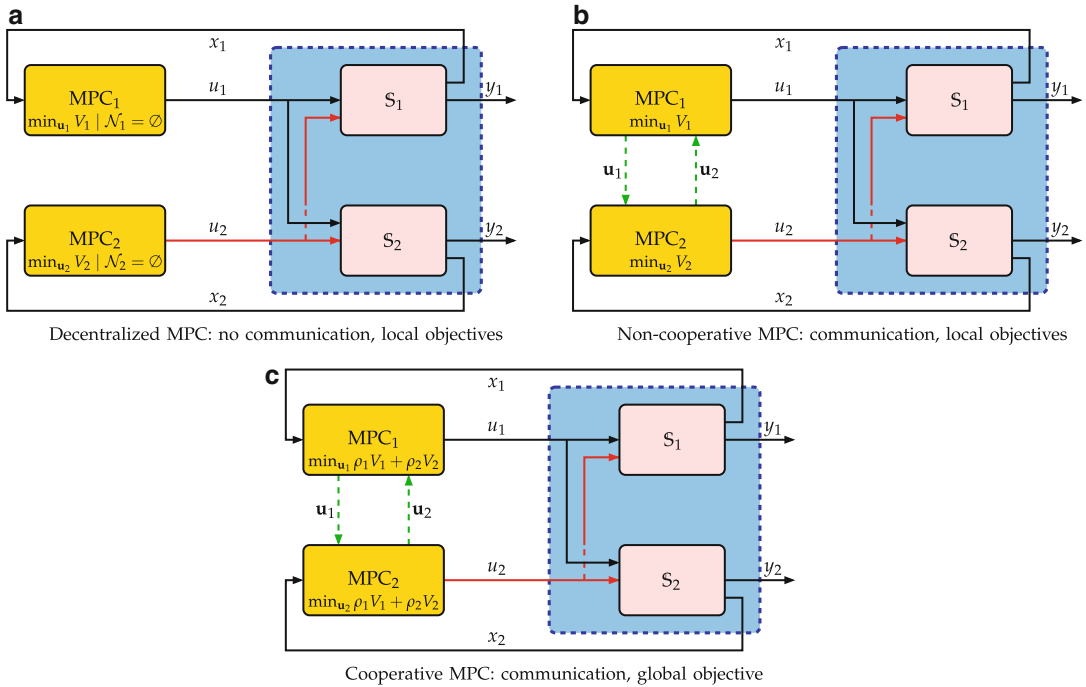
In *decentralized* MPC architectures, interactions among subsystems are neglected by forcing $\mathcal{N}_i = \emptyset$ for all i even if this is not true. That is, the subsystem model used in each local controller, instead of (2), is simply

$$x_i^+ = A_i x_i + B_i u_i, \quad y_i = C_i x_i \quad (5)$$

Therefore, an inherent mismatch exists between the model used by the local controllers (5) and the actual subsystem dynamics (2). Each local MPC solves the following finite-horizon optimal control problem (FHOCP):

$$\mathbb{P}_i^{\text{De}} : \min_{\mathbf{u}_i} V_i(\cdot) \quad \text{s.t. } \mathbf{u}_i \in \mathbb{U}_i^N, \mathcal{N}_i = \emptyset \quad (6)$$

We observe that in this case, $V_i(\cdot)$ depends only on local inputs, $\mathbf{u}_i$, because it is assumed that $\mathcal{N}_i = \emptyset$. Hence, each $\mathbb{P}_i^{\text{De}}$ is solved independently of the neighbors computations, and no iterations are performed. Clearly, depending on the actual level of interactions among subsystems, decentralized MPC architectures can perform poorly, namely,



Distributed Model Predictive Control, Fig. 2 Three distributed control architectures: decentralized MPC, noncooperative MPC, and cooperative MPC

being non-stabilizing. Performance certifications are still possible resorting to robust stability theory, i.e., by treating the neglected dynamics $\sum_{j \in \mathcal{N}_i} B_{ij} u_j$ as (bounded) disturbances (Riverso et al. 2013).

In *noncooperative* MPC architectures, the existing interactions among the subsystems are fully taken into account through (2). Given a known value of the neighbors' control input sequences, $\{\mathbf{u}_j\}_{j \in \mathcal{N}_i}$, each local MPC solves the following FHOC:

$$\mathbb{P}_i^{\text{NCDi}} : \min_{\mathbf{u}_i} V_i(\cdot) \text{ s.t. } \mathbf{u}_i \in \mathbb{U}_i^N \quad (7)$$

The obtained solution can be exchanged with the other local controllers to update the assumed neighbors' control input sequences, and iterations can be performed. We observe that this approach is noncooperative because local controllers try to optimize different, possibly competing, objectives. In general, no convergence is guaranteed in noncooperative iterations, and when this scheme converges, it leads to a so-called Nash equilibrium. However, the achieved local control inputs do not have proven stability properties (Rawlings et al. 2017, §6.2.3). To ensure closed-loop stability, variants can be formulated by including a sequential solution of local MPC problems; exploiting the notion (if any) of an auxiliary stabilizing decentralized control law, non-iterative noncooperative schemes are also proposed, in which stability guarantees are provided by ensuring a decrease of a centralized Lyapunov function at each decision time.

Finally, in *cooperative* MPC architectures, each local controller optimizes a common (plantwide) objective:

$$V(x(0), \mathbf{u}) \triangleq \sum_{i=1}^M \rho_i V_i(x_i(0), \mathbf{u}_i, \{\mathbf{u}_j\}_{j \in \mathcal{N}_i}) \quad (8)$$

in which $\rho_i > 0$, for all i , are given scalar weights and $\mathbf{u} \triangleq (\mathbf{u}_1, \dots, \mathbf{u}_M)$ is the overall control sequence. In particular, given a known value of other subsystems' control input sequences, $\{\mathbf{u}_j\}_{j \neq i}$, each local MPC solves the following FHOC:

$$\mathbb{P}_i^{\text{CDi}} : \min_{\mathbf{u}_i} V(\cdot) \text{ s.t. } \mathbf{u}_i \in \mathbb{U}_i^N \quad (9)$$

As in noncooperative schemes, the obtained solution can be exchanged with the other local controllers, and further iterations can be performed. Notice that in $\mathbb{P}_i^{\text{CDi}}$, the (possible) implications of the local control sequence $\mathbf{u}_i$ to all other subsystems' objectives, $V_j(\cdot)$ with $j \neq i$, are taken into account, as well as the effect of the neighbors' sequences $\{\mathbf{u}_j\}_{j \in \mathcal{N}_i}$ on the local state evolution through (2). Clearly, this approach is termed cooperative because all controllers compute *local* inputs to minimize a *global* objective. Convergence of cooperative iterations is guaranteed, and under suitable assumptions the converged solution is the centralized Pareto-optimal solution (Rawlings et al. 2017, §6.2.4). Furthermore, the achieved local control inputs have proven stabilizing properties (Stewart et al. 2010). Variants are also proposed in which each controller still optimizes a local objective, but cooperative iterations are performed to ensure a decrease of the global objective at each decision time (Maestre et al. 2011).

Cooperative Distributed MPC

Cooperative schemes are preferable over noncooperative schemes from many points of view, namely, in terms of superior theoretical guarantees and no larger computational requirements. In this section we focus on a prototype cooperative distributed MPC algorithm adapted from Stewart et al. (2010), highlighting the required computations and discussing the associated theoretical properties and guarantees.

Basic Algorithm

We present in Algorithm 1 a streamlined description of a cooperative distributed MPC algorithm, in which each local controller solves $\mathbb{P}_i^{\text{CDi}}$, given a previously computed value of all other subsystems' input sequences. For each local controller, the new iterate is defined as a convex combination of the newly computed solution with the previous iteration. A relative tolerance is defined, so that

cooperative iterations stop when all local controllers have computed a new iterate sufficiently close to the previous one. A maximum number of cooperative iterations can also be defined, so that a finite bound on the execution time can be established.

Algorithm 1 (*Cooperative MPC*). **Require:**

Overall warm start $\mathbf{u}^0 \triangleq (\mathbf{u}_1^0, \dots, \mathbf{u}_M^0)$, convex step weights $w_i > 0$, s.t. $\sum_{i=1}^M w_i = 1$, relative tolerance parameter $\epsilon > 0$, maximum cooperative iterations $c_{\max}$

```

1: Initialize:  $c \leftarrow 0$  and  $e_i \leftarrow 2\epsilon$  for  $i = 1, \dots, M$ .
2: while ( $c < c_{\max}$ ) and ( $\exists i | e_i > \epsilon$ ) do
3:    $c \leftarrow c + 1$ .
4:   for  $i = 1$  to  $M$  do
5:     Solve  $\mathbb{P}_i^{\text{CDi}}$  in (9) obtaining  $\mathbf{u}_i^*$ .
6:   end for
7:   for  $i = 1$  to  $M$  do
8:     Define new iterate:  $\mathbf{u}_i^c \triangleq w_i \mathbf{u}_i^* + (1 - w_i) \mathbf{u}_i^{c-1}$ .
9:     Compute convergence error:  $e_i \triangleq \frac{\|\mathbf{u}_i^c - \mathbf{u}_i^{c-1}\|}{\|\mathbf{u}_i^{c-1}\|}$ .
10:  end for
11: end while
12: return Overall solution:  $\mathbf{u}^c \triangleq (\mathbf{u}_1^c, \dots, \mathbf{u}_M^c)$ .

```

We observe that Step 8 implicitly defines the new overall iterate as a convex combination of the overall solutions *achieved* by each controller, that is,

$$\mathbf{u}^c = \sum_{i=1}^M w_i (\mathbf{u}_1^{c-1}, \dots, \mathbf{u}_i^*, \dots, \mathbf{u}_M^{c-1}) \quad (10)$$

It is also important to observe that Steps 5, 8, and 9 are performed separately by each controller.

Properties

The basic cooperative MPC described in Algorithm 1 enjoys several nice theoretical and practical properties, as detailed (Rawlings et al. 2017, §6.3.1):

1. *Feasibility of each iterate:* $\mathbf{u}_i^{c-1} \in \mathbb{U}_i^N$ implies $\mathbf{u}_i^c \in \mathbb{U}_i^N$, for all $i = 1, \dots, M$ and $c \in \mathbb{I}_{>0}$.
2. *Cost decrease at each iteration:* $V(x(0), \mathbf{u}^c) \leq V(x(0), \mathbf{u}^{c-1})$ for all $c \in \mathbb{I}_{>0}$.

3. *Cost convergence to the centralized optimum:* $\lim_{c \rightarrow \infty} V(x(0), \mathbf{u}^c) = \min_{\mathbf{u} \in \mathbb{U}^N} V(x(0), \mathbf{u})$, in which $\mathbb{U} \triangleq \mathbb{U}_1 \times \dots \times \mathbb{U}_M$.

Resorting to suboptimal MPC theory, the above properties (1) and (2) can be exploited to show that the origin of closed-loop system

$$x^+ = Ax + B\kappa^c(x), \text{ with } \kappa^c(x) \triangleq \mathbf{u}^c(0) \quad (11)$$

is *exponentially stable* for any finite $c \in \mathbb{I}_{>0}$. This result is of paramount (practical and theoretical) importance because it ensures closed-loop stability using cooperative distributed MPC with any finite number of cooperative iterations. As in centralized MPC based on the solution of a FHOC (Rawlings et al. 2017, §2.4.2), particular care of the terminal cost function $V_{fi}(\cdot)$ is necessary, possibly in conjunction with a terminal constraint $x_i(N) \in \mathbb{X}_{fi}$. Several options can be adopted as discussed, e.g., in Stewart et al. (2010, 2011).

Moreover, the results in Pannocchia et al. (2011) can be used to show *inherent robust stability* to system's disturbances and measurement errors. Therefore, we can confidently state that well-designed distributed cooperative MPC and centralized MPC algorithms share the *same guarantees* in terms of stability and robustness.

Complementary Aspects

We discuss in this section a number of complementary aspects of distributed MPC algorithms, omitting technical details for the sake of space.

Coupled Input Constraints and State Constraints

Convergence of the solution of cooperative distributed MPC toward the centralized (global) optimum holds when input constraints are in the form of (3), i.e., when no constraints involve inputs of different subsystems. Sometimes this assumption fails to hold, e.g., when several units share a common utility resource, that is, in addition to (3) some constraints involve inputs of more than one unit. In this situation, it is possible

that Algorithm 1 remains stuck at a fixed point, without improving the cost, even if it is still away from the centralized optimum (Rawlings et al. 2017, §6.3.2). It is important to point out that this situation is harmless from a closed-loop stability and robustness point of view. However, the degree of suboptimality in comparison with centralized MPC could be undesired from a performance point of view. To overcome this situation, a slightly different partitioning of the overall inputs into non-disjoint sets can be adopted (Stewart et al. 2010).

Similarly the presence of state constraints, even in decentralized form $x_i \in \mathbb{X}_i$ (with $i = 1, \dots, M$), can prevent convergence of a cooperative algorithm toward the centralized optimum. It is also important to point out that the local MPC controlling $\mathbb{S}_i$ needs to consider in the optimal control problem, besides local state constraints $x_i \in \mathbb{X}_i$ and also state constraints of all other subsystems $\mathbb{S}_j$ such that $i \in \mathcal{N}_j$. This ensures feasibility of each iterate and cost reduction; hence closed-loop stability (and robustness) can be established.

Another route to address the presence of coupling constraints is that based on suitable decomposition methods for distributed optimization, such as the dual decomposition and alternative direction method of multipliers (Doan et al. 2011; Farokhi et al. 2014).

Parsimonious Local System Representations

In cooperative distributed algorithms, each local optimization problem (9) is solved considering the evolution of the overall system state over the prediction horizon, so that the centralized cost function can be optimized by each local controller. Depending on the interconnections among the subsystems, this approach could be simplified by considering a subset of overall state dynamics, without losing the global optimality guarantees, as proposed in Razzanelli and Pannocchia (2017) and briefly reviewed.

From graph theory, we define the *outlet star* of each local subsystem $\mathbb{S}_i$, as the set of subsystems $\mathbb{S}_j$ of which $\mathbb{S}_i$ is neighbor, i.e., $\mathcal{O}_i = \{j | i \in \mathcal{N}_j\}$. Thus, the augmented state dynamics that should

be considered by $\mathbb{S}_i$ is composed by (2) and

$$\begin{aligned} x_j^+ &= A_j x_j + B_{ji} u_i + B_j u_j + \sum_{k \in \mathcal{N}_j \setminus i} B_{jk} u_k, \\ y_j &= C_j x_j \quad \forall j \in \mathcal{O}_i \end{aligned} \quad (12)$$

Thus, the FHOC (9) can be simplified to the following, more parsimonious, problem:

$$\mathbb{P}_i^{\text{PaCDi}} : \min_{\mathbf{u}_i} V_i(\cdot) + \sum_{j \in \mathcal{O}_i} V_j(\cdot) \quad \text{s.t. } \mathbf{u}_i \in \mathbb{U}_i^N \quad (13)$$

obtaining the same solution with significant computational savings (Razzanelli and Pannocchia 2017).

Output Feedback and Offset-Free Tracking

When the subsystem state cannot be directly measured, each local controller can use a local state estimator, namely, a Kalman filter (or Luenberger observer). Assuming that the pair (A_i, C_i) is detectable, the subsystem state estimate evolves as follows:

$$\hat{x}_i^+ = A_i \hat{x}_i + B_i u_i + \sum_{j \in \mathcal{N}_i} B_{ij} u_j + L_i (y_i - C_i \hat{x}_i) \quad (14)$$

in which $L_i \in \mathbb{R}^{n_i \times p_i}$ is the local Kalman predictor gain, chosen such that the matrix $(A_i - L_i C_i)$ is Schur. Stability of the closed-loop origin can be still established using minor variations (Rawlings et al. 2017, §6.3.3).

When offset-free control is sought, each local MPC can be equipped with an integrating disturbance model similarly to centralized offset-free MPC algorithms (Pannocchia 2015). Given the current estimate of the subsystem state and disturbance, a target calculation problem is solved to compute the state and input equilibrium pair such that (a subset of) output variables correspond to given set points. Such a target calculation problem can be performed in a centralized fashion or in a distributed manner (Razzanelli and Pannocchia 2017). Furthermore, the target calculation problem can be embedded into the FHOC (Ferramosca et al. 2013; Razzanelli and Pannocchia 2017).

Distributed Control for Nonlinear Systems

Several nonlinear distributed MPC algorithms have been recently proposed (Liu et al. 2009; Stewart et al. 2011). Some schemes require the presence of a coordinator, thus introducing a hierarchical structure (Scattolini 2009). In Stewart et al. (2011), instead, a cooperative distributed MPC architecture similar to the one discussed in the previous section has been proposed for nonlinear systems. Each local controller considers the following subsystem model:

$$\begin{aligned} x_i^+ &= f_i(x_i, u_i, u_j), \quad \text{with } j \in \mathcal{N}_i \\ y_i &= h_i(x_i) \end{aligned} \quad (15)$$

A problem (formally) identical to $\mathbb{P}_i^{\text{CDi}}$ in (9) is solved by each controller, and cooperative iterations are performed. However, non-convexity of $\mathbb{P}_i^{\text{CDi}}$ can make a convex combination step similar to Step 8 in Algorithm 1 not necessarily a cost improvement. As a workaround in such cases, Stewart et al. (2011) propose deleting the least effective control sequence computed by a local controller (repeating this deletion if necessary). In this way it is possible to show a monotonic decrease of the cost function at each cooperative iteration (Rawlings et al. 2017, §6.5).

Summary and Future Directions

We presented the basics and foundations of distributed model predictive control (DMPC) schemes, which prove useful and effective in the control of large-scale systems for which a single centralized predictive controller is not regarded as a possible or desirable solution, e.g., due to organizational requirements and/or computational limitations. In DMPCs, the overall controlled system is organized into a number of subsystems, in general featuring some dynamic couplings, and for each subsystem a local MPC is implemented.

Different flavors of communication and cooperation among the local controllers can be chosen by the designer, ranging from *decentralized* to

cooperative schemes. In cooperative DMPC algorithms, the dynamic interactions among the subsystems are fully taken into account, with limited communication overheads, and the same overall objective can be optimized by each local controller. When cooperative iterations are performed upon convergence, such DMPC algorithms achieve the same global minimum control sequence as that of the centralized MPC. Termination prior to convergence does not hinder stability and robustness guarantees.

In this contribution, after discussing an overview on possible communication and cooperation schemes, we addressed the design of a state-feedback and distributed MPC algorithm for linear systems subject to input constraints, with convergence and stability guarantees. Then, we discussed various extensions to coupled input constraints and state constraints, output feedback, reference target tracking, and nonlinear systems.

The research on DMPC algorithms has been extensive during the last decade, and some excellent review papers have been recently made available (Christofides et al. 2013; Scattolini 2009). Still, we expect DMPC to attract research efforts in various directions, as briefly discussed:

- *Nonlinear DMPC* algorithms (Liu et al. 2009; Stewart et al. 2011; Hours and Jones 2015) will require improvements in terms of global optimum goals.
- *Economic DMPC* and *tracking DMPC* (Ferramosca et al. 2013; Razzanelli and Pannocchia 2017; Köhler et al. 2018) will replace current formulations designed for regulation around the origin.
- *Reconfigurability*, e.g., addition/deletion of new local subsystems and controllers, is an ongoing topic, which finds particular applicability in the context of smart-grid systems (Molzahn et al. 2017). Preliminary results, discussed in Rivero et al. (2013) for decentralized control structures, have been also extended to distributed nonlinear systems (Rivero et al. 2016).

- *Constrained distributed estimation*, preliminarily addressed in Farina et al. (2012), is drawing attention (Yin and Liu 2017), although further insights are needed to bridge the gap between constrained estimation and control algorithms.
- Specific *distributed optimization algorithms* tailored to DMPC problems (Doan et al. 2011; Farokhi et al. 2014; Molzahn et al. 2017) will increase the effectiveness of DMPC algorithms.

Cross-References

- [Cooperative Solutions to Dynamic Games](#)
- [Nominal Model-Predictive Control](#)
- [Optimization Algorithms for Model Predictive Control](#)
- [Tracking Model Predictive Control](#)

Recommended Reading

General overviews on DMPC can be found in Christofides et al. (2013), Rawlings et al. (2017), and Scattolini (2009). DMPC algorithms for linear systems are discussed in Ferramosca et al. (2013), Rivero et al. (2013), Stewart et al. (2010), Maestre et al. (2011), and Razzanelli and Pannocchia (2017) and for nonlinear systems in Farina et al. (2012), Liu et al. (2009), and Stewart et al. (2011).

Bibliography

- Christofides PD, Scattolini R, Muñoz de la Peña D, Liu J (2013) Distributed model predictive control: a tutorial review and future research directions. *Comput Chem Eng* 51:21–41
- Doan MD, Keviczky T, De Schutter B (2011) An iterative scheme for distributed model predictive control using Fenchel's duality. *J Process Control* 21:746–755
- Farina M, Ferrari-Trecate G, Scattolini R (2012) Distributed moving horizon estimation for nonlinear constrained systems. *Int J Robust Nonlinear Control* 22:123–143
- Farokhi F, Shames I, Johansson KH (2014) Distributed MPC via dual decomposition and alternative direction method of multipliers. In: Maestre J, Negenborn R (eds) *Distributed model predictive control made easy. Intelligent systems, control and automation: science and engineering*, vol 69. Springer, Dordrecht
- Ferramosca A, Limon D, Alvarado I, Camacho EF (2013) Cooperative distributed MPC for tracking. *Automatica* 49:906–914
- Hours JH, Jones CN (2015) A parametric nonconvex decomposition algorithm for real-time and distributed NMPC. *IEEE Trans Autom Control* 61(2):287–302
- Köhler PN, Müller M, Allgöwer F (2018) A distributed economic MPC framework for cooperative control under conflicting objectives. *Automatica* 96:368–379
- Liu J, Muñoz de la Peña D, Christofides PD (2009) Distributed model predictive control of nonlinear process systems. *AIChE J* 55(5):1171–1184
- Maestre JM, Muñoz de la Peña D, Camacho EF (2011) Distributed model predictive control based on a cooperative game. *Optim Control Appl Methods* 32:153–176
- Molzahn DK, Dörfler F, Sandberg H, Low SH, Chakrabarti S, Baldick R, Lavaei J (2017) A survey of distributed optimization and control algorithms for electric power systems. *IEEE Trans Smart Grid* 8(6):2941–2962
- Pannocchia G (2015) Offset-free tracking MPC: a tutorial review and comparison of different formulations. In: 2015 European control conference (ECC), pp 527–532
- Pannocchia G, Rawlings JB, Wright SJ (2011) Conditions under which suboptimal nonlinear MPC is inherently robust. *Syst Control Lett* 60:747–755
- Rawlings JB, Mayne DQ, Mayne DQ, Diehl MM (2017) *Model predictive control: theory, computation, and design*, 2nd edn. Nob Hill Publishing, Madison
- Razzanelli M, Pannocchia G (2017) Parsimonious cooperative distributed MPC algorithms for offset-free tracking. *J Process Control* 60:1–13
- Rivero S, Farina M, Ferrari-Trecate G (2013) Plug-and-play decentralized model predictive control for linear systems. *IEEE Trans Autom Control* 58:2608–2614
- Rivero S, Boem F, Ferrari-Trecate G, Parisini T (2016) Plug-and-play fault detection and control-reconfiguration for a class of nonlinear large-scale constrained systems. *IEEE Trans Autom Control* 61:3963–3978
- Scattolini R (2009) A survey on hierarchical and distributed model predictive control. *J Process Control* 19:723–731
- Stewart BT, Venkat AN, Rawlings JB, Wright SJ, Pannocchia G (2010) Cooperative distributed model predictive control. *Syst Control Lett* 59:460–469
- Stewart BT, Wright SJ, Rawlings JB (2011) Cooperative distributed model predictive control for nonlinear systems. *J Process Control* 21:698–704
- Yin X, Liu J (2017) Distributed moving horizon state estimation of two-time-scale nonlinear systems. *Automatica* 79(5):152–161

Distributed Optimization

Angelia Nedić

Industrial and Enterprise Systems Engineering,
University of Illinois, Urbana, IL, USA

Abstract

The paper provides an overview of the distributed first-order optimization methods for solving a constrained convex minimization problem, where the objective function is the sum of local objective functions of the agents in a network. This problem has gained a lot of interest due to its emergence in many applications in distributed control and coordination of autonomous agents and distributed estimation and signal processing in wireless networks.

Keywords

Collaborative multi-agent systems ·
Consensus protocol · Gradient-projection
method · Networked systems

Introduction

There has been much recent interest in distributed optimization pertinent to optimization aspects arising in control and coordination of networks consisting of multiple (possibly mobile) agents and in estimation and signal processing in sensor networks (Mesbahi and Egerstedt 2010; Bullo et al. 2009; Hendrickx 2008; Olshevsky 2010; Kar and Moura 2011; Martinoli et al. 2013). In many of these applications, the network system goal is to optimize a global objective through local agent-based computations and local information exchange with immediate neighbors in the underlying communication network. This is motivated mainly by the emergence of large-scale data and/or large-scale networks and new networking applications such as mobile ad hoc networks and wireless sensor networks, characterized by the lack of centralized access to information and time-varying connectivity.

Control and optimization algorithms deployed in such networks should be completely distributed (relying only on local observations and information), robust against unexpected changes in topology (i.e., link or node failures) and against unreliable communication (noisy links or quantized data) (see ► [Networked Systems](#)). Furthermore, it is desired that the algorithms are scalable in the size of the network.

Generally speaking, the problem of distributed optimization consists of three main components:

1. The optimization problem that the network of agents wants to solve collectively (specifying an objective function and constraints)
2. The local information structure, which describes what information is locally known or observable by each agent in the system (who knows what and when)
3. The communication structure, which specifies the connectivity topology of the underlying communication network and other features of the communication environment

The algorithms for solving such global network problems need to comply with the distributed knowledge about the problem among the agents and obey the local connectivity structure of the communication network (► [Networked Systems](#); ► [Graphs for Modeling Networked Interactions](#)).

Networked System Problem

Given a set $N = \{1, 2, \dots, n\}$ of agents (also referred to as nodes), the global system problem has the following form:

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^n f_i(x) \\ &\text{subject to} && x \in X. \end{aligned} \quad (1)$$

Each $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ is a convex function which represents the local objective of agent i , while $X \subseteq \mathbb{R}^d$ is a closed convex set. The function f_i is a private function known only to agent i ,

while the set X is commonly known by all agents $i \in N$. The vector $x \in X$ represents a global decision vector which the agents want to optimize using local information. The problem is a simple constrained convex optimization problem, where the global objective function is given by the sum of the individual objective functions $f_i(x)$ of the agents in the system. As such, the objective function is the sum of non-separable convex functions corresponding to multiple agents connected over a network.

As an example, consider the problem arising in support vector machines (SVMs), which are a popular tool for classification problems. Each agent i has a set $S_i = \left\{ \left(a_j^{(i)}, b_j^{(i)} \right) \right\}_{j=1}^{m_i}$ of m_i sample-label pairs, where $a_j^{(i)} \in \mathbb{R}^d$ is a data point and $b_j^{(i)} \in \{+1, -1\}$ is its corresponding (correct) label. The number m_i of data points for every agent i is typically very large (hundreds of thousands). Without sharing the data points, the agents want to collectively find a hyperplane that separates all the data, i.e., a hyperplane that separates (with a maximal separation distance) the data with label 1 from the data with label -1 in the global data set $\bigcup_{i=1}^n S_i$. Thus, the agents need to solve an unconstrained version of the problem (1), where the decision variable $x \in \mathbb{R}^d$ is a hyperplane normal and the objective function f_i of agent i is given by

$$f_i(x) = \frac{\lambda}{2} \|x\|^2 + \sum_{j=1}^{m_i} \max \left\{ 0, 1 - b_j^{(i)} \left(x' a_j^{(i)} \right) \right\},$$

where λ is a regularization parameter (common to all agents).

The network communication structure is represented by a directed (or undirected) graph $G = (N, E)$, with the vertex set N and the edge set E . The network is used as a medium to diffuse the information from an agent to every other agent through local agent interactions over time ([► Graphs for Modeling Networked Interactions](#)). To accommodate the information spread through the entire network, it is typically assumed that the network communication graph $G = (N, E)$ is strongly connected

([► Dynamic Graphs, Connectivity of](#)). In the graph, a link (i, j) means that agent $i \in N$ receives the relevant information from agent $j \in N$.

Distributed Algorithms

The algorithms for solving problem (1) are constructed by using standard optimization techniques in combination with a mechanism for information diffusion through local agent interactions. A control point of view for the design of distributed algorithms has a nice exposition in Wang and Elia (2011).

One of the existing optimization techniques is the so-called incremental method, where the information is processed along a directed cycle in the graph. In this approach the estimate is passed from an agent to its neighbor (along the cycle), and only one agent updates at a time (Bertsekas 1997; Tseng 1998; Nedić and Bertsekas 2000, 2001; Nedić et al. 2001; Rabbat and Nowak 2004; Johansson et al. 2009; Blatt et al. 2007; Ram et al. 2009; Johansson 2008); for a detailed literature on incremental methods, see the textbooks Bertsekas (1999) and Bertsekas et al. (2003).

More recently, one of the techniques that gained popularity as a mechanism for information diffusion is a consensus protocol, in which the agent diffuses the information through the network through locally weighted averaging of their incoming data ([► Averaging Algorithms and Consensus](#)). The problem of reaching a consensus on a particular scalar value, or computing exact averages of the initial values of the agents, has gained an unprecedented interest as a central problem inherent to cooperative behavior in networked systems (Olshevsky 2010; Vicsek et al. 1995; Jadbabaie et al. 2003; Boyd et al. 2005; Olfati-Saber and Murray 2004; Cao et al. 2005, 2008a,b; Olshevsky and Tsitsiklis 2006, 2009; Wan and Lemmon 2009; Blondel et al. 2005; Touri 2011).

Using the consensus technique, a class of distributed algorithms has emerged, as a combination of the consensus protocols and the gradient-type methods. The gradient-based

approaches are particularly suitable, as they have a small overhead per iteration and are, in general, robust to various sources of errors and uncertainties.

The technique of using the network as a medium to propagate the relevant information for optimization purpose has its origins in the work by Tsitsiklis (1984), Tsitsiklis et al. (1986), and Bertsekas and Tsitsiklis (1997), where the network has been used to decompose the vector x components across different agents, while all agents share the same objective function.

Algorithms Using Weighted Averaging

The technique has recently been employed in Nedić and Ozdaglar (2009) (see also Nedić and Ozdaglar 2007, 2010) to deal with problems of the form (1) when the agents have different objective functions f_i , but their decisions are fully coupled through the common vector variable x . In a series of recent work (to be detailed later), the following distributed algorithm has emerged. Letting $x_i(k) \in X$ be an estimate (of the optimal decision) at agent i and time k , the next iterate is constructed through two updates. The first update is a consensus-like iteration, whereby, upon receiving the estimates $x_j(k)$ from its (in)neighbors j , the agent i aligns its estimate with its neighbors through averaging, formally given by

$$v_i(k) = \sum_{j \in N_i} w_{ij} x_j(k), \quad (2)$$

where N_i is the neighbor set

$$N_i = \{j \in N | (j, i) \in E\} \cup \{i\}.$$

The neighbor set N_i includes agent i itself, since the agent always has access to its own information. The scalar w_{ij} is a nonnegative weight that agent i places on the incoming information from neighbor $j \in N_i$. These weights sum to 1, i.e., $\sum_{j \in N_i} w_{ij} = 1$, thus yielding $v_i(k)$ as a (local) convex combination of $x_j(k)$, $j \in N_i$, obtained by agent i .

After computing $v_i(k)$, agent i computes a new iterate $x_i(k+1)$ by performing a gradient-

projection step, aimed at minimizing its own objective f_i , of the following form:

$$x_i(k+1) = \Pi_X [v_i(k) - \alpha(k) \nabla f_i(v_i(k))], \quad (3)$$

where $\Pi_X[x]$ is the projection of a point x on the set X (in the Euclidean norm), $\alpha(k) > 0$ is the stepsize at time k , and $\nabla f_i(z)$ is the gradient of f_i at a point z .

When all functions are zero and X is the entire space $\mathbb{P}^d$, the distributed algorithm (2) and (3) reduces to the linear-iteration method:

$$x_i(k+1) = \sum_{j \in N_i} w_{ij} x_j(k), \quad (4)$$

which is known as *consensus* or *agreement protocol*. This protocol is employed when the agents in the network wish to align their decision vectors $x_i(k)$ to a common vector $\hat{x}$. The alignment is attained asymptotically (as $k \rightarrow \infty$).

In the presence of objective function and constraints, the distributed algorithm in (2) and (3) corresponds to “forced alignment” guided by the gradient forces $\sum_{i=1}^n \nabla f_i(x)$. Under appropriate conditions, the alignment is forced to a common vector x^* that minimizes the network objective $\sum_{i=1}^n f_i(x)$ over the set X . This corresponds to the convergence of the iterates $x_i(k)$ to a common solution $x^* \in X$ as $k \rightarrow \infty$ for all agents $i \in N$.

The conditions under which the convergence of $\{x_i(k)\}$ to a common solution $x^* \in X$ occurs are a combination of the conditions needed for the consensus protocol to converge and the conditions imposed on the functions f_i and the stepsize $\alpha(k)$ to ensure the convergence of standard gradient-projection methods. For the consensus part, the conditions should guarantee that the pure consensus protocol in (4) converges to the average $\frac{1}{n} \sum_{i=1}^n x_i(0)$ of the initial agent values. This requirement transfers to the condition that the weights w_{ij} give rise to a doubly stochastic weight matrix W , whose entries are w_{ij} defined by the weights in (4) and augmented by $w_{ij} = 0$ for $j \notin N_i$.

Intuitively, the requirement that the weight matrix W is doubly stochastic ensures that each

agent has the same influence on the system behavior, in a long run. More specifically, the doubly stochastic weights ensure that the system as whole minimizes $\sum_{i=1}^n \frac{1}{n} f_i(x)$, where the factor $1/n$ is seen as portion of the influence of agent i . When the weight matrix W is only row stochastic, the consensus protocol converges to a weighted average $\sum_{i=1}^n \pi_i x_i(0)$ of the agent initial values, where π is the left eigenvector of W associated with the eigenvalue 1. In general, the values π_i can be different for different indices i , and the distributed algorithm in (2) and (3) results in minimizing the function $\sum_{i=1}^n \pi_i f_i(x)$ over X , and thus not solving problem (1).

The distributed algorithm (2) and (3) has been proposed and analyzed in Ram et al. (2010a, 2012), where the convergence had been established for diminishing stepsize rule (i.e., $\sum_k \alpha(k) = \infty$ and $\sum_k \alpha^2(k) < \infty$). As seen from the convergence analysis (see, e.g., Nedić and Ozdaglar 2009; Ram et al. 2012), for the convergence of the method, it is critical that the iterate disagreements $\|x_i(k) - x_j(k)\|$ converge linearly in time, for all $i \neq j$. This fast disagreement decay seems to be indispensable for ensuring the stability of the iterative process (2) and (3).

According to the distributed algorithm (2) and (3), at first, each agent aligns its estimate $x_i(k)$ with the estimates $x_j(k)$ that are received from its neighbors and, then, updates based on its local objective f_i which is to be minimized over $x \in X$. Alternatively, the distributed method can be constructed by interchanging the alignment step and the gradient-projection step. Such a method, while having the same asymptotic performance as the method in (2) and (3), exhibits a somewhat slower (transient) convergence behavior due to a larger misalignment resulting from taking the gradient-based updates at first. This alternative has been initially proposed independently in Lopes and Sayed (2006) (where simulation results have been reported), in Nedić and Ozdaglar (2007, 2009) (where the convergence analysis for a time-varying networks and a constant stepsize is given), and in Nedić et al. (2008) (with the quantization effects) and further investigated in Lopes and Sayed (2008), Nedić

et al. (2010), and Cattivelli and Sayed (2010). More recently, it has been considered in Lobel et al. (2011) for state-dependent weights and in Tu and Sayed (2012) where the performance is compared with that of an algorithm of the form (2) and (3) for estimation problems.

Algorithm Extensions: Over the past years, many extensions of the distributed algorithm in (2) and (3) have been developed, including the following:

- (a) *Time-varying communication graphs:* The algorithm naturally extends to the case of time-varying connectivity graphs $\{G(k)\}$, with $G(k) = (N, E(k))$ defined over the node set N and time-varying links $E(k)$. In this case, the weights w_{ij} in (2) are replaced with $w_{ij}(k)$ and, similarly, the neighbor set N_i is replaced with the corresponding time-dependent neighbor set $N_i(k)$ specified by the graph $G(k)$. The convergence of the algorithm (2) and (3) with these modifications typically requires some additional assumptions of the network connectivity over time and the assumptions on the entries in the corresponding weight matrix sequence $\{W(k)\}$, where $w_{ij}(k) = 0$ for $j \notin N_i(k)$. These conditions are the same as those that guarantee the convergence of the (row stochastic) matrix sequence $\{W(k)\}$ to a rank-one row-stochastic matrix, such as a connectivity over some fixed period (of a sliding-time window), the nonzero diagonal entries in $W(k)$, and the existence of a uniform lower bound on positive entries in $W(k)$; see, for example, Cao et al. (2008b), Touri (2011), Tsitsiklis (1984), Nedić and Ozdaglar (2010), Moreau (2005), and Ren and Beard (2005).
- (b) *Noisy gradients:* The algorithm in (2) and (3) works also when the gradient computations $\nabla f_i(x)$ in update (3) are erroneous with random errors. This corresponds to using a stochastic gradient $\tilde{\nabla} f_i(x)$ instead of $\nabla f_i(x)$, resulting in the following stochastic gradient-projection step:

$$x_i(k+1) = \prod_X \left[v_i(k) - \alpha(k) \tilde{\nabla} f_i(v_i(k)) \right]$$

instead of (3). The convergence of these methods is established for the cases when the stochastic gradients are consistent estimates of the actual gradient, i.e.,

$$\mathbb{E} \left[\tilde{\nabla} f_i(v_i(k)) | v_i(k) \right] = \nabla f_i(v_i(k)).$$

The convergence of these methods typically requires the use of non-summable but square-summable stepsize sequence $\{\alpha(k)\}$ (i.e., $\sum_k \alpha(k) = \infty$ and $\sum_k \alpha^2(k) < \infty$), e.g., Ram et al. (2010a).

- (c) *Noisy or unreliable communication links:* Communication medium is not always perfect and, often, the communication links are characterized by some random noise process. In this case, while agent $j \in N_i$ sends its estimate $x_j(k)$ to agent i , the agent does not receive the intended message. Rather, it receives $x_j(k)$ with some random link-dependent noise $\xi_{ij}(k)$, i.e., it receives $x_j(k) + \xi_{ij}(k)$ instead of $x_j(k)$ (see Kar and Moura 2011; Patterson et al. 2009; Touri and Nedić 2009 for the influence of noise and link failure on consensus). In such cases, the distributed optimization algorithm needs to be modified to include a stepsize for noise attenuation and the standard stepsize for the gradient scaling. These stepsizes are coupled through an appropriate relative-growth conditions which ensure that the gradient information is maintained at the right level and, at the same time, link-noise is attenuated appropriately (Srivastava et al. 2010; Srivastava and Nedić 2011). Other imperfections of the communication links can also be modeled and incorporated into the optimization method, such as link failures and quantization effects, which can be built using the existing results for consensus protocol (e.g., Kar and Moura 2010, 2011; Nedić et al. 2008; Carli et al. 2007).

- (d) *Asynchronous implementations:* The method in (2) and (3) has simultaneous updates, evident in all agents exchanging and updating information in synchronous time steps indexed by k . In some communication

settings, the synchronization of agents is impractical, and the agents are using their own clocks which are not synchronized but do tick according to a common time interval. Such communications result in random weights $w_{ij}(k)$ and random neighbor set $N_i(k) \subseteq E$ in (2), which are typically independent and identically distributed (over time). Most common models are random gossip and random broadcast. In the gossip model, at any time, two randomly selected agents i and j communicate and update, while the other agents sleep (Boyd et al. 2005; Srivastava and Nedić 2011; Kashyap et al. 2007; Ram et al. 2010b; Srivastava 2011). In the broadcast model, a random agent i wakes up and broadcasts its estimate $x_i(k)$. Its neighbors that receive the estimate update their iterates, while the other agents (including the agent who broadcasted) do not update (Aysal et al. 2008; Nedić 2011).

- (e) *Distributed constraints:* One of the more challenging aspects is the extension of the algorithm to the case when the constraint set X in (1) is given as an intersection of closed convex sets X_i , one set per agent. Specifically, the set X in (1) is defined by

$$X = \bigcap_{i=1}^n X_i,$$

where the set X_i is known to agent i only. In this case, the algorithm has a slight modification at the update of $x_i(k+1)$ in (3), where the projection is on the local set X_i instead of X , i.e., the update in (3) is replaced with the following update:

$$x_i(k+1) = \prod_{X_i} \left[v_i(k) - \alpha(k) \nabla f_i(v_i(k)) \right]. \quad (5)$$

The resulting method (2), (5) converges under some additional assumptions on the sets X_i , such as the nonempty interior assumption (i.e., the set $\bigcap_{i=1}^n X_i$ has a nonempty interior), a linear-intersection assumption (each X_i is an intersection of finitely many linear equality and/or inequality constraints), or the

Slater condition (Nedić et al. 2010; Srivastava and Nedić 2011; Srivastava 2011; Lee and Nedić 2012; Zhu and Martínez 2012).

In principle, most of the simple first-order methods that solve a centralized problem of the form (1) can also be distributed among the agents (through the use of consensus protocols) to solve distributed problem (1). For example, the Nesterov dual-averaging subgradient method (Nesterov 2005) can be distributed as proposed in Duchi et al. (2012), a distributed Newton-Raphson method has been proposed and studied in Zanella et al. (2011), while a distributed simplex algorithm has been constructed and analyzed in Bürger et al. (2012). An interesting method based on finding a zero of the gradient $\nabla f = \sum_{i=1}^n \nabla f_i$, distributedly, has been proposed and analyzed in Lu and Tang (2012). Some other distributed algorithms and their implementations can be found in Johansson et al. (2007), Tsianos et al. (2012a, b), Dominguez-Garcia and Hadjicostis (2011), Tsianos (2013), Gharesifard and Cortés (2012a), Jakovetic et al. (2011a, b), and Zargham et al. (2012).

Summary and Future Directions

The distributed optimization algorithms have been developed mainly using consensus protocols that are based on weighted averaging, also known as linear-iterative methods. The convergence behavior and convergence rate analysis of these methods combines the tools from optimization theory, graph theory, and matrix analysis. The main drawback of these algorithms is that they require (at least theoretically) the use of doubly stochastic weight matrix W (or $W(k)$ in time-varying case) in order to solve problem (1). This requirement can be accommodated by allowing agents to exchange locally some additional information on the weights that they intend to use or their degree knowledge. However, in general, constructing such doubly stochastic weights distributedly on directed graphs is rather a complex problem (Gharesifard and Cortés 2012b).

As an alternative, which seems a promising direction for future research, is the use of so-called push-sum protocol (or sum-ratio algorithm) for consensus problem (Kempe et al. 2003; Benezit et al. 2010). This direction is pioneered in Tsianos et al. (2012b), Tsianos (2013), and Tsianos and Rabbat (2011) for static graphs and recently extended to directed graphs Nedić and Olshevsky (2013) for an unconstrained version of problem (1).

Another promising direction lies in the use of alternating direction method of multipliers (ADMM) in combination with the graph-Laplacian formulation of consensus constraints $N_i x_i = \sum_{j \in N_i} x_j$. A nice exposure to ADMM method is given in Boyd et al. (2010). The first work to address the development of distributed ADMM over a network is Wei and Ozdaglar (2012), where a static network is considered. Its distributed implementation over time-varying graphs will be an important and challenging task.

Cross-References

- Averaging Algorithms and Consensus
- Dynamic Graphs, Connectivity of
- Graphs for Modeling Networked Interactions
- Networked Systems

Acknowledgments The author would like to thank J. Cortés for valuable suggestions to improve the article. Also, the author gratefully acknowledges the support by the National Science Foundation under grant CCF 11-11342 and by the Office of Naval Research under grant N00014-12-1-0998.

Recommended Reading

In addition to the below cited literature the useful material relevant to consensus and matrix-product convergence theory includes:

- Chatterjee S, Seneta E (1977) Towards consensus: some convergence theorems on repeated averaging. *J Appl Probab* 14(1):89–97
- Cogburn R (1986) On products of random stochastic matrices. *Random matrices and their applications*. American Mathematical Society, vol. 50, pp 199–213

- DeGroot MH (1974) Reaching a consensus. *J Am Stat Assoc* 69(345):118–121
- Lorenz J (2005) A stabilization theorem for continuous opinion dynamics. *Physica A: Stat Mech Appl* 355:217–223
- Rosenblatt M (1965) Products of independent identically distributed stochastic matrices. *J Math Anal Appl* 11(1):1–10
- Shen J (2000) A geometric approach to ergodic non-homogeneous Markov chains. In: T.-X. He (Ed.), *Proc. Wavelet Analysis and Multiresolution Methods*. Marcel Dekker Inc., New York, vol 212, pp. 341–366
- Tahbaz-Salehi A, Jadbabaie A (2010) Consensus over ergodic stationary graph processes. *IEEE Trans Autom Control* 55(1):225–230
- Touri B (2012) *Product of random stochastic matrices and distributed averaging*. Springer Theses. Springer, Berlin/New York
- Wolfowitz J (1963) Products of indecomposable, aperiodic, stochastic matrices. *Proc Am Math Soc* 14(4):733–737
- In: *Proceedings of IEEE INFOCOM*, Miami, vol 3, pp 1653–1664
- Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2010) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach Learn* 3(1): 1–122
- Bullo F, Cortés J, Martínez S (2009) *Distributed control of robotic networks*. Applied mathematics series. Princeton University Press, Princeton
- Bürger M, Notarsetfano G, Bullo F, Allgöwer F (2012) A distributed simplex algorithm for degenerate linear programs and multi-agent assignments. *Automatica* 48(9):2298–2304
- Cao M, Spielman D, Morse A (2005) A lower bound on convergence of a distributed network consensus algorithm. In: *Proceedings of IEEE CDC*, Seville, pp 2356–2361
- Cao M, Morse A, Anderson B (2008a) Reaching a consensus in a dynamically changing environment: a graphical approach. *SIAM J Control Optim* 47(2): 575–600
- Cao M, Morse A, Anderson B (2008b) Reaching a consensus in a dynamically changing environment: convergence rates, measurement delays, and asynchronous events. *SIAM J Control Optim* 47(2):601–623
- Carli R, Fagnani F, Frasca P, Taylor T, Zampieri S (2007) Average consensus on networks with transmission noise or quantization. In: *Proceedings of European control conference*, Kos
- Cattivelli F, Sayed A (2010) Diffusion LMS strategies for distributed estimation. *IEEE Trans Signal Process* 58(3):1035–1048
- Dominguez-Garcia A, Hadjicostis C (2011) Distributed strategies for average consensus in directed graphs. In: *Proceedings of the IEEE conference on decision and control*, Orlando, Dec 2011
- Duchi J, Agarwal A, Wainwright M (2012) Dual averaging for distributed optimization: convergence analysis and network scaling. *IEEE Trans Autom Control* 57(3):592–606
- Gharesifard B, Cortés J (2012a) Distributed continuous-time convex optimization on weight-balanced digraphs. <http://arxiv.org/pdf/1204.0304.pdf>
- Gharesifard B, Cortés J (2012b) Distributed strategies for generating weight-balanced and doubly stochastic digraphs. *Eur J Control* 18(6): 539–557
- Hendrickx J (2008) *Graphs and networks for the analysis of autonomous agent systems*. Ph.D. dissertation, Université Catholique de Louvain
- Jadbabaie A, Lin J, Morse S (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans Autom Control* 48(6): 988–1001
- Jakovetic D, Xavier J, Moura J (2011a) Cooperative convex optimization in networked systems: augmented lagrangian algorithms with directed gossip communication. *IEEE Trans Signal Process* 59(8): 3889–3902

Bibliography

- Jakovetic D, Xavier J, Moura J (2011b) Fast distributed gradient methods. Available at: <http://arxiv.org/abs/1112.2972>
- Johansson B (2008) On distributed optimization in networked systems. Ph.D. dissertation, Royal Institute of Technology, Stockholm
- Johansson B, Rabi M, Johansson M (2007) A simple peer-to-peer algorithm for distributed optimization in sensor networks. In: Proceedings of the 46th IEEE conference on decision and control, New Orleans, Dec 2007, pp 4705–4710
- Johansson B, Rabi M, Johansson M (2009) A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM J Control Optim* 20(3):1157–1170
- Kar S, Moura J (2010) Distributed consensus algorithms in sensor networks: quantized data and random link failures. *IEEE Trans Signal Process* 58(3):1383–1400
- Kar S, Moura J (2011) Convergence rate analysis of distributed gossip (linear parameter) estimation: fundamental limits and tradeoffs. *IEEE J Sel Top Signal Process* 5(4):674–690
- Kashyap A, Basar T, Srikant R (2007) Quantized consensus. *Automatica* 43(7):1192–1203
- Kempe D, Dobra A, Gehrke J (2003) Gossip-based computation of aggregate information. In: Proceedings of the 44th annual IEEE symposium on foundations of computer science, Cambridge, Oct 2003, pp 482–491
- Lee S, Nedić A (2012) Distributed random projection algorithm for convex optimization. *IEEE J Sel Top Signal Process* 48(6):988–1001 (accepted to appear)
- Lobel I, Ozdaglar A, Feijer D (2011) Distributed multi-agent optimization with state-dependent communication. *Math Program* 129(2):255–284
- Lopes C, Sayed A (2006) Distributed processing over adaptive networks. In: Adaptive sensor array processing workshop, MIT Lincoln Laboratory, Lexington, pp 1–5
- Lopes C, Sayed A (2008) Diffusion least-mean squares over adaptive networks: formulation and performance analysis. *IEEE Trans Signal Process* 56(7):3122–3136
- Lu J, Tang C (2012) Zero-gradient-sum algorithms for distributed convex optimization: the continuous-time case. *IEEE Trans Autom Control* 57(9):2348–2354
- Martinoli A, Mondada F, Mermoud G, Correll N, Egerstedt M, Hsieh A, Parker L, Stoy K (2013) Distributed autonomous robotic systems. Springer tracts in advanced robotics. Springer, Heidelberg/New York
- Mesbahi M, Egerstedt M (2010) Graph theoretic methods for multiagent networks. Princeton University Press, Princeton
- Moreau L (2005) Stability of multiagent systems with time-dependent communication links. *IEEE Trans Autom Control* 50(2):169–182
- Nedić A (2011) Asynchronous broadcast-based convex optimization over a network. *IEEE Trans Autom Control* 56(6):1337–1351
- Nedić A, Bertsekas D (2000) Convergence rate of incremental subgradient algorithms. In: Uryasev S, Pardalos P (eds) Stochastic optimization: algorithms and applications. Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 263–304
- Nedić A, Bertsekas D (2001) Incremental subgradient methods for nondifferentiable optimization. *SIAM J Optim* 56(1):109–138
- Nedić A, Olshevsky A (2013) Distributed optimization over time-varying directed graphs. Available at <http://arxiv.org/abs/1303.2289>
- Nedić A, Ozdaglar A (2007) On the rate of convergence of distributed subgradient methods for multi-agent optimization. In: Proceedings of IEEE CDC, New Orleans, pp 4711–4716
- Nedić A, Ozdaglar A (2009) Distributed subgradient methods for multi-agent optimization. *IEEE Trans Autom Control* 54(1):48–61
- Nedić A, Ozdaglar A (2010) Cooperative distributed multi-agent optimization. In: Eldar Y, Palomar D (eds) Convex optimization in signal processing and communications. Cambridge University Press, Cambridge/New York, pp 340–386
- Nedić A, Bertsekas D, Borkar V (2001) Distributed asynchronous incremental subgradient methods. In: Butnariu D, Censor Y, Reich S (eds) Inherently parallel algorithms in feasibility and optimization and their applications. Studies in computational mathematics. Elsevier, Amsterdam/New York
- Nedić A, Olshevsky A, Ozdaglar A, Tsitsiklis J (2008) Distributed subgradient methods and quantization effects. In: Proceedings of 47th IEEE conference on decision and control, Cancún, Dec 2008, pp 4177–4184
- Nedić A, Ozdaglar A, Parrilo PA (2010) Constrained consensus and optimization in multi-agent networks. *IEEE Trans Autom Control* 55(4):922–938
- Nesterov Y (2005) Primal-dual subgradient methods for convex problems, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain (UCL), Technical report 67
- Olfati-Saber R, Murray R (2004) Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans Autom Control* 49(9):1520–1533
- Olshevsky A (2010) Efficient information aggregation for distributed control and signal processing. Ph.D. dissertation, MIT
- Olshevsky A, Tsitsiklis J (2006) Convergence rates in distributed consensus averaging. In: Proceedings of IEEE CDC, San Diego, pp 3387–3392
- Olshevsky A, Tsitsiklis J (2009) Convergence speed in distributed consensus and averaging. *SIAM J Control Optim* 48(1):33–55
- Patterson S, Bamieh B, Abbadi A (2009) Distributed average consensus with stochastic communication failures. *IEEE Trans Signal Process* 57:2748–2761
- Rabbat M, Nowak R (2004) Distributed optimization in sensor networks. In: Symposium on information processing of sensor networks, Berkeley, pp 20–27

- Ram SS, Nedić A, Veeravalli V (2009) Incremental stochastic sub-gradient algorithms for convex optimization. *SIAM J Optim* 20(2):691–717
- Ram SS, Nedić A, Veeravalli VV (2010a) Distributed stochastic sub-gradient projection algorithms for convex optimization. *J Optim Theory Appl* 147:516–545
- Ram S, Nedić A, Veeravalli V (2010b) Asynchronous gossip algorithms for stochastic optimization: constant stepsize analysis. In: *Recent advances in optimization and its applications in engineering: the 14th Belgian-French-German conference on optimization (BFG)*. Springer-Verlag, Berlin Heidelberg, pp 51–60
- Ram SS, Nedić A, Veeravalli VV (2012) A new class of distributed optimization algorithms: application to regression of distributed data. *Optim Method Softw* 27(1):71–88
- Ren W, Beard R (2005) Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Trans Autom Control* 50(5):655–661
- Srivastava K (2011) Distributed optimization with applications to sensor networks and machine learning. Ph.D. dissertation, University of Illinois at Urbana-Champaign, Industrial and Enterprise Systems Engineering
- Srivastava K, Nedić A (2011) Distributed asynchronous constrained stochastic optimization. *IEEE J Sel Top Signal Process* 5(4):772–790
- Srivastava K, Nedić A, Stipanović D (2010) Distributed constrained optimization over noisy networks. In: *Proceedings of the 49th IEEE conference on decision and control (CDC)*, Atlanta, pp 1945–1950
- Tseng P (1998) An incremental gradient(-projection) method with momentum term and adaptive stepsize rule. *SIAM J Optim* 8:506–531
- Tsianos K (2013) The role of the network in distributed optimization algorithms: convergence rates, scalability, communication/computation tradeoffs and communication delays. Ph.D. dissertation, McGill University, Department of Electrical and Computer Engineering
- Tsianos K, Rabbat M (2011) Distributed consensus and optimization under communication delays. In: *Proceedings of Allerton conference on communication, control, and computing*, Monticello, pp 974–982
- Tsianos K, Lawlor S, Rabbat M (2012a) Consensus-based distributed optimization: practical issues and applications in large-scale machine learning. In: *Proceedings of the 50th Allerton conference on communication, control, and computing*, Monticello
- Tsianos K, Lawlor S, Rabbat M (2012b) Push-sum distributed dual averaging for convex optimization. In: *Proceedings of the IEEE conference on decision and control*, Maui
- Tsitsiklis J (1984) Problems in decentralized decision making and computation. Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology
- Tsitsiklis J, Bertsekas D, Athans M (1986) Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans Autom Control* 31(9):803–812
- Touri B (2011) Product of random stochastic matrices and distributed averaging. Ph.D. dissertation, University of Illinois at Urbana-Champaign, Industrial and Enterprise Systems Engineering
- Touri B, Nedić A (2009) Distributed consensus over network with noisy links. In: *Proceedings of the 12th international conference on information fusion*, Seattle, pp 146–154
- Tu S-Y, Sayed A (2012) Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. <http://arxiv.org/abs/1205.3993>
- Vicsek T, Czirok A, Ben-Jacob E, Cohen I, Schochet O (1995) Novel type of phase transitions in a system of self-driven particles. *Phys Rev Lett* 75(6):1226–1229
- Wan P, Lemmon M (2009) Event-triggered distributed optimization in sensor networks. In: *Symposium on information processing of sensor networks*, San Francisco, pp 49–60
- Wang J, Elia N (2011) A control perspective for centralized and distributed convex optimization. In: *IEEE conference on decision and control*, Florida, pp 3800–3805
- Wei E, Ozdaglar A (2012) Distributed alternating direction method of multipliers. In: *Proceedings of the 51st IEEE conference on decision and control and European control conference*, Maui
- Zanella F, Varagnolo D, Cenedese A, Pillonetto G, Schenato L (2011) Newton-Raphson consensus for distributed convex optimization. In: *IEEE conference on decision and control*, Florida, pp 5917–5922
- Zargham M, Ribeiro A, Ozdaglar A, Jadbabaie A (2014) Accelerated dual descent for network flow optimization. *IEEE Trans Autom Control* 59(4):905–920
- Zhu M, Martínez S (2012) On distributed convex optimization under inequality and equality constraints. *IEEE Trans Autom Control* 57(1):151–164

Distributed Sensing for Monitoring Water Distribution Systems

Lina Sela¹ and Waseem Abbas²

¹Department of Civil, Architectural and Environmental Engineering, The University of Texas at Austin, Austin, TX, USA

²Vanderbilt University, Nashville, TN, USA

Abstract

The primary goal of sensing technologies for water systems is to monitor the quantity and quality of water and to collect information pertaining to the physical state of system

components. Combined with advanced modeling and computational tools, data collected from various sensing devices distributed throughout the water network allows for a real-time decision support system for analyzing, modeling, and controlling water supply systems. Since only a limited number of sensors and monitoring devices can be deployed due to physical constraints and limited budget, a crucial design aspect is to determine the best possible locations for these devices within a network. Optimal sensor placement problems in turn depend on several factors including performance objectives under consideration, type of data and measurements being collected from sensors, sensing and detection models used, water distribution system dynamics, and other deployment challenges. In this entry, we provide an overview of these important aspects of sensors deployment in water distribution networks and discuss applications and usefulness of continuously monitoring parameters associated with water system hydraulics and water quality. We also highlight major challenges and future directions of research in this area.

Keywords

Water distribution systems · Sensor placement · Hydraulic and water quality · Event detection

Introduction

Water distribution systems (WDSs) are complex infrastructure systems that span large geographical distances and are responsible for providing water from few points of source (PoS) to numerous points of use (PoU) through thousands to tens of thousands of hydraulic components of the WDS. Intelligent water systems aided by sensing technologies have been identified as a primary mechanism enabling resilient urban systems (Sela Perelman et al. 2016; Sela and Amin 2018). Continuous monitoring of the state of the WDS, PoS, and PoU can support network

operation in terms of real-time flow distribution, pressure management, water quality, and water loss control. Traditionally, the most common hydraulic sensors in water infrastructure systems measure the water level in storage tanks and flow and pressure in pipelines, whereas chlorine, total organic carbon, and pH levels are the focus of the most common water quality sensors (US EPA 2018). Management of modern infrastructures needs advanced sensing technologies, automated data transfer, processing, and computing to provide useful information to water managers to support decision-making. Among many technological components of a modern WDS, this entry focuses on technologies for sensing the performance of hydraulic components and water quality.

Traditionally, the Supervisory Control and Data Acquisition (SCADA) system provides hydraulic and water quality (H&WQ) measurements at remote PoS such as pumping stations, storage tanks, and water treatment facilities. SCADA system keeps operators informed of the states of the water system's key components. Using this information, operators can then react to the conditions and perform control actions, such as emergency shutdowns of processes, starting or stopping pumps, and opening or closing valves. However, due to a limited spatial coverage of the traditional SCADA systems, whose measurements include information only at the few PoS, the collected information prevents monitoring network-wide state of the WDS. It follows that unlike the PoS, the WDSs are highly unmonitored given the small amount of sensing compared to its large scale and huge number of components. On the consumer side, that is, at the PoU, advanced metering infrastructure (AMI) automatically collects and transmits water consumption information at fine temporal-spatial scales. This information has a potential to improve demand management including characterizing end-users and end-uses, modeling demand, identifying opportunities for technology implementation, and guiding infrastructure management and investment decisions. Today, however, AMIs in water infrastructure are primarily used for improved

billing and for limited sharing of information with end-users.

Rapid developments in wireless/wired sensor networks (WSN) for various applications in the built environment have motivated the application of WSN for monitoring WDS, employing different sensors capable of continuously collecting and transmitting H&WQ measurements. In urban WDS, motivating examples of intelligent systems include the PipeNet@Boston (Stoianov et al. 2007), WaterWise@Singapore (Allen et al. 2011), SWND@Bristol (Wright et al. 2015), and WATSUP@Austin (Xing and Sela 2019). Such deployments extend the capabilities of traditional SCADA systems by providing real-time information at a fine spatial-temporal resolution that was previously unavailable and improve water management by monitoring pressure and water quality (Aisopou et al. 2012; Sela and Amin 2018). The application of WSN in water supply systems for securing H&WQ principally focuses on (1) selecting measurable H&WQ parameters that are good indicators of system state and the corresponding sensing technology, (2) deciding on the number and locations of sensors in a WDS, and (3) performing continuous analysis of the data combined with other available information and models for state estimation, event detection, and forecasting. We discuss these important aspects in this entry with a focus on the sensor placement problem in WDS.

In the next section, we discuss some of the common challenges in managing modern WDS that can greatly benefit from advances in sensing technology. We then briefly review common event detection and sensor placement models for monitoring WDS and also provide an overview of challenges faced in sensors deployment. Finally, we present some future directions in this area.

Managing WDS with Sensors

We begin by highlighting some examples of the usefulness of information collected from online H&WQ sensors in routine and emergency operation of water systems. Although traditionally, problems of managing the hydraulic and water

quality integrity of WDSs are treated separately, water quality practices heavily rely on hydraulic applications.

Model calibration is a process of adjusting model parameters to make model prediction consistent with measured data (Ostfeld et al. 2009). The most common WDS parameters for calibration are roughness of pipes, consumer demands, and pump characteristics. While more information can reduce the uncertainty about demands and pump characteristics, the roughness of pipes is not directly observable. By assuming that some partial information pertaining to flows, pressures, and demands in the WDS is available, various calibration approaches have been proposed including (1) trial-and-error, (2) explicit inverse methods for fully determined systems, and (3) implicit inverse optimization methods for overdetermined systems. Due to the limited data availability, calibration of water quality in WDS has been primarily restricted to simulated case studies (Pasha and Lansey 2009). Many WDS problems rely on calibrated models for their management, and advanced sensing and monitoring can greatly contribute to model calibration.

State estimation typically refers to near-real-time estimation of state variables, for instance, flows, pressures, and concentrations, given some limited observations. In WDS, the state estimation problem typically tries to overcome the limitations of conventional forward hydraulic solvers that allow solving fully determined problems for the set of unknown flows and pressures in the system given known end-user demands, pump and valve settings, and water level in tanks. With increasing availability of sensing technologies in WDS and PoU, system observability can be improved, thus alleviating some modeling assumptions and the need for pseudo-measurements (Preis et al. 2011; Vrachimis et al. 2018).

Real-time control involves optimizing pumping schedules to minimize associated energy costs, valve control for pressure management, and operation of booster disinfectant stations for water quality purposes while satisfying changing demands and operational constraints. Ultimately, real-time control depends on the ability to

perform network H&WQ simulations efficiently, the ability of the optimization problem to find good solutions in near-real-time, the effectiveness of state estimation, and making good forecasts of future water demands (Preis et al. 2011). The latter heavily relies on network observability and the available information from distributed sensors.

Event detection has notably received the most attention by the water systems research community in terms of utilizing the information collected by distributed H&WQ sensors. For water quality, events include intentional and unintentional contaminant intrusion into the WDS. For hydraulics, events include the detection and localization of water leaks and pipe bursts (Abbas et al. 2015; Sela Perelman et al. 2016). Sensors are part of a warning system, which integrates data from different sources, performs analysis of the given information, and raises alarms when an event is detected. Subsequent localization of the event can then inform response and recovery actions. The sensor placement problems for detecting contamination events as well as leaks and bursts in WDSs have been well-studied in the literature (Hart and Murray 2010; Sela Perelman et al. 2016).

Condition assessment and hydraulic integrity of pipelines can be indirectly inferred from pressure readings and used to assess the condition and develop risk indicators (Stephens et al. 2012; Zhang et al. 2018). WDS unsteady response to rapid changes is manifested as transient pressure waves traveling through the system. By monitoring pressure and analyzing the pressure signal, the physical condition of the system can be estimated through inverse transient analysis. Raw pressure readings can be converted using various time-frequency domain techniques into distress indicators of pipe aging and deterioration process, blockages, and presence of leaks, thus providing additional benefits to managing WDSs (Xu and Karney 2017).

Event Detection and Sensor Placement Models

A variety of sensors deployed in WDSs that collect data and information regarding the overall

state of the system can be further used to make inferences about potential H&WQ events as follows:

Water quality – contamination event detection methods rely on identifying abnormal behavior of the measured parameters. Online instrumentation capable of continuously measuring and detecting all possible contaminants is impractical to deploy, but the presence of some contaminants can be inferred through measuring surrogate water quality parameters suitable for online monitoring. The main assumption behind monitoring surrogate parameters is that a contaminant transported with the flow will react with some of the routinely monitored water quality parameters; then, detecting irregularities in monitored water quality parameters can give early indication of a potential contamination event in the WDS. The United States Environmental Protection Agency (EPA), academic institutions, and private companies have been conducting research exploring the range of the water quality parameters that are responsive to a variety of contaminants (e.g., pesticides, insecticides, metals, bacteria) and have provided results from experiments that tested the response of commercial water quality sensors of different designs and technologies to chemical and biological loads (US EPA 2015). These tests suggested that free chlorine, total organic carbon (TOC), oxidation reduction potential (ORP), conductivity, and chloride were the most reactive parameters to the majority of contaminants. Hence, information from online water quality sensors deployed in WDS, which are reactive to many contaminants, can provide an early indication of possible pollution, as opposed to intermittent grab sampling targeting specific contaminants (Perelman et al. 2012).

Hydraulics – WDS hydraulics is governed by the flow continuity and energy conservation principles. Thus, disturbances in flows (e.g., leaks and excessive demands) and structure (e.g., valve closure and pipe isolation) cause changes in flows and pressures in the system. The basic premise of model- and data-based detection models is that flow and pressure sensors that continuously collect and transmit information can be utilized for detecting the abovementioned events (Perez et al. 2014; Puust et al. 2010). Since the effect

of medium and small events is relatively local, many flow and pressure sensors need to be deployed in the WDS to achieve good coverage and localization of the events. More precise and accurate surface and inline direct inspection techniques include acoustic, umbilical, and autonomous robots; however, these tools are mostly suitable for local inspections, and their operation is typically time-consuming and expensive, making them not suitable for continuous operation (Zan et al. 2014).

Once the sensing model is established, the detection model, for both water quality and hydraulics, relies on model-based and/or data-driven detection schemes. Data-driven detection ranges from threshold-based models in time-frequency domain (Srirangarajan et al. 2013; Zan et al. 2014) to k -nearest neighbor, logit models, and genetic algorithms (Housh and Ostfeld 2015). Aside from the specific approach, data-driven models process either single- or multi-parameter signals collected from single- or multiple-sensor locations. Model-based

approaches range from simple (Deshpande et al. 2013) to complex hydraulic and water quality models including single- and multi-species using EPANET and EPANET-MSX (Shang et al. 2008).

As discussed next, in practice, several challenges limit the number of sensors that can be deployed within a system. Consequently, it becomes crucial to place sensors at strategic locations such that they provide the most useful information regarding water quality and hydraulics of the WDS. Before formulating and solving the optimal sensor placement problem, the scope of the problem and its components need to be defined, including determining the WDS dynamics, event and sensing characteristics, as well as the impact and performance models. The common design objectives that can be found in the literature include maximizing detection likelihood, minimizing the time to detection, minimizing the volume of delivered contaminated water, and maximizing event localization. The main steps involved in framing the sensor placement problem are summarized in Table 1.

Distributed Sensing for Monitoring Water Distribution Systems, Table 1 Main steps involved in the sensor placement framework

<i>Design steps</i>	<i>Hydraulic features</i>	<i>Water quality features</i>
Process dynamics	Steady-state hydraulics Unsteady hydraulics	Transport and reaction model Single/multi-species
Event characteristics	Excess loads, reduced connectivity Duration, intensity, location Random, strategic	Contaminant intrusion Duration, magnitude, location Random, strategic
Sensor characteristic	Continuous, intermittent Measurements uncertainty, sensor robustness	Continuous, intermittent Measurements uncertainty, sensor robustness
Detection model	Deterministic, stochastic Distance, threshold, model-based	Deterministic, stochastic Distance, threshold, model-based
Impact measure	Detection likelihood, water loss, energy loss	Detection likelihood, contaminated volume, population at risk
Performance measure	Mean, worst case, value at risk	Mean, worst case, value at risk
Solution approach	Engineering principles Network connectivity Model-driven optimization	Engineering principles Network connectivity Model-driven optimization
Testing and validation	Test-bed Simulation	Test-bed Simulations
Routine management activities	Pressure management Leak detection State estimation	Water quality management Residual disinfectant State estimation
Emergency management activities	Network recovery to emergency event Firefighting conditions	Contaminant intrusion Changes in source water quality

To obtain a sensor placement that optimizes one or multiple performance objectives, a number of solution approaches have been proposed in the literature (Rathi and Gupta 2014). *Mixed integer programming (MIP)*, in which the problem is often formulated similar to facility location, was used to accommodate a wide range of design objectives in the formulation and taking advantage of efficient solution schemes for solving large-scale problems (Berry et al. 2006; Sela and Amin 2018). *Greedy approximation* is another useful solution approach, in which sensor locations are typically selected in an iterative manner. In each iteration, a sensor location that optimizes a single or multiple performance objectives under consideration is selected. In many cases, it is also possible to provide performance guarantees of greedy approximations, for instance, if the objective function is submodular (Krause et al. 2008). *Evolutionary algorithms* have also been successfully employed to obtain sensor placements in water networks (Aral et al. 2009). These solution approaches primarily rely on heuristic search techniques and the integration of hydraulic simulations to optimize various performance objectives. *Graph-theoretic* methods have also been proposed, in which the underlying network topology is explored and various network-based centrality measures are suggested to find the optimal sensor locations (Perelman and Ostfeld 2013).

Challenges in Sensors Deployment

Despite the extensive literature, the application of continuous monitoring within water WDS has been limited in practice for several reasons, some of which include:

Cost – high initial investment, operations, and management costs limit the widespread deployment of sensors by budget-constrained water utilities.

Accuracy and reliability – although better and cheaper water quality sensors suitable for continuous deployment in the field are continuously being developed and improved, the low level of reliability of the generated data, frequent calibration requirements, and demanding power needs

introduce additional challenges in field implementation (Aisopou et al. 2012).

Physical constraints – since most of the hydraulic components are buried underground, physical access to locations of interest, while minimizing interference and disruptions to routine operations, additionally hinders the deployment of sensing apparatus within the WDS.

Data management – integration of distributed sensing increases the amount and variety of data generated. For example, sampling water consumption at the PoU at one minute resolution implies replacing the typical 12 annual meter readings with 525,600 readings (Shafiee et al. 2018). Unavoidably, several related challenges in collecting and managing the data are introduced including data heterogeneity necessitates advanced warehousing, cloud dependency requires constant accessibility to the data, and errors can arise from various sources including hardware, firmware, and communication.

Security and privacy – sabotaging and disrupting WDS, physically or through cyber means, could result in catastrophic consequences on population health and dependent infrastructure. Moreover, the access to vast amounts of data generated from various sensing devices across the system must be controlled and regulated to prevent any misuse and to safeguard users' privacy.

Advanced sensing and metering promises benefits to water utilities by enabling intelligent operation and planning decisions. Nevertheless, it is crucial to understand the trade-offs between the information gained from distributed sensors and the associated challenges in deployment and operation. In summary, currently the benefits of smart sensing technologies for WDS have not been realized to their full potential, and there is a need for proven methods for incorporating these cyber-systems in current decision-making processes.

Summary and Future Directions

Information and data collected through an assortment of sensing devices enable water system

operators to continuously monitor, control, and effectively deal with any disruptive events in real time. Despite the many opportunities that advanced sensing and metering technologies create, in the context of WDS, the sensor deployment for contamination events has indisputably received the most attention, while there are still several future research directions in the area of sensor placement in WDS, as we briefly outline next.

Resilience to cyber-physical attacks – in smart water networks, sensing devices are a part of closed, real-time control loops, and hence observed data from them can directly impact the operation of physical infrastructure assets (pumps, valves, tanks). By exploiting vulnerabilities, a malicious attacker can compromise a subset of these devices and attain a certain level of control to manipulate physical processes. Thus, a resilient sensor network that maintains the task of transmitting correct sensor data with minimal and tolerable disruption despite adversarial attack is crucial for a safe water infrastructure. There has been increased focus in recent years toward security and resilience of water distribution systems to malicious disruption (Perelman and Amin 2014; Taormina et al. 2017; Adepu et al. 2019). There is a need to develop analytical tools to assess vulnerabilities and risks associated with malicious attacks on WDS.

Heterogeneous sensors – in WDSs, heterogeneity of sensors can be interpreted and exploited in many different ways for a better performance. Sensors can be heterogeneous in terms of parameters they measure and information they collect about the system's state, such as water flow, pressure, and quality. Another aspect of heterogeneity is that sensors measuring the same network parameter can be of different types and have multiple variants. For instance, they might have distinct operating systems, software packages, and hardware platforms. Diverse implementations of sensing devices can be conducive toward improving the overall security of the sensor network against cyberattacks (Laszka et al. 2019). Sensors can also be heterogeneous in terms of the level of information they provide (Abbas et al. 2015). For instance,

sensors deployed to detect the disturbance in the form of a pressure wave caused by a pipe burst can be single-level in which case they only report the existence of some disturbance without much detail, or they can be multi-level in which case they also report the physical features of the disturbance, such as the intensity, duration, and frequency of the detected disturbance. This information can be helpful in distinguishing between different events. Sensors with better accuracy and more information are typically expensive. Thus, to better exploit cost versus performance trade-off, employing a combination of more expensive and low-priced sensors could yield better results as compared to using only one type of sensors. Interesting problems would then be to determine the right distribution and locations of expensive but highly accurate and low-priced but relatively less accurate sensors within the network. Incorporating these factors in the design of sensor networks for WDS can lead to more effective and applicable solutions in practice.

Modeling framework and benchmarking – optimal sensor locations with respect to any performance criteria highly depend on the selection of network flow, event (disturbance), and sensing models considered during the problem formulation. Several variants of these models have been developed over the years and are used by researchers and practitioners (Hart and Murray 2010). However, there has been little effort in establishing standardized conditions for such formulations, for instance, what hydraulic dynamics should be used? What is the right level of abstraction regarding the network structure? Which event propagation dynamics should be considered? To have sensor placement solutions that work well in practice, it is crucial to clearly understand, outline, and compare the scope of various modeling frameworks. Moreover, although a myriad of optimization tools have been developed for optimal sensor placement, little effort has been made to make these advances accessible to the practitioners, and these tools typically remain limited to the domain of their developers. Some noteworthy freely available software for water quality event detection are

CANARY and PECOS (US EPA 2012; Klise and Stein 2016) and TEVA-SPOT (Berry et al. 2012) for sensor placement.

Cross-References

- [Cyber-Physical Security](#)
- [Demand-Driven Automatic Control of Irrigation Channels](#)
- [Fault Detection and Diagnosis](#)

Bibliography

- Abbas W, Perelman LS, Amin S, Koutsoukos X (2015) An efficient approach to fault identification in urban water networks using multi-level sensing. In: Proceedings of the 2nd ACM international conference on embedded systems for energy-efficient built environments (BuildSys), pp 147–156
- Adepu S, Palleti VR, Mishra G, Mathur A (2019) Investigation of cyber attacks on a water distribution system. arXiv preprint arXiv:1906.02279
- Aisopou A, Stoianov I, Graham NJ (2012) In-pipe water quality monitoring in water supply systems under steady and unsteady state flow conditions: a quantitative assessment. *Water Res* 46(1):235–246
- Allen M, Preis A, Iqbal M, Stitangarajan S, Lim HN, Girod L, Whittle AJ (2011) Real time in-network monitoring to improve operational efficiency. *J Am Water Works Assoc* 103(7):63–75
- Aral MM, Guan J, Maslia ML (2009) Optimal design of sensor placement in water distribution networks. *J Water Resour Plan Manag* 136(1):5–18
- Berry J, Hart W, Phillips C, Uber J, Watson J (2006) Sensor placement in municipal water networks with temporal integer programming models. *J Water Resour Plan Manag* 132(4):218–224
- Berry J, Riesen LA, Hart WE, Phillips CA, Watson J-P (2012) TEVA-SPOT toolkit and user's manual – version 2.5.2. Technical report EPA/600/R-08/041B, Sandia National Laboratories
- Deshpande A, Sarma SE, Youcef-Toumi K, Mekid S (2013) Optimal coverage of an infrastructure network using sensors with distance-decaying sensing quality. *Automatica* 49(11):3351–3358
- Hart WE, Murray R (2010) Review of sensor placement strategies for contamination warning systems in drinking water distribution systems. *J Water Resour Plan Manag* 136(6):611–619
- Housh M, Ostfeld A (2015) An integrated logit model for contamination event detection in water distribution systems. *Water Res* 75:210–223
- Klise K, Stein J (2016) Performance monitoring using pecos. Technical report SAND2016-3583, Sandia National Laboratories
- Krause A, Leskovec J, Guestrin C, Vanbriesen JM, Faloutsos C (2008) Efficient sensor placement optimization for securing large water distribution networks. *J Water Resour Plan Manag* 134(6):516–526
- Laszka A, Abbas W, Vorobeychik Y, Koutsoukos X (2019) Integrating redundancy, diversity, and hardening to improve security of industrial internet of things. *Cyber-Phys Syst* 6:1–32
- Ostfeld A, Uber JG, Salomons E, Berry JW, Hart WE, Phillips CA, Watson J-P, Dorini G, Jonkergouw P, Kapelan Z, Pierro F (2009) The battle of the water sensor networks (BWSN): a design challenge for engineers and algorithms. *J Water Resour Plan Manag* 134(6):556–568
- Pasha MF, Lansey K (2009) Water quality parameter estimation for water distribution systems. *Civ Eng Environ Syst* 26(3):231–248
- Perelman L, Amin S (2014) A network interdiction model for analyzing the vulnerability of water distribution systems. In: Proceedings of the 3rd international conference on high confidence networked systems. ACM, pp 135–144
- Perelman L, Ostfeld A (2013) Application of graph theory to sensor placement in water distribution systems. In: World environmental and water resources congress 2013, pp 617–625
- Perelman L, Arad J, Housh M, Ostfeld A (2012) Event detection in water distribution systems from multivariate water quality time series. *Environ Sci Technol* 46(15):8212–8219
- Perez R, Sanz G, Puig V, Quevedo J, Cuguelero Escofet M, Nejari F, Meseguer J, Cembrano G, Mirats Tur J, Sarrate R (2014) Leak localization in water networks: a model-based methodology using pressure sensors applied to a real network in barcelona. *IEEE Control Syst* 34(4):24–36
- Preis A, Whittle AJ, Ostfeld A, Perelman L (2011) Efficient hydraulic state estimation technique using reduced models of urban water networks. *J Water Resour Plan Manag* 137(4):343–351
- Puust R, Kapelan Z, Savic DA, Koppel T (2010) A review of methods for leakage management in pipe networks. *Urban Water J* 7(1):25–45
- Rathi S, Gupta R (2014) Sensor placement methods for contamination detection in water distribution networks: a review. In: Proceedings of 16th conference on water distribution system analysis
- Sela L, Amin S (2018) Robust sensor placement for pipeline monitoring: mixed integer and greedy optimization. *Adv Eng Inform* 36:55–63
- Sela Perelman L, Abbas W, Koutsoukos X, Amin S (2016) Sensor placement for fault location identification in water networks: a minimum test cover approach. *Automatica* 72:166–176
- Shafiee E, Barker Z, Rasekh A (2018) Enhancing water system models by integrating big data. *Sustain Cities Soc* 37:485–491

- Shang F, Uber JG, Rossman LA (2008) Modeling reaction and transport of multiple species in water distribution systems. *Environ Sci Technol* 42(3): 808–814
- Srirangarajan S, Allen M, Preis A, Iqbal M, Lim HB, Whittle AJ (2013) Wavelet-based burst event detection and localization in water distribution systems. *J Signal Process Syst* 72(1): 1–16
- Stephens ML, Lambert MF, Simpson AR (2012) Determining the internal wall condition of a water pipeline in the field using an inverse transient. *J Hydraul Eng* 139(3):310–324
- Stoianov I, Nachman L, Madden S, Tokmouline T (2007) PIPENET a wireless sensor network for pipeline monitoring. In: *Proceedings of the 6th international conference on information processing in sensor networks, IPSN'07*. ACM, pp 264–273
- Taormina R, Galelli S, Tippenhauer NO, Salomons E, Ostfeld A (2017) Characterizing cyber-physical attacks on water distribution systems. *J Water Resour Plan Manag* 143(5):04017009
- US EPA (2012) Canary user's manual version 4.3.2. Technical report EPA 600-R-08-040B, U.S. Environmental Protection Agency: Office of Water
- US EPA (2015) Summary of implementation approaches and lessons learned from the water security initiative contamination warning system pilots. Technical report EPA 817-R-15-002, U.S. Environmental Protection Agency: Office of Water
- US EPA (2018) Online water quality monitoring in distribution systems. Technical report EPA 817-B-18-001, U.S. Environmental Protection Agency: Office of Water
- Vrachimis SG, Eliades DG, Polycarpou MM (2018) Real-time hydraulic interval state estimation for water transport networks: a case study. *Drink Water Eng Sci* 11(1):19–24
- Wright R, Abraham E, Pappas P, Stoianov I (2015) Control of water distribution networks with dynamic DMA topology using strictly feasible sequential convex programming. *Water Resour Res* 51(12): 9925–9941
- Xing L, Sela L (2019) Unsteady pressure patterns discovery from high-frequency sensing in water distribution systems. *Water Res* 158: 291–300
- Xu X, Karney B (2017) An overview of transient fault detection techniques. In: *Modeling and monitoring of pipelines and networks*. Springer, pp 13–37
- Zan TTT, Lim HB, Wong K-J, Whittle AJ, Lee B-S (2014) Event detection and localization in urban water distribution network. *IEEE Sensors J* 14(12):4134–4142
- Zhang C, Gong J, Zecchin A, Lambert M, Simpson A (2018) Faster inverse transient analysis with a head-based method of characteristics and a flexible computational grid for pipeline condition assessment. *J Hydraul Eng* 144(4):04018007

Disturbance Observers

Hyungbo Shim

Department of Electrical and Computer Engineering, Seoul National University, Seoul, Republic of Korea

Abstract

Disturbance observer is an inner-loop output-feedback controller whose role is to reject external disturbances and to make the outer-loop baseline controller robust against plant's uncertainties. Therefore, the closed-loop system with the DOB approximates the nominal closed-loop by the baseline controller and the nominal plant model with no disturbances. This article presents how the disturbance observer works under what conditions and how one can design a disturbance observer to guarantee robust stability and to recover the nominal performance not only in the steady state but also for the transient response under large uncertainty and disturbance.

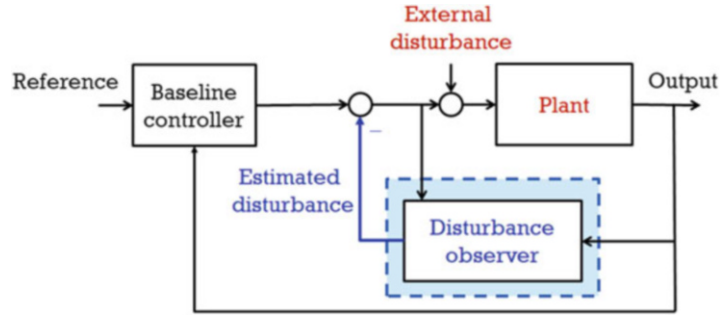
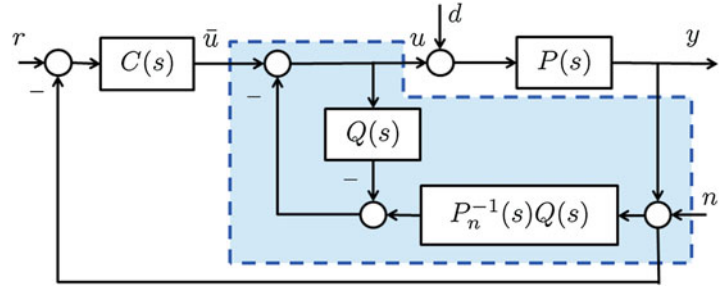
Keywords

Robust stabilization · Robust transient response · Disturbance attenuation · Singular perturbation · Normal form

Introduction

The term *disturbance observer* has been used for a few different algorithms and methods in the literature. In this article, we restrict the disturbance observer to imply the inner-loop controller as described conceptually in Fig. 1. This is one that has been employed in many practical applications and verified in industry and is often simply called “DOB.”

The primary goal of DOB is to estimate the external disturbance at the input stage of the plant, which is then used to counteract the exter-

Disturbance Observers,**Fig. 1** Disturbance observer as an inner-loop feedback controller**Disturbance Observers,****Fig. 2** Disturbance observer for a linear plant

nal disturbance as in Fig. 1. This initial idea is extended to deal with the plant's uncertainties when uncertain terms can be lumped into the external disturbance. Therefore, DOB is considered as a method for robust control. More specifically, DOB robustifies the (possibly non-robust) baseline controller. The baseline controller is supposed to be designed for a nominal model of the plant that does not have external disturbances nor uncertainties. By inserting the DOB in the inner-feedback loop, the nominal stability and performance that would have been obtained by the baseline controller and the nominal plant, can be approximately recovered even in the presence of disturbances and uncertainties. There is effectively no restriction on the baseline controller as long as it stabilizes the nominal plant.

When there is no uncertainty and no disturbance with a DOB being equipped, the closed-loop system recovers the nominal performance completely, and as the amount of uncertainty and disturbance grows, the performance degrades gradually while it is still close to the nominal performance. This is in contrast to other robust controls based on the worst-case design, which sacrifices the nominal performance for the worst uncertainty.

Disturbance Observer for Linear System

Let $P(s)$ be the transfer function of a unknown linear plant, $P_n(s)$ be its nominal model, and $C(s)$ be a baseline controller designed for the nominal model $P_n(s)$. Then, the closed-loop system with the DOB can be depicted as Fig. 2. In the figure, $Q(s)$ is a stable low-pass filter whose dc gain is one. The relative degree (i.e., the order of denominator minus the order of numerator) of $Q(s)$ is greater than or equal to the relative degree of $P_n(s)$, so that the block $P_n^{-1}(s)Q(s)$ is proper and implementable. The signals d and r are the external disturbance and the reference, respectively, which are assumed to have little high-frequency components, and the signal n is the measurement noise.

From Fig. 2, the output y in the frequency domain is written as

$$y(s) = \frac{P_n P C}{P_n(1 + PC) + Q(P - P_n)} r(s) + \frac{P_n P(1 - Q)}{P_n(1 + PC) + Q(P - P_n)} d(s) - \frac{P(Q + P_n C)}{P_n(1 + PC) + Q(P - P_n)} n(s). \quad (1)$$

Assume that all the transfer functions are stable (i.e., all the poles have negative real parts), and let ω_c be the cutoff frequency of the low-pass filter so that $Q(j\omega) \approx 1$ for $\omega \ll \omega_c$ and $Q(j\omega) \approx 0$ for $\omega \gg \omega_c$. Therefore, it is seen from (1) that, for $\omega \ll \omega_c$,

$$y(j\omega) \approx \frac{P_n C}{1 + P_n C} r(j\omega) - n(j\omega) \quad (2)$$

and for $\omega \gg \omega_c$ where $d(j\omega) \approx 0$ and $r(j\omega) \approx 0$,

$$\begin{aligned} y(j\omega) &\approx \frac{PC}{1 + PC} r(j\omega) \\ &\quad + \frac{P}{1 + PC} d(j\omega) \\ &\quad - \frac{PC}{1 + PC} n(j\omega) \\ &\approx -\frac{PC}{1 + PC} n(j\omega). \end{aligned}$$

The property (2) is of particular interest because the input-output relation recovers the nominal closed-loop system without being affected by the disturbance d .

The low-pass filter $Q(s)$ is often called “Q-filter” and it is typically given by

$$Q(s) = \frac{a_0}{(\tau s)^\nu + a_{\nu-1}(\tau s)^{\nu-1} + \dots + a_1(\tau s) + a_0}$$

where ν is the relative degree of $P_n(s)$, the coefficients a_i are chosen such that $s^\nu + a_{\nu-1}s^{\nu-1} + \dots + a_0$ be a Hurwitz polynomial, and the constant $\tau > 0$ determines the cutoff frequency.

Robust Stability Condition

The beneficial property (2) is obtained under the assumption that the transfer functions in (1) are stable for all possible uncertain plants $P(s)$. Therefore, it is necessary to design $Q(s)$ (or, choose τ and a_i 's) such that the closed-loop system remains stable for all possible $P(s)$.

In order to deal with uncertain plants having parametric uncertainty, consider the set $\mathcal{P}$ of uncertain transfer functions

$$\mathcal{P} = \left\{ P(s) = g \frac{s^{n-\nu} + \beta_{n-\nu-1}s^{n-\nu-1} + \dots + \beta_0}{s^n + \alpha_{n-1}s^{n-1} + \dots + \alpha_0} : \alpha_i \in [\underline{\alpha}_i, \bar{\alpha}_i], \beta_i \in [\underline{\beta}_i, \bar{\beta}_i], g \in [\underline{g}, \bar{g}] \right\}$$

where the intervals $[\underline{\alpha}_i, \bar{\alpha}_i]$, $[\underline{\beta}_i, \bar{\beta}_i]$, and $[\underline{g}, \bar{g}]$ are known, and $\underline{g} > 0$.

Theorem 1 (Shim and Jo 2009) Assume that (a) the nominal model P_n belongs to $\mathcal{P}$ and the baseline controller C internally stabilizes P_n , (b) all transfer functions in $\mathcal{P}$ are minimum phase, and (c) the coefficients a_i of Q are chosen such that

$$p_f(s) := s^\nu + a_{\nu-1}s^{\nu-1} + \dots + a_1s + \frac{g}{g^*}a_0$$

is Hurwitz for all $g \in [\underline{g}, \bar{g}]$, where $g^* \in [\underline{g}, \bar{g}]$ is the nominal value of g , then there exists $\tau^* > 0$ such that, for all $\tau \leq \tau^*$, the transfer functions in (1) are stable for all $P(s) \in \mathcal{P}$.

The value of τ^* can be computed from the knowledge of the bounds of the intervals in $\mathcal{P}$ (Shim and Jo 2009), but it may also be conservatively chosen based on repeated simulations in practice. Smaller τ implies larger bandwidth of Q-filter, which is desired in the sense that the property (2) holds for larger frequency range.

The proof of Theorem 1 proceeds by observing the closed-loop system in Fig. 2 has $2n + m$ poles where m is the order of $C(s)$. Then one can inspect the behavior of those poles by changing the design parameter τ . For this, let us denote the poles by $\lambda_i(\tau)$, $i = 1, \dots, 2n + m$. Then, it can be shown that

$$\lim_{\tau \rightarrow 0} \tau \lambda_i(\tau) = \lambda_i^*, \quad i = 1, \dots, \nu \quad (3)$$

where λ_i^* , $i = 1, \dots, \nu$, are the roots of $p_f(s)$, and

$$\lim_{\tau \rightarrow 0} \lambda_i(\tau) = \lambda_i^*, \quad i = \nu + 1, \dots, 2n + m$$

where λ_i^* , $i = \nu + 1, \dots, n$, are the zeros of $P(s)$, and λ_i^* , $n + 1, \dots, 2n + m$, are the poles of the nominal closed-loop with $P_n(s)$ and $C(s)$. Thus, Theorem 1 follows.

This argument shows that, if there is no λ_i^* on the imaginary axis, then the conditions (a), (b), and (c) are also necessary for robust stability with sufficiently small τ . It is also seen by (3) that, when $p_f(s)$ is Hurwitz, the poles $\lambda_i(\tau)$, $i = 1, \dots, v$, escape to the negative infinity as $\tau \rightarrow 0$. Therefore, with large bandwidth of $Q(s)$, the closed-loop system shows two-time scale behavior. In fact, it turns out that $p_f(s)$ is the characteristic polynomial of the fast dynamics called “boundary layer system” in the singular perturbation analysis with τ being the parameter of singular perturbation.

Design of $Q(s)$ for Robust Stability

In order to satisfy the condition (c) of Theorem 1, one can choose $\{a_i : i = 1, \dots, v-1\}$ such that

$$s^{v-1} + a_{v-1}s^{v-2} + \dots + a_2s + a_1$$

be a Hurwitz polynomial, and then pick $a_0 > 0$ sufficiently small. Then, the polynomial $p_f(s)$ remains Hurwitz for all variation of $g \in [g, \bar{g}]$.

This can be justified by, for example, the circle criterion. With $G(s) := a_0/(s^v + a_{v-1}s^{v-1} + \dots + a_1s)$ and a static gain (g/g^*) that belongs to the sector $[\underline{g}/g^*, \bar{g}/g^*]$, the characteristic polynomial of the closed-loop system $G(s)/(1 + (g/g^*)G(s))$ becomes $s^v + a_{v-1}s^{v-1} + \dots + a_1s + (g/g^*)a_0$. Therefore, if the Nyquist plot of $G(s)$ does not enter nor encircle the closed disk in the complex plane whose diameter is the line segment connecting $-g^*/\underline{g}$ and $-g^*/\bar{g}$, then $p_f(s)$ is Hurwitz for all variation of $g \in [g, \bar{g}]$. Since $G(s)$ has one pole at the origin and the rest poles have negative real parts, its Nyquist plot is bounded to the direction of real axis. Therefore, by choosing a_0 sufficiently small, the Nyquist plot is disjoint from and does not encircle the disk.

Disturbance Observer for Nonlinear System

Intuitive Introduction of the DOB for Nonlinear System

DOB for nonlinear systems inherits all the ingredients and properties of the DOB for linear

systems. The DOB can be constructed for a class of systems that can be represented by the Byrnes-Isidori normal form in certain coordinates as

$$P : \begin{cases} y = x_1, \\ \dot{x}_i = x_{i+1}, & i = 1, \dots, v-1, \\ \dot{x}_v = f(x, z) + g(x, z)(u + d), \\ \dot{z} = h(x, z, d_z) \end{cases}$$

where $u \in \mathbb{R}$ is the input, $y \in \mathbb{R}$ is the measured output, $x = [x_1, \dots, x_v]^T \in \mathbb{R}^v$ and $z \in \mathbb{R}^{n-v}$ are the states of the n -th order system, and unknown external disturbances are denoted by d and d_z . The disturbances and their first time-derivatives are assumed to be bounded. The functions f and g and the vector field h contain uncertainty. The corresponding nominal model of the plant is considered as

$$P_n : \begin{cases} y = x_1, \\ \dot{x}_i = x_{i+1}, & i = 1, \dots, v-1, \\ \dot{x}_v = f_n(x, z) + g_n(x, z)\bar{u} \\ \dot{z} = h_n(x, z, 0) \end{cases}$$

where $\bar{u}$ represents the input to the nominal model, and the nominal f_n , g_n , and h_n are known. Let us assume all functions and vector fields are smooth.

Assumption 1 (a) A baseline feedback controller

$$C : \begin{cases} \dot{\eta} = \Pi(\eta, y) & \in \mathbb{R}^m, \\ \bar{u} = \pi(\eta, y) \end{cases}$$

stabilizes the nominal model P_n .

(b) The zero dynamics $\dot{z} = h(x, z, d_z)$ is input-to-state stable (ISS) from x and d_z to the state z .

The underlying idea of the DOB is that, if one can apply the input u_{desired} to the plant P , which is generated by

$$\begin{aligned} \dot{\bar{z}} &= h_n(x, \bar{z}) \\ u_{\text{desired}} &= -d + \frac{1}{g(x, z)}(-f(x, z) \\ &\quad + f_n(x, \bar{z}) + g_n(x, \bar{z})\bar{u}) \end{aligned} \quad (4)$$

then the plant P with (4) behaves identical to the nominal model P_n plus $\dot{z} = h(x, z, d_z)$. Then, the plant P with (4) interacts with the baseline controller C , and so, it is stabilized, while the plant's zero dynamics $\dot{z} = h(x, z, d_z)$ becomes stand-alone. However, the zero dynamics is ISS, and so, the state z is bounded under the bounded inputs x and d_z .

This idea is, however, not implementable because d , f , and g , as well as the states x and z in (4), are unknown. It turns out that the role of the DOB is to *estimate* the state x and the signal u_{desired} . In the linear case, the structure of the DOB in Fig. 2 actually does this job (Shim et al 2016).

Implementation of the DOB

The idea of estimating x and u_{desired} is realized in the semi-global sense. Suppose that $S_0 \subset \mathbb{R}^{n+m}$ be a compact set for possible initial conditions $(x(0), z(0), \eta(0))$ and U be a compact subset of region of attraction for the nominal closed-loop system P_n and C , which is assumed to contain a compact set S that is strictly larger than S_0 .

Let S_z and U_x be the projections of S and U to the z -axis and the x -axis, respectively, and let M_d and M_{d_z} be an upper bounds for the norms of d and d_z , respectively. Uncertain f , g , and h are supposed to belong to the sets $\mathcal{F}$, $\mathcal{G}$, and $\mathcal{H}$, respectively, which are defined as follows. Let $\mathcal{H}$ be a collection of uncertain h . For each $h \in \mathcal{H}$, there is a compact set $Z_h \subset \mathbb{R}^{n-v}$ to which all possible states $z(t)$ belongs, where $z(t)$ is the solution to the ISS system $\dot{z} = h(u_1, z, u_2)$ with any $z(0) \in S_z$ and any bounded inputs $u_1(t) \in U_x$ and $\|u_2(t)\| \leq M_{d_z}$. It is assumed that $\mathcal{H}$ is such that there is a compact set $Z \supset \cup_{h \in \mathcal{H}} Z_h$. The set $\mathcal{F}$ is a collection of uncertain f such that, for every $f \in \mathcal{F}$, there are uniform bounds M_f and M_{df} such that $|f(x, z)| \leq M_f$ and $\|\partial f(x, z)/\partial(x, z)\| \leq M_{df}$ on $U_x \times Z$. The set $\mathcal{G}$ is a collection of uncertain g such that, for every $g \in \mathcal{G}$, there are uniform bounds $\underline{g}$, $\bar{g}$, and M_{dg} such that $\|\partial g(x, z)/\partial(x, z)\| \leq M_{dg}$ and

$$0 < \underline{g} \leq g(x, z) \leq \bar{g}, \quad \forall (x, z) \in U_x \times Z.$$

The DOB is illustrated in Fig. 3 and is given by

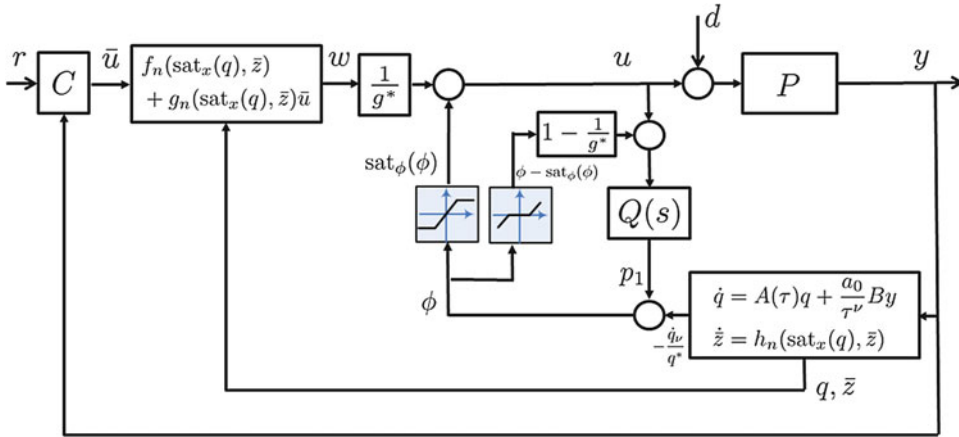
$$D : \begin{cases} \dot{\bar{z}} = h_n(\text{sat}_x(q), \bar{z}) & \in \mathbb{R}^{n-v} \\ \dot{q} = A(\tau)q + \frac{a_0}{\tau}By & \in \mathbb{R}^v \\ \dot{p} = A(\tau)p + \frac{a_0}{\tau}B \left(\phi - \frac{1}{g^*}(\phi - \text{sat}_\phi(\phi)) + \frac{1}{g^*}w \right) & \in \mathbb{R}^v \\ u = \text{sat}_\phi(\phi) + \frac{1}{g^*}w \end{cases}$$

where

$$\phi = p_1 + \frac{1}{g^*} \left[\frac{a_0}{\tau^v}, \frac{a_1}{\tau^{v-1}}, \dots, \frac{a_{v-1}}{\tau} \right] q - \frac{1}{g^*} \frac{a_0}{\tau^v} y \quad (= p_1 - \frac{\dot{q}_v}{g^*})$$

$$w = f_n(\text{sat}_x(q), \bar{z}) + g_n(\text{sat}_x(q), \bar{z})\bar{u}$$

$$A(\tau) = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -\frac{a_0}{\tau^v} & -\frac{a_1}{\tau^{v-1}} & \dots & -\frac{a_{v-1}}{\tau} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$



Disturbance Observers, Fig. 3 Disturbance observer for a nonlinear plant P . The state of $Q(s)$ is p

and g^* is any constant between $\underline{g}$ and $\bar{g}$, $\bar{u}$ is the output of the baseline controller C , and two saturations are globally bounded, continuously differentiable functions such that $\text{sat}_x(x) = x$ for all $x \in U_x$ and $\text{sat}_\phi(\phi) = \phi$ for all $\phi \in S_\phi$ where the interval S_ϕ is given by

$$S_\phi = \left\{ \left(\frac{1}{g(z, x)} - \frac{1}{g^*} \right) (f_n(x, \bar{z}) + g_n(x, \bar{z})\pi(\eta, x_1)) - \frac{f(x, z)}{g(x, z)} - d : z \in Z, (\bar{z}, x, \eta) \in U, |d| \leq M_d, f \in \mathcal{F}, g \in \mathcal{G} \right\}.$$

Suppose that $(\bar{z}(0), q(0), p(0)) \in S_{0z} \times S_{qp}$ where S_{0z} is the projection of S_0 to the z -axis and S_{qp} is any compact set in $\mathbb{R}^{2\nu}$.

In practice, choosing the sets such as U , Z , and S_ϕ may not be easy. If so, conservative choice of them works. Based on repeated simulations or numerical computations, one can take sufficiently large compact sets for them. If everything is linear and the saturations become identity maps in Fig. 3, then the above controller D becomes effectively the same as the state-space realization of the linear DOB in Fig. 2.

Robust Stability

Assumption 1 (c) The coefficients a_i in the DOB are chosen such that $s^{\nu-1} + a_{\nu-1}s^{\nu-2} + \dots + a_1$ is Hurwitz and $a_0 > 0$ is sufficiently

small such that the Nyquist plot of $G(s) := a_0/(s^\nu + a_{\nu-1}s^{\nu-1} + \dots + a_1s)$ is disjoint from and does not encircle the closed disk in the complex plane whose diameter is the line segment connecting $-g^*/\underline{g}$ and $-g^*/\bar{g}$.

Under this assumption, the polynomial

$$p_f(s) := s^\nu + a_{\nu-1}s^{\nu-1} + \dots + a_1s + \frac{g(x, z)}{g^*}a_0$$

is Hurwitz for all frozen states $(x, z) \in U_x \times Z$ and for all uncertain functions $g \in \mathcal{G}$, as discussed in the section “[Design of \$Q\(s\)\$ for Robust Stability](#)”.

Theorem 2 Suppose that the conditions (a), (b), and (c) of Assumption 1 hold. Then, there exists $\tau^* > 0$ such that, for each $\tau \leq \tau^*$, the closed-loop system of P , C , and D is stable for all $f \in \mathcal{F}$, $g \in \mathcal{G}$, and $h \in \mathcal{H}$, and the region of attraction for $(x, z, \eta, \bar{z}, q, p)$ includes $S_0 \times S_{0z} \times S_{qp}$.

To prove the theorem, one can convert the closed-loop system into the standard singular perturbation form with τ being the singular perturbation parameter. Then, it can be seen that the quasi-steady-state system on the slow manifold is simply the nominal closed-loop system of P_n and C without external disturbance d plus the actual zero dynamics $\dot{z} = h(x, z, d_z)$. Since the quasi-steady-state system is assumed to be stable by Assumption 1, the overall system is stable with

sufficiently small τ if the boundary layer system is also stable. Then, it turns out that the boundary layer system is linear since all the slow variables such as $x(t)$ and $z(t)$ are treated as frozen parameters, and the characteristic polynomial of the linear boundary layer system is $p_f(s)$. For details, see Back and Shim (2008).

It can be also seen that, on the slow manifold, all the saturation functions in the DOB become inactive. The role of the saturations is to prevent the peaking phenomenon (Sussmann and Kokotovic 1991) from transferring into the plant. Without saturations, the region of attraction may shrink in general as τ gets smaller, as in Kokotovic and Marino (1986), and only local stability is achieved. On the other hand, even if the plant is protected from the peaking components by the saturation functions, some internal components must peak for fast convergence of the DOB states. In this regard, the role of the dead-zone nonlinearity in Fig. 3 is to allow peaking inside the DOB structure.

Robust Transient Response

Additional benefit of the DOB with saturation functions is robustness of transient response. If the baseline controller C is designed for the nominal model P_n to achieve desired transients such as small overshoot or fast settling time for example, then similar transients can be obtained for the actual plant P under external disturbances by adding the nonlinear DOB to the baseline controller. The same benefit is obtained for the linear case as long as the saturations in Fig. 3 are employed.

Theorem 3 (Back and Shim 2008) *Suppose that the conditions (a), (b), and (c) of Assumption 1 hold. For any given $\epsilon > 0$, there exists $\tau^* > 0$ such that, for each $\tau \leq \tau^*$, the solution of the closed-loop system denoted by $(z(t), x(t), \bar{z}(t), \eta(t))$, initiated in $S_{0z} \times S_0$ with $(q(0), p(0)) \in S_{qp}$, is bounded and satisfies*

$$\left\| \begin{bmatrix} x(t) \\ \bar{z}(t) \\ \eta(t) \end{bmatrix} - \begin{bmatrix} x_N(t) \\ \bar{z}_N(t) \\ \eta_N(t) \end{bmatrix} \right\| \leq \epsilon, \quad \forall t \geq 0$$

where the reference $(x_N(t), \bar{z}_N(t), \eta_N(t))$ is the solution of the nominal closed-loop system of P_n and C with $(x_N(0), \bar{z}_N(0), \eta_N(0)) = (x(0), \bar{z}(0), \eta(0))$.

Since $y = x_1$, this theorem ensures robust transient response that $y(t)$ remains close to its nominal counterpart $y_N(t)$ from the initial time $t = 0$. However, nothing can be said regarding the state $z(t)$ except that it is bounded by the ISS property of the zero dynamics.

Theorem 3 is basically an application of Tikhonov's theorem (Hoppensteadt 1966).

Discussions

- The baseline controller C may depend on a reference r , which is the case for tracking control. Theorems 2 and 3 also hold for this case (Back and Shim 2008).
- If uncertainties are small in modeling so that $f \approx f_n$, $g \approx g_n$, and $h \approx h_n$, then the DOB can be used to estimate the input disturbance d . This is clearly seen from (4).
- Regarding a design of DOB for multi-input-multi-output plants, refer to Back and Shim (2009), which requires well-defined vector relative degree of the plant. For the case of extended state observer (ESO), this requirement has been relaxed in Wu et al (2019).
- If the external disturbance is a sum of a modeled disturbance, which is generated by a known model, and an unmodeled disturbance which is slowly varying, then the modeled disturbance can be exactly canceled by embedding the known model into the DOB structure, while the unmodeled disturbance can still be canceled approximately. This is done by utilizing the internal model principle, and the details can be found in Joo et al (2015).
- For linear systems with mismatched disturbances:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}u + \mathbf{E}d, & \mathbf{x} &\in \mathbb{R}^n, & u &\in \mathbb{R}, \\ y &= \mathbf{c}\mathbf{x}, & \mathbf{d} &\in \mathbb{R}^q, & y &\in \mathbb{R}, \end{aligned}$$

one can transfer the disturbances into the input stage by redefining the state combined with the disturbances, if the disturbance $\mathbf{d}$ is smooth in t . Indeed, there is a coordinate change $[x^T, z^T]^T = \Phi \mathbf{x} + \Theta[\mathbf{d}^T, \dot{\mathbf{d}}^T, \dots, (\mathbf{d}^{(v-2)})^T]^T$ from $\mathbf{x}$ to (x, z) with a nonsingular matrix Φ , and the system becomes

$$\begin{aligned} y &= x_1, & \dot{x}_i &= x_{i+1}, & i &= 1, \dots, v-1, \\ \dot{x}_v &= F_x x + F_z z + g u + g d \\ \dot{z} &= H_x x + H_z z + d_z \end{aligned}$$

where d and d_z are linear combinations of $\mathbf{d}$ and its derivatives (Shim et al 2016).

- Robust control based on the linear DOB is relatively simple to construct, which is one of the reasons why it is frequently used in industry. Stability condition is often ignored in practice, but as seen in Theorem 1, all three conditions are often automatically met. In particular, with small amount of uncertainty, the condition (c) tends to hold with $g \approx g^*$ for any stable Q-filter because of structural robustness of Hurwitz polynomials.
- In the case of linear DOB, there is another robust stability condition based on the small-gain theorem (Choi et al 2003).
- Basic philosophy of DOB is to treat plant's uncertainties and external disturbances together as a lumped disturbance and to estimate and compensate it. This philosophy is shared with other similar approaches such as *extended state observer (ESO)* (Freidovich and Khalil 2008) and *active disturbance rejection control (ADRC)* (Han 2009). The DOB also has been reviewed with comparison to other similar methods in Chen et al (2015).

Summary and Future Directions

Underlying theory for the disturbance observer is presented. The analysis is mainly based on large bandwidth of Q-filter. However, there are cases when the bandwidth cannot be increased in practice because of limited sampling rate in discrete-time implementation. Further study is

necessary to achieve satisfactory performance for discrete-time design of DOB.

Cross-References

- [Nonlinear Zero Dynamics](#)
- [Hybrid Observers](#)
- [Observers for Nonlinear Systems](#)
- [Observers in Linear Systems Theory](#)

Bibliography

- Back J, Shim H (2008) Adding robustness to nominal output-feedback controllers for uncertain nonlinear systems: a nonlinear version of disturbance observer. *Automatica* 44(10):2528–2537
- Back J, Shim H (2009) An inner-loop controller guaranteeing robust transient performance for uncertain MIMO nonlinear systems. *IEEE Trans Autom Control* 54(7):1601–1607
- Chen WH, Yang J, Guo L, Li S (2015) Disturbance-observer-based control and related methods—an overview. *IEEE Trans Ind Electron* 63(2):1083–1095
- Choi Y, Yang K, Chung WK, Kim HR, Suh IH (2003) On the robustness and performance of disturbance observers for second-order systems. *IEEE Trans Autom Control* 48(2):315–320
- Freidovich LB, Khalil HK (2008) Performance recovery of feedback-linearization-based designs. *IEEE Trans Autom Control* 53(10):2324–2334
- Han J (2009) From pid to active disturbance rejection control. *IEEE Trans Ind Electron* 56(3):900–906
- Hoppensteadt FC (1966) Singular perturbations on the infinite interval. *Trans Am Math Soc* 123(2):521–535
- Joo Y, Park G, Back J, Shim H (2015) Embedding internal model in disturbance observer with robust stability. *IEEE Trans Autom Control* 61(10):3128–3133
- Kokotovic P, Marino R (1986) On vanishing stability regions in nonlinear systems with high-gain feedback. *IEEE Trans Autom Control* 31(10):967–970
- Shim H, Jo NH (2009) An almost necessary and sufficient condition for robust stability of closed-loop systems with disturbance observer. *Automatica* 45(1):296–299
- Shim H, Park G, Joo Y, Back J, Jo N (2016) Yet another tutorial of disturbance observer: Robust stabilization and recovery of nominal performance. *Control Theory Technol* 4(3):237–249. Special issue on disturbance rejection: a snapshot, a necessity, and a beginning, preprint = arxiv:1601.02075
- Sussmann H, Kokotovic P (1991) The peaking phenomenon and the global stabilization of nonlinear systems. *IEEE Trans Autom Control* 36(4):424–440
- Wu Y, Isidori A, Lu R, Khalil H (2019, To appear) Performance recovery of dynamic feedback-linearization methods for multivariable nonlinear systems. *IEEE Trans Autom Control*. <https://doi.org/10.1109/TAC.2019.2924176>

DOC

► Discrete Optimal Control

Dynamic Graphs, Connectivity of

Yiannis Kantaros¹, Michael M. Zavlanos², and George J. Pappas³

¹Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA

²Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, USA

³Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA

Abstract

Wireless communication is known to play a pivotal role in enabling teams of robots to successfully accomplish global coordinated tasks. In fact, network connectivity is an underlying assumption in every distributed control and optimization algorithm. For this reason, in recent years, there is a growing research in designing controllers that ensure point-to-point or end-to-end network connectivity for all time. Such controllers either rely on graph theory to model robot communication or employ more realistic communication models that take into account path loss, shadowing, and multipath fading. Nevertheless, maintaining all-time connectivity can severely restrict the robots from accomplishing their tasks, as motion planning is always constrained by network connectivity constraints. Therefore, intermittent connectivity controllers have recently been proposed, as well, that allow the robots to communicate in an intermittent fashion and operate in disconnect mode

the rest of the time. This entry provides an overview of all-time and intermittent connectivity controllers.

Keywords

Algebraic graph theory · Graph connectivity · Network connectivity control · Distributed and hybrid control · Multi-agent systems

Introduction

Graph-Theoretic Connectivity Control

Recent methods for communication control of mobile robot networks typically rely on proximity graphs to model the exchange of information among the robots, and, therefore, the communication problem becomes equivalent to preserving graph connectivity. Methods to control graph connectivity typically rely on controlling the Fiedler value of the underlying graph. One possible way of doing so is maximizing the Fiedler value in a centralized (Kim and Mesbahi 2006) or distributed (DeGennaro and Jadbabaie 2006) fashion. Alternatively, potential fields that model loss of connectivity as an obstacle in the free space can be employed for this purpose, as shown in Zavlanos and Pappas (2007). A distributed hybrid approach to connectivity control is presented in Zavlanos and Pappas (2008), whereby communication links are efficiently manipulated by decoupling the continuous robot motion from the control of the discrete graph. Further distributed algorithms for graph connectivity maintenance have been implemented in Notarstefano et al. (June 2006), Ji and Egerstedt (2007), and Sabattini et al. (2013). A comprehensive survey of this literature can be found in Mesbahi and Egerstedt (2010) and Zavlanos et al. (2011).

Connectivity Control Using Realistic Communication Models

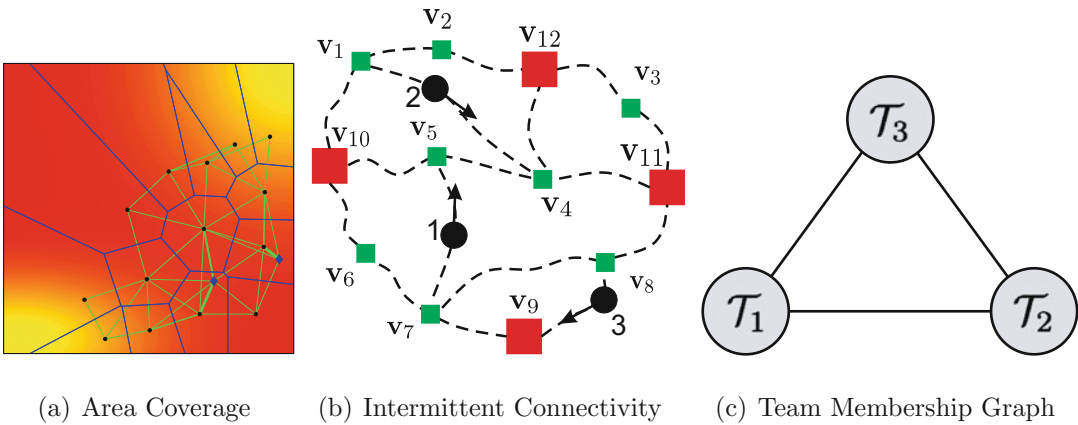
In practice, the above graph-based communication models turn out to be rather conservative, since proximity does not necessarily imply

tangible and reliable communication. More realistic communication models for mobile networks compared to the above graph-theoretic models are proposed in Zavlanos et al. (2010, 2013) and Stephan et al. (2017) that take into account the routing of packets as well as desired bounds on the transmitted rates. The key idea in these works is to define connectivity in terms of communication rates and to use optimization methods to determine optimal operating points of wireless networks. A conceptually similar communication model is proposed in Fink et al. (2010). Integration of these communication frameworks with area coverage control and navigation tasks in complex environments is presented in Kantaros and Zavlanos (2016b, a); see also Fig. 1a. A communication model that accounts for multipath fading of channels is proposed in Lindhé and Johansson (2010), where robot mobility is exploited in order to increase the throughput. Multipath fading, shadow fading, and path loss have also been used to model channels in Mostofi et al. (2010); Malmirchegini and Mostofi (2012). In these works, a probabilistic framework for channel prediction is developed based on a small number of measurements of the received signal power. The integration of the latter communica-

tion models with robot mobility is described in Ghaffarkhah and Mostofi (2010).

Intermittent Connectivity Control

Common in the above works is that point-to-point or end-to-end network connectivity is required to be preserved for all time. However, due to the uncertainty in the wireless channel, it is often impossible to ensure all-time connectivity in practice. Moreover, all-time connectivity constraints may prevent the robots from moving freely in their environment to fulfill their tasks and instead favor motions that maintain a reliable communication network. Motivated by this fact, intermittent communication frameworks have recently been proposed that allow the robots to communicate intermittently and operate in disconnect mode the rest of the time. Specifically, Hollinger and Singh (2010) and Hollinger et al. (2012) propose a receding horizon framework for periodic connectivity that ensures recovery of connectivity within a given time horizon. A similar recurrent connectivity framework for planning on environments modeled as graphs is proposed in Banfi et al. (2018). Unlike these approaches that require the whole network to become occasionally reconnected, alternative



Dynamic Graphs, Connectivity of, Fig. 1 (a) shows a communication network consisting of 14 robots (black dots) that are tasked with optimally sensing two sources, whose density is captured in yellow, while reliably transmitting the collected information to two access points (blue rhombuses). The thickness of communication links (green edges) depends on their reliability. (b) illustrates a joint task planning and intermittent connectivity frame-

work. The robots have to accomplish complex tasks, such as data gathering, by visiting the green waypoints, and exchange the collected information by communicating intermittently when they simultaneously visit the communication points (red squares). The teams are defined as $\mathcal{T}_1 = \{1, 2\}$, $\mathcal{T}_2 = \{2, 3\}$, and $\mathcal{T}_3 = \{1, 3\}$. The corresponding team membership graph is shown in (c)

approaches have been recently proposed that do not require the whole communication network to ever become connected at once, but they ensure connectivity over time, infinitely often (Zavlanos 2010). The key idea is to divide the robots into $M > 0$ smaller teams $\{\mathcal{T}_m\}_{m=1}^M$, where M is user-specified, and require that communication events take place when the robots in every team meet at a common location in space that can be fixed or optimally selected; see also Fig. 1b. While in disconnect mode, the robots can accomplish other tasks free of communication constraints. The teams $\mathcal{T}_m$ are constructed so that every robot belongs to at least one team giving rise to a team membership graph $\mathcal{G}_{\mathcal{T}} = (\mathcal{V}_{\mathcal{T}}, \mathcal{E}_{\mathcal{T}})$, where the set of nodes $\mathcal{V}_{\mathcal{T}}$ is indexed by the teams $\mathcal{T}_m$ and set of edges $\mathcal{E}_{\mathcal{T}}$ is defined as $\mathcal{E}_{\mathcal{T}} = \{(m, n) | \mathcal{T}_m \cap \mathcal{T}_n \neq \emptyset, \forall m, n \in \mathcal{V}_{\mathcal{T}}, m \neq n\}$. Assuming that the team membership graph $\mathcal{G}_{\mathcal{T}}$ is a connected bipartite graph, Zavlanos (2010) proposes a distributed control scheme that achieves periodic communication events, synchronously, at meeting/communication locations. Arbitrary connected team graphs are considered in Kantaros and Zavlanos (2017) and Kantaros and Zavlanos (2016c) where techniques from formal methods are employed to synthesize correct-by-construction discrete plans (schedules) that determine the order in which communication events should occur at the meeting locations. Specifically, in these works, intermittent connectivity is captured by the following global linear temporal logic (LTL) formula:

$$\phi = \wedge_{m \in \mathcal{M}} \left(\square \diamond \left(\bigvee_{j \in \mathcal{C}_m} (\wedge_{i \in \mathcal{T}_m} \pi_i^{\mathbf{v}_j}) \right) \right), \quad (1)$$

that requires all robots in a team $\mathcal{T}_m$ to meet infinitely often at a common communication point, denoted by $\mathbf{v}_j$, for all teams $\mathcal{T}_m$. In (1), $\pi_i^{\mathbf{v}_j}$ is a Boolean variable that is true if robot i is in location $\mathbf{v}_j$, and $\mathcal{C}_m$ is a set that collects all communication points $\mathbf{v}_j$ available to team $\mathcal{T}_m$. Also, $\square \diamond$ represents the infinitely often requirement. A detailed presentation of LTL semantics can be found in Baier and Katoen (2008). Then, Kantaros and Zavlanos

(2017, 2016c) propose a distributed control synthesis algorithm to synthesize schedules that satisfy ϕ . Integration of this intermittent connectivity framework with temporal logic task planning, state estimation tasks, and planning for time-critical and dynamic tasks is proposed in Kantaros et al. (2019), Khodayi-mehr et al. (2019), and Kantaros and Zavlanos (2018), respectively. An intermittent communication framework for starlike communication topologies is also presented in Guo and Zavlanos (2018) that considers information flow only to the root/user and not among the robots.

Summary and Future Research Directions

Network connectivity is an underlying assumption in every distributed control and optimization algorithm. As a result, all-time connectivity controllers that either rely on graph theory to model robot communication or employ more realistic communication models have been proposed to control communication between nodes. When all-time communication is impossible or very restrictive of a condition, intermittent connectivity controllers can be used that allow the robots to communicate in an intermittent fashion and operate in disconnect mode the rest of the time. Intermittent connectivity control in unknown and dynamic environments, such as crowded human environments or underwater environments, and resilient intermittent connectivity controllers that are, e.g., robust to robot failures are interesting future research directions.

Cross-References

- [Flocking in Networked Systems](#)
- [Graphs for Modeling Networked Interactions](#)

Bibliography

- Baier C, Katoen JP (2008) Principles of model checking, vol 26202649. MIT Press, Cambridge

- Banfi J, Basilico N, Amigoni F (2018) Multirobot reconnection on graphs: problem, complexity, and algorithms. *IEEE Trans Robot* 34(99):1–16
- DeGennaro MC, Jadbabaie A (2006) Decentralized control of connectivity for multi-agent systems. In: 45th IEEE conference on decision and control, San Diego, pp 3628–3633
- Fink J, Ribeiro A, Kumar V, Sadler BM (2010) Optimal robust multihop routing for wireless networks of mobile micro autonomous systems. In: IEEE military communications conference, 2010-MILCOM 2010, San Jose, pp 1268–1273
- Ghaffarkhah A, Mostofi Y (2010) Channel learning and communication-aware motion planning in mobile networks. In: IEEE American control conference, Baltimore, pp 5413–5420
- Guo M, Zavlanos MM (2018) Multirobot data gathering under buffer constraints and intermittent communication. *IEEE Trans Robot* 34(4):1082–1097
- Hollinger G, Singh S (2010) Multi-robot coordination with periodic connectivity. In: IEEE international conference on robotics and automation (ICRA), Anchorage, pp 4457–4462
- Hollinger G, Singh S et al (2012) Multirobot coordination with periodic connectivity: theory and experiments. *IEEE Trans Robot* 28(4):967–973
- Ji M, Egerstedt MB (2007) Distributed coordination control of multi-agent systems while preserving connectedness. *IEEE Trans Robot* 23(4):693–703
- Kantaros Y, Zavlanos MM (2016a) Distributed communication-aware coverage control by mobile sensor networks. *Automatica* 63:209–220
- Kantaros Y, Zavlanos MM (2016b) Global planning for multi-robot communication networks in complex environments. *IEEE Trans Robot* 32(5):1045–1061
- Kantaros Y, Zavlanos MM (2016c) Simultaneous intermittent communication control and path optimization in networks of mobile robots. In: Conference on decision and control (CDC). IEEE, Las Vegas, pp 1794–1795
- Kantaros Y, Zavlanos MM (2017) Distributed intermittent connectivity control of mobile robot networks. *Trans Autom Control* 62(7):3109–3121
- Kantaros Y, Zavlanos MM (2018) Distributed intermittent communication control of mobile robot networks under time-critical dynamic tasks. In: IEEE international conference on robotics and automation (ICRA), Brisbane, pp 5028–5033
- Kantaros Y, Guo M, Zavlanos MM (2019) Temporal logic task planning and intermittent connectivity control of mobile robot networks. *IEEE Trans Autom Control* 64:4105–4120
- Khodayi-Mehr R, Kantaros Y, Zavlanos MM (2019) Distributed state estimation using intermittently connected robot networks. *IEEE Trans Robot* 35:709–724
- Kim Y, Mesbahi M (2006) On maximizing the second smallest eigenvalue of a state-dependent graph laplacian. *IEEE Trans Autom Control* 51(1):116–120
- Lindhé M, Johansson KH (2010) Adaptive exploitation of multipath fading for mobile sensors. In: IEEE international conference on robotics and automation, Anchorage, pp 1934–1939
- Malmirchegini M, Mostofi Y (2012) On the spatial predictability of communication channels. *IEEE Trans Wirel Commun* 11(3):964–978
- Mesbahi M, Egerstedt M (2010) Graph theoretic methods in multiagent networks, vol 33. Princeton University Press, Princeton
- Mostofi Y, Malmirchegini M, Ghaffarkhah A (2010) Estimation of communication signal strength in robotic networks. In: IEEE international conference on robotics and automation, Anchorage, pp 1946–1951
- Notarstefano G, Savla K, Bullo F, Jadbabaie A (June 2006) Maintaining limited-range connectivity among second-order agents. In: IEEE American control conference, Minneapolis, p 6
- Sabattini L, Chopra N, Secchi C (2013) Decentralized connectivity maintenance for cooperative control of mobile robotic systems. *Int J Robot Res* 32(12):1411–1423
- Stephan J, Fink J, Kumar V, Ribeiro A (2017) Concurrent control of mobility and communication in multirobot systems. *IEEE Trans Robot* 33(5):1248–1254
- Zavlanos M, Pappas G (2008) Distributed connectivity control of mobile networks. *IEEE Trans Robot* 24(6):1416–1428
- Zavlanos M, Egerstedt M, Pappas G (2011) Graph theoretic connectivity control of mobile robot networks. *Proc IEEE* 99(9):1525–1540
- Zavlanos MM (2010) Synchronous rendezvous of very-low-range wireless agents. In: 49th IEEE conference on decision and control (CDC), Atlanta, pp 4740–4745
- Zavlanos MM, Pappas GJ (2007) Potential fields for maintaining connectivity of mobile networks. *IEEE Trans Robot* 23(4):812–816
- Zavlanos MM, Ribeiro A, Pappas GJ (2010) Mobility & routing control in networks of robots. In: 49th IEEE conference on decision and control, Atlanta, pp 7545–7550
- Zavlanos MM, Ribeiro A, Pappas GJ (2013) Network integrity in mobile robotic networks. *IEEE Trans Autom Control* 58(1):3–18

Dynamic Noncooperative Games

David A. Castañón
Boston University, Boston, MA, USA

Abstract

In this entry, we present models of dynamic noncooperative games, solution concepts and

algorithms for finding game solutions. For the sake of exposition, we focus mostly on finite games, where the number of actions available to each player is finite, and discuss briefly extensions to infinite games.

Keywords

Extensive form games · Finite games · Nash equilibrium

Introduction

Dynamic noncooperative games allow multiple actions by individual players, and include explicit representations of the information available to each player for selecting its decision. Such games have a complex temporal order of play and an information structure that reflects uncertainty as to what individual players know when they have to make decisions. This temporal order and information structure is not evident when the game is represented as a static game between players that select strategies. Dynamic games often incorporate explicit uncertainty in outcomes, by representing such outcomes as actions taken by a random player (called chance or “Nature”) with known probability distributions for selecting its actions.

We focus our exposition on models of finite games and discuss briefly extensions to infinite games at the end of the entry.

Finite Games in Extensive Form

The *extensive form* of a game was introduced by von Neumann (1928) and later refined by Kuhn (1953) to represent explicitly the order of play, the information and actions available to each player for making decisions at each of their turns, and the payoffs that players receive after a complete set of actions. Let $I = \{0, 1, \dots, n\}$ denote the set of players in a game, where player 0 corresponds to Nature. The extensive form is represented in terms of a game tree, consisting of a rooted tree with nodes $\mathcal{N}$ and edges $\mathcal{E}$. The

root node represents the initial state of the game. Nodes x in this tree correspond to positions or “states” of the game. Any non-root node with more than one incident edge is an internal node of the tree and is a *decision* node associated with a player in the game; the root node is also a decision node. Each decision node x has a player $o(x) \in I$ assigned to select a decision from a finite set of admissible decisions $A(x)$. Using distance from the root node to indicate direction of play, each edge that follows node x corresponds to an action in $A(x)$ taken by player $p(x)$, which evolves the state of the game into a subsequent node x' .

Non-root nodes with only one incident edge are terminal nodes, which indicate the end of the game; such nodes represent outcomes of the game. The unique path from the root node to a terminal node is called a *play* of the game. For each outcome node x , there are payoff functions $J_i(x)$ associated with each player $i \in \{1, \dots, n\}$.

The above form represents the different players, the order in which players select actions and their possible actions, and the resulting payoffs to each player from a complete set of plays of the game. The last component of interest is to represent the information available for players to select decisions at each decision node. Due to the tree structure of the extensive form of a game, each decision node x contains exact information on all of the previous actions taken that led to state x . In games of *perfect information*, each player knows exactly the decision node x at which he/she is selecting an action. To represent imperfect information, the extensive form uses the notion of an *information set*, which represents a group of decision nodes, associated with the same player, where the information available to that player is that the game is in one of the states in that information set. Formally, let $\mathcal{N}_i \subset \mathcal{N}$ be the set of decision nodes associated with player i , for $i \in I$. Let $\mathcal{H}_i$ denote a partition of $\mathcal{N}_i$ so that, for each set $h_i^k \in \mathcal{H}_i$, we have the following properties:

- If $x, x' \in h_i^k$, then they have the same admissible actions: $A(x) = A(x')$.

- If $x, x' \in h_i^k$, then they cannot both belong to a play of the game; that is, both x and x' cannot be on a path from the root node to an outcome.

Elements h_i^k for some player i are the *information sets*. Each decision node x belongs to one and only one information set associated with player $p(x)$. The constraints above ensure that, for each information set, there is a unique player identified to select actions, and the set of admissible actions is unambiguously defined. Denote by $A(h_i^k)$ the set of admissible actions at information set h_i^k . The last condition is a causality condition that ensures that a player who has selected a previous decision remembers that he/she has already made that previous decision.

Figure 1 illustrates an extensive form for a two-person game. Player 1 has two actions in any play of the game, whereas player 2 has only 1 action. Player 1 starts the game at the root node a ; the information set h_1^1 shows that player 2 is unaware of this action, as both nodes that descend from node a are in this information set. After player 2's action, player 1 gets to select a second action. However, the information sets h_1^2, h_1^3 indicate that player 1 recalls what earlier action

he/she selected, but he/she has not observed the action of player 2. The terminal nodes indicate the payoffs to player 1 and player 2 as an ordered pair.

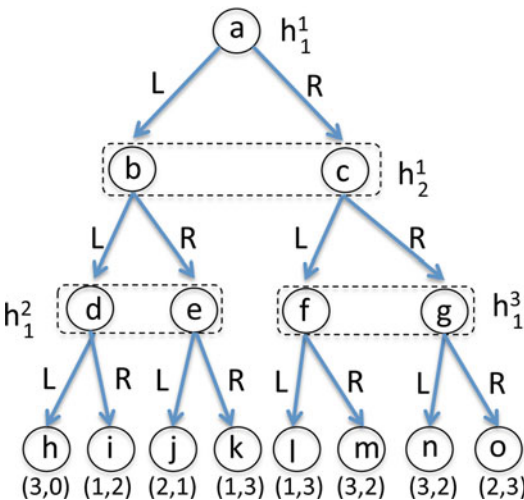
Strategies and Equilibrium Solutions

A pure strategy γ_i for player $i \in \{1, \dots, n\}$ is a function that maps each information set of player i into an admissible decision. That is,

$$\gamma_i : \mathcal{H}_i \rightarrow A$$

such that $\gamma_i(h_i^k) \in A(h_i^k)$. The set of pure strategies for player i is denoted as Γ_i .

Note that pure strategies do not include actions selected by Nature. Nature's actions are selected using probability distributions over the choices available for the information sets corresponding to Nature, where the choice at each information set is made independently of other choices. For finite games, the number of pure strategies for each player is also finite. Given a tuple of pure strategies $\underline{\gamma} = (\gamma_1, \dots, \gamma_n)$, we can define the probability of the play of the game resulting in outcome node x as $\pi(x)$, computed as follows: Each outcome node has a unique path (the play) p_{rx} from root node r . We initialize $\pi(r) = 1$ and node $n = r$. If the player at node n is Nature and the next node in p_{rx} is n' , then $\pi(n') = \pi(n) * p(n, n')$, where $p(n, n')$ is the probability that Nature chooses the action that leads to n' . Otherwise, let $i = p(n)$ be the player at node n and let $h_i^k(n)$ denote the information set containing node n . Then, if $\gamma^i(h_i^k(n)) = a(n, n')$, let $\pi(n') = \pi(n)$, where $a(n, n')$ is the action in $A(n)$ that leads to n' ; otherwise, set $\pi(n') = 0$. The above process is repeated letting $n' = n$, until n' equals the terminal node x . Using these probabilities, the resulting expected payoff to player i is



Dynamic Noncooperative Games, Fig. 1 Illustration of game in extensive form

$$J_i(\underline{\gamma}) = \sum_{x \text{ terminal}, x \in \mathcal{N}} \pi(x) J_i(x)$$

This representation of payoffs in terms of strategies transforms the game from an extensive form representation to a strategic or normal form representation, where the concept of dynamics and information has been abstracted away. The resulting strategic form looks like a static game as discussed in the encyclopedia entry ► [Strategic Form Games and Nash Equilibrium](#), where each player selects his/her strategy from a finite set, resulting in a vector of payoffs for the players. Using these payoffs, one can now define solution concepts for the game. Let the notation $\underline{\gamma}_{-i} = (\gamma_1, \dots, \gamma_{i-1}, \gamma_{i+1}, \gamma_n)$ denote the set of strategies in a tuple excluding the i th strategy. A Nash equilibrium solution is a tuple of feasible strategies $\underline{\gamma}^* = (\gamma_1^*, \dots, \gamma_n^*)$ such that

$$\begin{aligned} J_i(\underline{\gamma}^*) &\geq J_i(\underline{\gamma}_{-i}^*, \gamma_i) \text{ for all } \gamma_i \in \Gamma_i, \\ \text{for all } i &\in \{1, \dots, n\} \end{aligned} \quad (1)$$

The special case of two-person games where $J_1(\underline{\gamma}) = -J_2(\underline{\gamma})$ are known as zero-sum games.

As discussed in the encyclopedia entry on static games (► [Strategic Form Games and Nash Equilibrium](#)), the existence of Nash equilibria or even saddle point strategies in terms of pure strategies is not guaranteed for finite games. Thus, one must consider the use of *mixed strategies*. A mixed strategy μ_i for player $i \in \{1, \dots, n\}$ is a probability distribution over the set of pure strategies Γ_i . The definition of payoffs can be extended to mixed strategies by averaging the payoffs associated with the pure strategies, as

$$J_i(\underline{\mu}_i) = \sum_{\gamma_1 \in \Gamma_1} \dots \sum_{\gamma_n \in \Gamma_n} \mu_1(\gamma_1) \dots \mu_n(\gamma_n) J_i(\gamma_1, \dots, \gamma_n)$$

Denote the set of probability distributions over Γ_i as $\Delta(\Gamma_i)$. An n -tuple $\underline{\mu}^* = (\mu_1, \dots, \mu_n)$ of mixed strategies is said to be a Nash equilibrium if

$$\begin{aligned} J_i(\underline{\mu}^*) &\geq J_i(\underline{\mu}_{-i}^*, \mu_i) \text{ for all } \mu_i \\ &\in \Delta(\Gamma_i), \text{ for all } i \in \{1, \dots, n\} \end{aligned}$$

Theorem 1 (Nash 1950, 1951) *Every finite n -person game has at least one Nash equilibrium point in mixed strategies.*

Mixed strategies suggest that each player's randomization occurs before the game is played, by choosing a strategy at random from its choices of pure strategies. For games in extensive form, one can introduce a different class of strategies where a player makes a random choice of action at each information set, according to a probability distribution that depends on the specific information set. The choice of action is selected independently at each information set according to a selected probability distribution, in a manner similar to how Nature's actions are selected. These random choice strategies are known as *behavior strategies*. Let $\Delta(A(h_i^k))$ denote the set of probability distributions over the decisions $A(h_i^k)$ for player i at information set h_i^k . A behavior strategy for player i is an element $x_i \in \prod_{h_i^k \in \mathcal{H}_i} \Delta(A(h_i^k))$, where $x_i(h_i^k)$ denotes the probability distribution of the behavior strategy x_i over the available decisions at information set h_i^k .

Note that the space of admissible behavior strategies is much smaller than the space of admissible mixed strategies. To illustrate this, consider a player with K information sets and two possible decisions for each information set. The number of possible pure strategies would be 2^K , and thus the space of mixed strategies would be a probability simplex of dimension $2^K - 1$. In contrast, behavior strategies would require specifying probability distributions over two choices for each of K information sets, so the space of behavior strategies would be a product space of K probability simplices of dimension 1, totaling dimension K . One way of understanding this difference is that mixed strategies introduce correlated randomization across choices at different information sets, whereas behavior strategies introduce independent randomization across such choices.

For every behavior strategy, one can find an equivalent mixed strategy by computing the probabilities of every set of actions that result from the behavior strategy. The converse is not true for general games in extensive form. However,

there is a special class of games for which the converse is true. In this class of games, players recall what actions they have selected previously and what information they knew previously. A formal definition of perfect recall is beyond the scope of this exposition but can be found in Hart (1991) and Kuhn (1953). The implication of perfect recall is summarized below:

Theorem 2 (Kuhn 1953) *Given a finite n -person game in which player i has perfect recall, for each mixed strategy μ_i for player i , there exists a corresponding behavior strategy x_i that is equivalent, where every player receives the same payoffs under both strategies μ_i and x_i .*

This equivalence was extended by Aumann to infinite games (Aumann 1964). In dynamic games, it is common to assume that each player has perfect recall and thus solutions can be found in the smaller space of behavior strategies.

Computation of Equilibria

Algorithms for the computation of mixed-strategy Nash equilibria of static games can be extended to compute mixed-strategy Nash equilibria for games in extensive form when the pure strategies are enumerated as above. However, the number of pure strategies grows exponentially with the size of the extensive form tree, making these methods hard to apply. For two-person games in extensive form with perfect recall by both players, one can search for Nash equilibria in the much smaller space of behavior strategies. This was exploited in Koller et al. (1996) to obtain efficient linear complementarity problems for nonzero-sum games and linear programs for zero-sum games where the number of variables involved is linear in the number of internal decision nodes of the extensive form of the game. A more detailed overview of computation algorithms for Nash equilibria can be found in McKelvey and McLennan (1996). The Gambit Web site (McKelvey et al. 2010) provides software implementations of several techniques for computation of Nash equilibria in two- and n -person games.

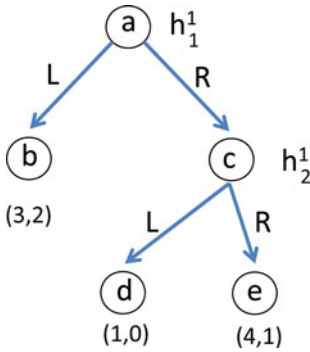
An alternative approach to computing Nash equilibria for games in extensive forms is based on subgame decomposition, discussed next.

Subgames

Consider a game G in extensive form. A node c is a successor of a node n if there is a path in the game tree from n to c . Let h be a node in G that is not terminal and is the only node in its information set. Assume that if a node c is a successor of h , then every node in the information set containing c is also a successor of h . In this situation, one can define a subgame H of G with root node h , which consists of node h and its successors, connected by the same actions as in the original game G , with the payoffs and terminal vertices equal to those in G . This is the subgame that would be encountered by the players had the previous play of the game reached the state at node h . Since the information set containing node h contains no other node and all the information sets in the subgame contain only nodes in the subgame, then every player in the subgame knows that they are playing only in the subgame once h has been reached.

Figure 2 illustrates a game where every non-terminal node is contained in its own information set. This game contains a subgame rooted at node c . Note that the full game has two Nash equilibria in pure strategies: strategies (L, L) and (R, R) . However, strategy (L, L) is inconsistent with how player 2 would choose its decision if it were to find itself at node c . This inconsistency arises because the Nash equilibria are defined in terms of strategies announced before the game is played and may not be a reasonable way to select decisions if unanticipated plays occur. When player 1 chooses L , node c should not be reached in the play of the game, and thus player 2 can choose L because it does not affect the expected payoff. One can define the concept of Nash equilibria that are sequentially consistent as follows.

Let x_i be behavior strategies for player i in game G . Denote the restriction of these strategies to the subgame H as x_i^H . This restriction describes the probabilities for choices of actions for the information sets in H . Suppose the game



Dynamic Noncooperative Games, Fig. 2 Simple game with multiple Nash equilibria

G has a Nash equilibrium achieved by strategies $(x_1, \dots, x_n)$. This Nash equilibrium is called *subgame perfect* (Selten 1975) if, for every subgame H of G , the strategies $(x_1^H, \dots, x_n^H)$ are a Nash equilibrium for the subgame H . In the game in Fig. 2, there is only one subgame perfect equilibrium, which is (R, R) . There are several other refinements of the Nash equilibrium concept to enforce sequential consistency, such as sequential equilibria (Kreps and Wilson 1982).

An important application of subgames is to compute subgame perfect equilibria by backward induction. The idea is to start with a subgame root node as close as possible to an outcome node (e.g., node c in Fig. 2). This small subgame can be solved for its Nash equilibria, to compute the equilibrium payoffs for each player. Then, in the original game G , the root node of the subgame can be replaced by an outcome node, with payoffs equal to the equilibrium payoffs in the subgame. This results in a smaller game, and the process can be applied inductively until the full game is solved. The subgame perfect equilibrium strategies can then be computed as the solution of the different subgames solved in this backward induction process. For Fig. 2, the subgame at c is solved by player 2 selecting R , with payoffs $(4,1)$. The reduced new game has two choices for player 1, with best decision R . This leads to the overall subgame perfect equilibrium (R, R) .

This backward induction procedure is similar to the dynamic programming approach to solving control problems. Backward induction was first

used by Zermelo (1912) to analyze zero-sum games of perfect information such as chess. An extension of Zermelo's work by Kuhn (1953) establishes the following result:

Theorem 3 *Every finite game of perfect information has a subgame perfect Nash equilibrium in pure strategies.*

This result follows because, at each step in the backward induction process, the resulting subgame consists of a choice among finite actions for a single player, and thus a pure strategy achieves the maximal payoff possible.

Infinite Noncooperative Games

When the number of options available to players is infinite, game trees are no longer appropriate representations for the evolution of the game. Instead, one uses state space models with explicit models of actions and observations. A typical multistage model for the dynamics of such games is

$$\begin{aligned} x(t+1) &= f(x(t), u_1(t), \dots, u_n(t), w(t), t), \\ t &= 0, \dots, T-1 \end{aligned} \quad (2)$$

with initial condition $\underline{x}(0) = w(0)$, where $x(t)$ is the state of the game at stage t , and $u_1(t), \dots, u_n(t)$ are actions selected by players $1, \dots, n$ at stage t , and $w(t)$ is an action selected by Nature at stage t . The space of actions for each player are restricted at each time to infinite sets $A_i(t)$ with an appropriate topological structure, and the admissible state $x(t)$ at each time belongs to an infinite set X with a topological structure. In terms of Nature's actions, for each stage t , there is a probability distribution that specifies the choice of Nature's action $w(t)$, selected independently of other actions.

Equation (2) describes a play of the game, in terms of how different actions by players at the various stages evolve the state of the game. A play of the game is thus a history $\underline{h} = (x(0), u_1(0), \dots, u_n(0), x(1), u_1(1), \dots, u_n(1), \dots, x(T))$. Associated with each play of the game is a set of real-valued functions $J_i(\underline{h})$

that indicates the payoff to player i in this play. This function is often assumed to be separable across the variables in each stage.

To complete the description of the extensive form, one must now introduce the available information to each player at each stage. Define observation functions

$$y_i(t) = g_i(x(t), v(t), t), \quad i = 1, \dots, n$$

where $y_i(t)$ takes values in observations spaces which may be finite or infinite, and $v(t)$ are selected by Nature given their probability distributions, independent of other selections. Define the information available for player i at stage t to be $I_i(t)$, a subset of $\{y_1(0), \dots, y_n(0), \dots, y_1(t), \dots, y_n(t); u_1(0), \dots, u_n(0), \dots, u_1(t-1), \dots, u_n(t-1)\}$. With this notation, strategies $\gamma_i(t)$ for player i are functions that map, at each stage, the available information $I_i(t)$ into admissible decisions $u_i(t) \in A_i(t)$. With appropriate measurability conditions, specifying a full set of strategies $\underline{\gamma} = (\gamma_1, \dots, \gamma_n)$ for each of the players induces a probability distribution on the plays of the game, which leads to the expected payoff $J_i(\underline{\gamma})$. Nash equilibria are defined in identical fashion to (1).

Obtaining solutions of multistage games is a difficult task that depends on the ability to use the subgame decomposition techniques discussed previously. Such subgame decompositions are possible when games do not include actions by Nature and the payoff functions have a stagewise additive property. Under such cases, backward induction allows the construction of Nash equilibrium strategies through the recursive solutions of static infinite games, such as those discussed in the encyclopedia entry on static games.

Generalizations of multistage games to continuous time result in differential games, covered in two articles in the encyclopedia, but for the zero-sum case. Additional details on infinite dynamic noncooperative games, exploiting different models and information structures and studying the existence, uniqueness, or nonuniqueness of equilibria, can be found in Basar and Olsder (1982).

Conclusions

In this entry, we reviewed models for dynamic noncooperative games that incorporate temporal order of play and uncertainty as to what individual players know when they have to make decisions. Using these models, we defined solution concepts for the games and discussed algorithms for determining solution strategies for the players. Active directions of research include development of new solution concepts for dynamic games, new approaches to computation of game solutions, the study of games with a large number of players, evolutionary games where players' greedy behavior evolves toward equilibrium strategies, and special classes of dynamic games such as Markov games and differential games. Several of these topics are discussed in other entries in the encyclopedia.

Cross-References

- [Stochastic Dynamic Programming](#)
- [Stochastic Games and Learning](#)
- [Strategic Form Games and Nash Equilibrium](#)

Bibliography

- Aumann RJ (1964) Mixed and behavior strategies in infinite extensive games. In: Dresher M, Shapley LS, Tucker AW (eds) *Advances in game theory*. Princeton University Press, Princeton
- Basar T, Olsder GJ (1982) *Dynamic noncooperative game theory*. Academic press, London
- Hart S (1991) Games in extensive and strategic forms. In: Aumann R, Hart S (eds) *Handbook of game theory with economic applications*, vol 1. Elsevier, Amsterdam
- Koller D, Megiddo N, von Stengel B (1996) Efficient computation of equilibria for extensive two-person games. *Games Econ Behav* 14:247–259
- Kreps DM, Wilson RB (1982) Sequential equilibria. *Econometrica* 50:863–894
- Kuhn HW (1953) Extensive games and the problem of information. In: Kuhn HW, Tucker AW (eds) *Contributions to the theory of games*, vol 2. Princeton University Press, Princeton
- McKelvey RD, McLennan AM (1996) Computation of equilibria in finite games. In: Amman HM, Kendrick DA, Rust J (eds) *Handbook of computational economics*, vol 1. Elsevier, Amsterdam

- McKelvey RD, McLennan AM, Turocy TL (2010) Gambit: software tools for game theory, version 0.2010.09.01. <http://www.gambit-project.org>
- Nash J (1950) Equilibrium points in n-person games. *Proc Natl Acad Sci* 36(1):48–49
- Nash J (1951) Non-cooperative games. *Ann Math* 54(2):286–295
- Selten R (1975) Reexamination of the perfectness concept for equilibrium points in extensive games. *Int J Game Theory* 4(1):25–55
- von Neumann J (1928) Zur theorie der gesellschaftsspiele. *Mathematische Annale* 100:295–320
- Zermelo E (1912) Über eine anwendung der mengenlehre auf die theorie des schachspiels. In: *Proceedings of the fifth international congress of mathematicians*, Cambridge University Press, Cambridge, vol 2

Dynamic Positioning Control Systems for Ships and Underwater Vehicles

Asgeir J. Sørensen

Department of Marine Technology, Centre for Autonomous Marine Operations and Systems (AMOS), Norwegian University of Science and Technology, NTNU, Trondheim, Norway

Abstract

In 2012 the fleet of dynamically positioned (DP) ships and rigs was probably larger than 3,000 units, predominately operating in the offshore oil and gas industry. The complexity and functionality vary subject to the targeted marine operation, vessel concept, and risk level. DP systems with advanced control functions and redundant sensor, power, and thruster/propulsion configurations are designed in order to provide high-precision fault-tolerant control in safety-critical marine operations. The DP system is customized for the particular application with integration to other control systems, e.g., power management, propulsion, drilling, oil and gas production, off-loading, crane operation, and pipe and cable laying. For underwater vehicles such as remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs),

DP functionality also denoted as *hovering* is implemented on several vehicles.

Keywords

Autonomous underwater vehicles (AUVs) · Fault-tolerant control · Remotely operated vehicles (ROVs)

Introduction

The offshore oil and gas industry is the dominating market for DP vessels. The various offshore applications include offshore service vessels, drilling rigs (semisubmersibles) and ships, shuttle tankers, cable and pipe layers, floating production, storage, and off-loading units (FPSOs), crane and heavy lift vessels, geological survey vessels, rescue vessels, and multipurpose construction vessels. DP systems are also installed on cruise ships, yachts, fishing boats, navy ships, tankers, and others.

A DP vessel is by the International Maritime Organization (IMO) and the maritime class societies (DNV GL, ABS, LR, etc.) defined as *a vessel that maintains its position and heading (fixed location denoted as stationkeeping or pre-determined track) exclusively by means of active thrusters*. The DP system as defined by class societies is not only limited to the DP control system including computers and cabling. Position reference systems of various types measuring North-East coordinates (satellites, hydroacoustic, optics, taut wire, etc.), sensors (heading, roll, pitch, wind speed and direction, etc.), the power system, thruster and propulsion system, and independent joystick system are also essential parts of the DP system. In addition, the DP operator is an important element securing safe and efficient DP operations. Further development of human-machine interfaces, alarm systems, and operator decision support systems is regarded as top priority bridging advanced and complex technology to safe and efficient marine operations. Sufficient DP operator training is a part of this.

The thruster and propulsion system controlled by the DP control system is regarded as one of

the main power consumers on the DP vessel. An important control system for successful integration with the power plant and the other power consumers such as drilling system, process system, heating, and ventilation system is the power and energy management system (PMS/EMS) balancing safety requirements and energy efficiency. The PMS/EMS controls the power generation and distribution and the load control of heavy power consumers. In this context both transient and steady-state behaviors are of relevance. A thorough understanding of the hydrodynamics, dynamics between coupled systems, load characteristics of the various power consumers, control system architecture, control layers, power system, propulsion system, and sensors is important for successful design and operation of DP systems and DP vessels.

Thruster-assisted position mooring is another important stationkeeping application often used for FPSOs, drilling rigs, and shuttle tanker operations where the DP system has to be redesigned accounting for the effect of the mooring system dynamics. In thruster-assisted position mooring, the DP system is renamed to position mooring (PM) system. PM systems have been commercially available since the 1980s. While for DP-operated ships the thrusters are the sole source of the stationkeeping, the assistance of thrusters is only complementary to the mooring system. Here, most of the stationkeeping is provided by a deployed anchor system. In severe environmental conditions, the thrust assistance is used to minimize the vessel excursions and line tension by mainly increasing the damping in terms of velocity feedback control and adding a bias force minimizing the mean tensions of the most loaded mooring lines. Modeling and control of turret-anchored ships are treated in Strand et al. (1998) and Nguyen and Sørensen (2009).

Overview of DP systems including references can be found in Fay (1989), Fossen (2011), and Sørensen (2011). The scientific and industrial contributions since the 1960s are vast, and many research groups worldwide have provided important results. In Sørensen et al. (2012), the development of DP system for ROVs is presented.

Mathematical Modeling of DP Vessels

Depending on the operational conditions, the vessel models may briefly be classified into stationkeeping, low-velocity, and high-velocity models. As shown in Sørensen (2011) and Fossen (2011) and the references therein, different model reduction techniques are used for the various speed regimes. Vessel motions in waves are defined as seakeeping and will here apply both for stationkeeping (zero speed) and forward speed. DP vessels or PM vessels can in general be regarded as stationkeeping and low-velocity or low Froude number applications. This assumption will particularly be used in the formulation of mathematical models used in conjunction with the controller design. It is common to use a two-time scale formulation by separating the total model into a low-frequency (LF) model and a wave-frequency (WF) model (seakeeping) by superposition. Hence, the total motion is a sum of the corresponding LF and the WF components. The WF motions are assumed to be caused by first-order wave loads. Assuming small amplitudes, these motions will be well represented by a linear model. The LF motions are assumed to be caused by second-order mean and slowly varying wave loads, current loads, wind loads, mooring (if any), and thrust and rudder forces and moments.

For underwater vehicles operating below the wave zone, estimated to be deeper than half the wavelength, the wave loads can be disregarded and of course the effect of the wind loads as well.

Modeling Issues

The mathematical models may be formulated in two complexity levels:

- *Control plant model* is a simplified mathematical description containing only the main physical properties of the process or plant. This model may constitute a part of the controller. The control plant model is also used in analytical stability analysis based on, e.g., Lyapunov stability.

- *Process plant model* or simulation model is a comprehensive description of the actual process and should be as detailed as needed. The main purpose of this model is to simulate the real plant dynamics. The process plant model is used in numerical performance and robustness analysis and testing of the control systems. As shown above, the process plant models may be implemented for off-line or real-time simulation (e.g., HIL testing; see Johansen et al. 2007) purposes defining different requirements for model fidelity.

Kinematics

The relationship between the Earth-fixed position and orientation of a floating structure and its body-fixed velocities is

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1(\boldsymbol{\eta}_2) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{J}_2(\boldsymbol{\eta}_2) \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} \quad (1)$$

The vectors defining the Earth-fixed vessel position ($\boldsymbol{\eta}_1$) and orientation ($\boldsymbol{\eta}_2$) using Euler angles and the body-fixed translation ($\mathbf{v}_1$) and rotation ($\mathbf{v}_2$) velocities are given by

$$\boldsymbol{\eta}_1 = [x \ y \ z]^T, \boldsymbol{\eta}_2 = [\phi \ \theta \ \psi]^T, \\ \mathbf{v}_1 = [u \ v \ w]^T, \mathbf{v}_2 = [p \ q \ r]^T. \quad (2)$$

The rotation matrix $\mathbf{J}_1(\boldsymbol{\eta}_2) \in \mathbf{SO}(3)$ and the velocity transformation matrix $\mathbf{J}_2(\boldsymbol{\eta}_2) \in \mathbb{R}^{3 \times 3}$ are defined in Fossen (2011). For ships, if only surge, sway and yaw (3DOF) are considered, the kinematics and the state vectors are reduced to

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\mathbf{v}, \text{ or } \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} \\ = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix}. \quad (3)$$

Process Plant Model: Low-Frequency Motion

The 6-DOF LF model formulation is based on Fossen (2011) and Sørensen (2011). The

equations of motion for the nonlinear LF model of a floating vessel are given by

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v})\mathbf{v} + \mathbf{C}_A(\mathbf{v}_r)\mathbf{v}_r + \mathbf{D}(\mathbf{v}_r) + \mathbf{G}(\boldsymbol{\eta}) \\ = \boldsymbol{\tau}_{\text{wave2}} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{thr}} + \boldsymbol{\tau}_{\text{moor}}, \quad (4)$$

where $\mathbf{M} \in \mathbb{R}^{6 \times 6}$ is the system inertia matrix including added mass; $\mathbf{C}_{RB}(\mathbf{v}) \in \mathbb{R}^{6 \times 6}$ and $\mathbf{C}_A(\mathbf{v}_r) \in \mathbb{R}^{6 \times 6}$ are the skew-symmetric Coriolis and centripetal matrices of the rigid body and the added mass; $\mathbf{G}(\boldsymbol{\eta}) \in \mathbb{R}^6$ is the generalized restoring vector caused by the mooring lines (if any), buoyancy, and gravitation; $\boldsymbol{\tau}_{\text{thr}} \in \mathbb{R}^6$ is the control vector consisting of forces and moments produced by the thruster system; $\boldsymbol{\tau}_{\text{wind}}$ and $\boldsymbol{\tau}_{\text{wave2}} \in \mathbb{R}^6$ are the wind and second-order wave load vectors, respectively.

The damping vector may be divided into linear and nonlinear terms according to

$$\mathbf{D}(\mathbf{v}_r) = \mathbf{d}_L(\mathbf{v}_r, \kappa)\mathbf{v}_r + \mathbf{d}_{NL}(\mathbf{v}_r, \gamma_r)\mathbf{v}_r, \quad (5)$$

where $\mathbf{v}_r \in \mathbb{R}^6$ is the relative velocity vector between the current and the vessel. The nonlinear damping, $\mathbf{d}_{NL}$, is assumed to be caused by turbulent skin friction and viscous eddy-making, also denoted as vortex shedding (Faltinsen 1990). The strictly positive linear damping matrix $\mathbf{d}_L \in \mathbb{R}^{6 \times 6}$ is caused by linear laminar skin friction and is assumed to vanish for increasing speed according to

$$\mathbf{d}_L(\mathbf{v}_r, \kappa) = \begin{bmatrix} X_{u_r} e^{-\kappa|u_r|} & .. & X_r e^{-\kappa|r|} \\ .. & .. & .. \\ N_{u_r} e^{-\kappa|u_r|} & .. & N_r e^{-\kappa|r|} \end{bmatrix}, \quad (6)$$

where κ is a positive scaling constant such that $\kappa \in \mathbb{R}^+$.

Process Plant Model: Wave-Frequency Motion

The coupled equations of the WF motions in surge, sway, heave, roll, pitch, and yaw are assumed to be linear and can be formulated as

$$\begin{aligned} \mathbf{M}(\omega)\dot{\boldsymbol{\eta}}_{Rw} + \mathbf{D}_p(\omega)\dot{\boldsymbol{\eta}}_{Rw} + \mathbf{G}\boldsymbol{\eta}_{Rw} &= \boldsymbol{\tau}_{\text{wave1}}, & \dot{\omega}_p &= 0, & (12) \\ \dot{\boldsymbol{\eta}}_w &= \mathbf{J}(\bar{\boldsymbol{\eta}}_2)\dot{\boldsymbol{\eta}}_{Rw}, & \mathbf{y} &= \left[(\boldsymbol{\eta} + \mathbf{C}_{pw}\mathbf{p}_w)^T \quad \omega_p \right]^T, & (13) \end{aligned} \quad (7)$$

where $\boldsymbol{\eta}_{Rw} \in \mathbb{R}^6$ is the WF motion vector in the hydrodynamics frame. $\boldsymbol{\eta}_w \in \mathbb{R}^6$ is the WF motion vector in the Earth-fixed frame. $\boldsymbol{\tau}_{\text{wave1}} \in \mathbb{R}^6$ is the first-order wave excitation vector, which will be modified for varying vessel headings relative to the incident wave direction. $\mathbf{M}(\omega) \in \mathbb{R}^{6 \times 6}$ is the system inertia matrix containing frequency dependent added mass coefficients in addition to the vessel's mass and moment of inertia. $\mathbf{D}_p(\omega) \in \mathbb{R}^{6 \times 6}$ is the wave radiation (potential) damping matrix. The linearized restoring coefficient matrix $\mathbf{G} \in \mathbb{R}^{6 \times 6}$ is due to gravity and buoyancy affecting heave, roll, and pitch only. For anchored vessels, it is assumed that the mooring system will not influence the WF motions.

Remark 1 Generally, a time domain equation cannot be expressed with frequency domain coefficient $-\omega$. However, this is a common used formulation denoted as a pseudo-differential equation. An important feature of the added mass terms and the wave radiation damping terms is the memory effects, which in particular are important to consider for nonstationary cases, e.g., rapid changes of heading angle. Memory effects can be taken into account by introducing a convolution integral or a so-called retardation function (Newman 1977) or state space models as suggested by Fossen (2011).

Control Plant Model

For the purpose of controller design and analysis, it is convenient to apply model reduction and derive a LF and WF control plant model in surge, sway, and yaw about zero vessel velocity according to

$$\dot{\mathbf{p}}_w = \mathbf{A}_{pw}\mathbf{p}_w + \mathbf{E}_{pw}\mathbf{w}_{pw}, \quad (8)$$

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\mathbf{v}, \quad (9)$$

$$\dot{\mathbf{b}} = -\mathbf{T}_b\mathbf{b} + \mathbf{E}_b\mathbf{w}_b, \quad (10)$$

$$\mathbf{M}\dot{\mathbf{v}} = -\mathbf{D}_L\mathbf{v} + \mathbf{R}^T(\psi)\mathbf{b} + \boldsymbol{\tau}, \quad (11)$$

where $\omega_p \in \mathbb{R}$ is the peak frequency of the waves (PFW). The estimated PFW can be calculated by spectral analysis of the pitch and roll measurements assumed to dominantly oscillate at the peak wave frequency. In the spectral analysis, the discrete Fourier transforms of the measured roll and pitch, which are collected through a period of time, are done by taking the n-point fast Fourier transform (FFT). The PFW may be found to be the frequency at which the power spectrum is maximal. The assumption $\dot{\omega}_p = 0$ is valid for slowly varying sea state. You can also find the wave frequency using nonlinear observers/EKF and signal processing techniques. It is assumed that the second-order linear model is sufficient to describe the first-order wave-induced motions, and then $\mathbf{p}_w \in \mathbb{R}^6$ is the state of the WF model. $\mathbf{A}_{pw} \in \mathbb{R}^{6 \times 6}$ is assumed Hurwitz and describes the first-order wave-induced motion as a mass-damper-spring system. $\mathbf{w}_{pw} \in \mathbb{R}^3$ is a zero-mean Gaussian white noise vector. $\mathbf{y}$ is the measurement vector. The WF measurement matrix $\mathbf{C}_{pw} \in \mathbb{R}^{3 \times 6}$ and the disturbance matrix $\mathbf{E}_{pw} \in \mathbb{R}^{6 \times 3}$ are formulated as

$$\mathbf{C}_{pw} = [\mathbf{0}_{3 \times 3} \quad \mathbf{I}_{3 \times 3}], \mathbf{E}_{pw}^T = [\mathbf{0}_{3 \times 3} \quad \mathbf{K}_w^T]^T. \quad (14)$$

Here, a 3-DOF model is assumed adopting the notation in (3) such that $\boldsymbol{\eta} \in \mathbb{R}^3$ and $\mathbf{v} \in \mathbb{R}^3$ are the LF position vector in the Earth-fixed frame and the LF velocity vector in the body-fixed frame, respectively. $\mathbf{M} \in \mathbb{R}^{3 \times 3}$ and $\mathbf{D}_L \in \mathbb{R}^{3 \times 3}$ are the mass matrix including hydrodynamic added mass and linear damping matrix, respectively. The bias term accounting for unmodeled affects and slowly varying disturbances $\mathbf{b} \in \mathbb{R}^3$ is modeled as Markov processes with positive definite diagonal matrix $\mathbf{T}_b \in \mathbb{R}^{3 \times 3}$ of time constants. If $\mathbf{T}_b$ is removed, a Wiener process is used. $\mathbf{w}_b \in \mathbb{R}^3$ is a bounded disturbance vector, and $\mathbf{E}_b \in \mathbb{R}^{3 \times 3}$ is a disturbance scaling matrix. $\boldsymbol{\tau} \in \mathbb{R}^3$

is the control force. As mentioned later in the paper, the proposed model reduction considering only horizontal motions may create problems conducting DP operations of structures with low waterplane area such as semisubmersibles. More details can be found in Sørensen (2011) and Fossen (2011) and the references therein.

For underwater vehicles, 6-DOF model should be used. For underwater vehicles with self-stabilizing roll and pitch, a 4-DOF model with surge, sway, yaw, and heave may be used; see Sørensen et al. (2012).

Control Levels and Integration Aspects

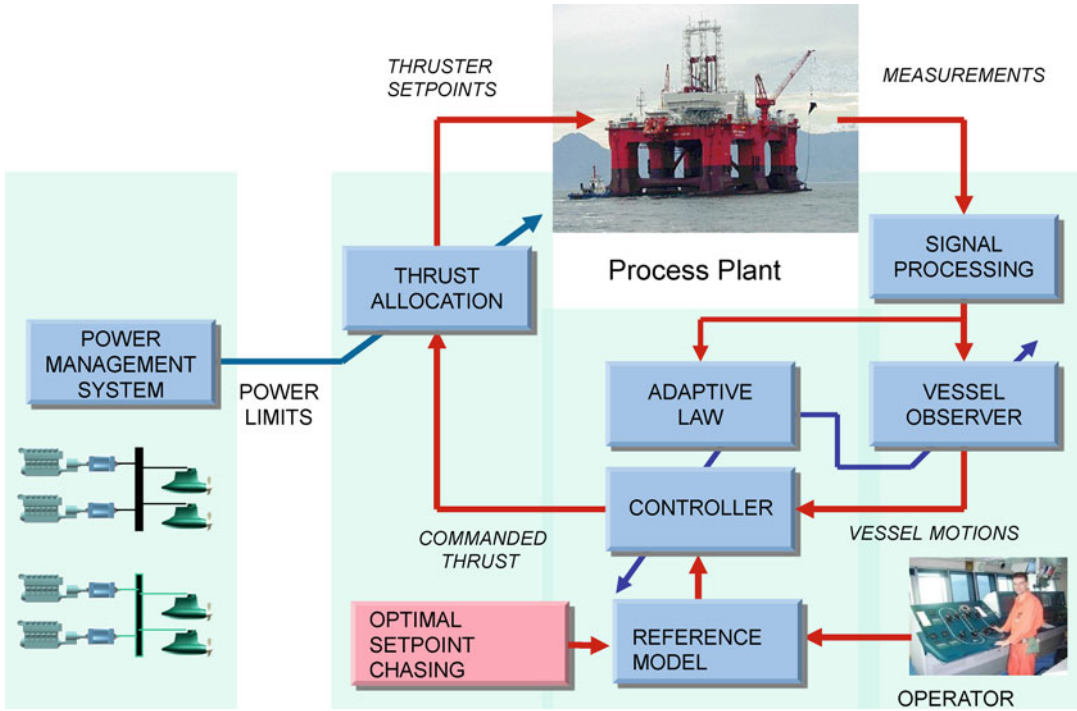
The real-time control hierarchy of a marine control system (Sørensen 2005) may be divided into three levels: *the guidance system and local optimization*, *the high-level plant control* (e.g., DP controller including thrust allocation), and *the low-level thruster control*. The DP control system consists of several modules as indicated in Fig. 1:

- *Signal processing* for analysis and testing of the individual signals including voting and weighting when redundant measurements are available. Ensuring robust and fault-tolerant control proper diagnostics and change detection algorithms is regarded as maybe one of the most important research areas. For an overview of the field, see Basseville and Nikiforov (1993) and Blanke et al. (2003).
- *Vessel observer* for state estimation and wave filtering. In case of lost sensor signals, the predictor is used to provide dead reckoning, which is required by class societies. Prediction error which is the deviation between the measurements and the estimated measurements is also one important barrier in the failure detection.
- *Feedback control law* is often of multivariable PID type, where feedback is produced from the estimated low-frequency (LF) position and heading deviations and estimated LF velocities.

- *Feedforward control law* is normally the wind force and moment. For the different applications (pipe laying, ice operations, position mooring), tailor-made feedforward control functions are also used.
- *Guidance system with reference models* is needed in achieving a smooth transition between setpoints. In the most basic case, the operator specifies a new desired position and heading, and a reference model generates smooth reference trajectories/paths for the vessel to follow. A more advanced guidance system involves way-point tracking functionality with optimal path planning.
- *Thrust allocation* computes the force and direction commands to each thruster device based on input from the resulting feedback and feedforward controllers. The low-level thruster controllers will then control the propeller pitch, speed, torque, and power.
- *Model adaptation* provides the necessary corrections of the vessel model and the controller settings subject to changes in the vessel draft, wind area, and variations in the sea state.
- *Power management system* performs diesel engine control, power generation management with frequency and voltage monitoring, active and passive load sharing, and load dependent start and stop of generator sets.

DP Controller

In the 1960s the first DP system was introduced for horizontal modes of motion (surge, sway, and yaw) using single-input single-output PID control algorithms in combination with low-pass and/or notch filters. In the 1970s more advanced output control methods based on multivariable optimal control and Kalman filter theory were proposed by Balchen et al. (1976) and later refined in Sælid et al. (1983); Grimble and Johnson (1988); and others as referred to in Sørensen (2011). In the 1990s nonlinear DP controller designs were proposed by several research groups; for an overview see Strand et al. (1998), Fossen and Strand (1999), Pettersen and Fossen (2000), and Sørensen (2011). Nguyen et al. (2007) proposed



Dynamic Positioning Control Systems for Ships and Underwater Vehicles, Fig. 1 Controller structure

the design of hybrid controller for DP from calm to extreme sea conditions.

Plant Control

By copying the control plant model (8)–(13) and adding an injection term, a passive observer may be designed. A nonlinear output horizontal-plane positioning feedback controller of PID type may be formulated as

$$\tau_{PID} = -\mathbf{R}_e^T \mathbf{K}_p \mathbf{e} - \mathbf{R}_e^T \mathbf{K}_{p3} \mathbf{f}(\mathbf{e}) - \mathbf{K}_d \tilde{\mathbf{v}} - \mathbf{R}^T \mathbf{K}_i \mathbf{z}, \quad (15)$$

where $\mathbf{e} \in \mathbb{R}^3$ is the position and heading deviation vector, $\tilde{\mathbf{v}} \in \mathbb{R}^3$ is the velocity deviation vector, $\mathbf{z} \in \mathbb{R}^3$ is the integrator states, and $\mathbf{f}(\mathbf{e})$ is a third-order stiffness term defined as

$$\mathbf{e} = [e_1, e_2, e_3]^T = \mathbf{R}^T(\phi_d)(\hat{\boldsymbol{\eta}} - \boldsymbol{\eta}_d),$$

$$\tilde{\mathbf{v}} = \hat{\mathbf{v}} - \mathbf{R}^T(\phi_d)\boldsymbol{\eta}_d,$$

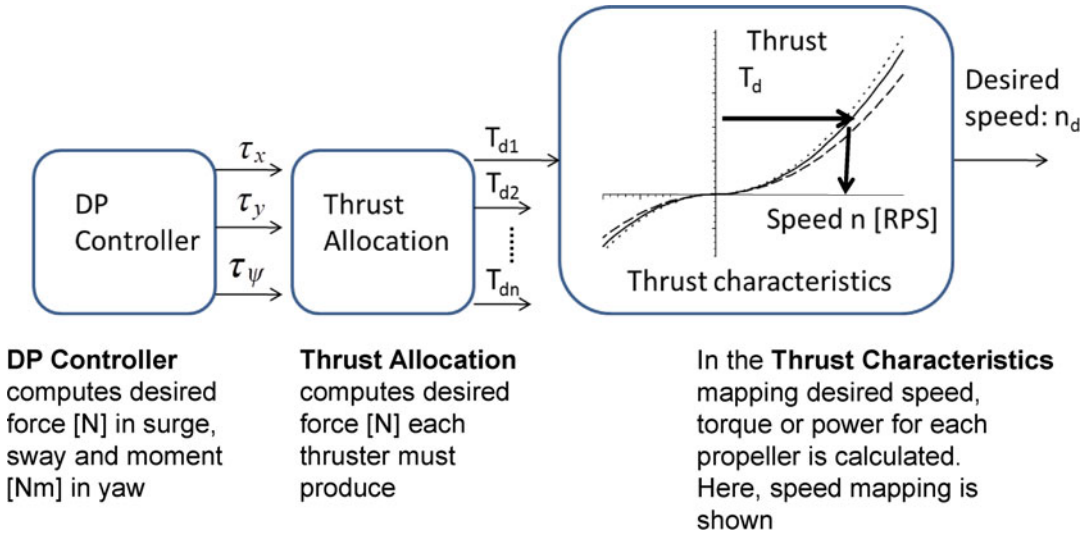
$$\dot{\mathbf{z}} = \hat{\boldsymbol{\eta}} - \boldsymbol{\eta}_d,$$

$$\mathbf{R}_e = \mathbf{R}(\phi - \phi_d) = \mathbf{R}^T(\phi_d)\mathbf{R}(\phi),$$

$$\mathbf{f}(\mathbf{e}) = [e_1^3, e_2^3, e_3^3]^T.$$

Experience from implementation and operations of real DP control systems has shown that ϕ_d in the calculation of the error vector $\mathbf{e}$ generally gives a better performance with less noisy signals than using ϕ . However, this is only valid under the assumption that the vessel maintains its desired heading with small deviations. As the DP capability for ships is sensitive to the heading angle, i.e., minimizing the environmental loads, heading control is prioritized in case of limitations in the thrust capacity. This feature is handled in the thrust allocation.

An advantage of this is the possibility to reduce the first-order proportional gain matrix, resulting in reduced dynamic thruster action for smaller position and heading deviations. Moreover, the third-order restoring term will make the thrusters to work more aggressive for larger deviations. $\mathbf{K}_p$, $\mathbf{K}_{p3}$, $\mathbf{K}_d$, and $\mathbf{K}_i \in \mathbb{R}^{3 \times 3}$ are the nonnegative controller gain matrices for proportional, third-order restoring,



Dynamic Positioning Control Systems for Ships and Underwater Vehicles, Fig. 2 Thrust allocation

derivative, and integrator controller terms, respectively, found by appropriate controller synthesis methods.

For small-waterplane-area marine vessels such as semisubmersibles, often used as drilling rigs, Sørensen and Strand (2000) proposed a DP control law with the inclusion of roll and pitch damping according to

$$\tau_{rpd} = - \begin{bmatrix} 0 & g_{xq} \\ g_{yp} & 0 \\ g_{\phi p} & 0 \end{bmatrix} \begin{bmatrix} \hat{p} \\ \hat{q} \end{bmatrix}, \quad (16)$$

where $\hat{p}$ and $\hat{q}$ are the estimated pitch and roll angular velocities. The resulting positioning control law is written as

$$\tau = \tau_{wff} + \tau_{PID} + \tau_{rpd}, \quad (17)$$

where $\tau_{wff} \in \mathbb{R}^3$ is the wind feedforward control law.

Thrust allocation or control allocation (Fig. 2) is the mapping between plant and actuator control. It is assumed to be a part of the plant control. The DP controller calculated the desired force in surge and sway and moment in yaw. Dependent on the particular thrust configuration with installed and enabled propellers, tunnel thrusters, azimuthing thrusters, and rudders, the allocation

is a nontrivial optimization problem calculating the desired thrust and direction for each enabled thruster subject to various constraints such as thruster ratings, forbidden sectors, and thrust efficiency. References on thrust allocation are found in Johansen and Fossen (2013).

In Sørensen and Smogeli (2009), torque and power control of electrically driven marine propellers are shown. Ruth et al. (2009) proposed anti-spin thrust allocation, and Smogeli et al. (2008) and Smogeli and Sørensen (2009) presented the concept of anti-spin thruster control.

Cross-References

- [Control of Ship Roll Motion](#)
- [Fault-Tolerant Control](#)
- [Mathematical Models of Ships and Underwater Vehicles](#)
- [Motion Planning for Marine Control Systems](#)
- [Underactuated Marine Control Systems](#)

Bibliography

- Balchen JG, Jenssen NA, Sælid S (1976) Dynamic positioning using Kalman filtering and optimal control theory. In: IFAC/IFIP symposium on automation in offshore oil field operation, Amsterdam, pp 183–186

Basseville M, Nikiforov IV (1993) Detection of abrupt changes: theory and application. Prentice-Hall, Englewood Cliffs. ISBN:0-13-126780-9

Blanke M, Kinnaert M, Lunze J, Staroswiecki M (2003) Diagnostics and fault-tolerant control. Springer, Berlin

Faltinsen OM (1990) Sea loads on ships and offshore structures. Cambridge University Press, Cambridge

Fay H (1989) Dynamic positioning systems, principles, design and applications. Editions Technip, Paris. ISBN:2-7108-0580-4

Fossen TI (2011) Handbook of marine craft hydrodynamics and motion control. Wiley, Chichester

Fossen TI, Strand JP (1999) Passive nonlinear observer design for ships using Lyapunov methods: experimental results with a supply vessel. *Automatica* 35(1):3–16

Grimble MJ, Johnson MA (1988) Optimal control and stochastic estimation: theory and applications, vols 1 and 2. Wiley, Chichester

Johansen TA, Fossen TI (2013) Control allocation—a survey. *Automatica* 49(5):1087–1103

Johansen TA, Sørensen AJ, Nordahl OJ, Mo O, Fossen TI (2007) Experiences from hardware-in-the-loop (HIL) testing of dynamic positioning and power management systems. In: OSV Singapore, Singapore

Newman JN (1977) Marine hydrodynamics. MIT, Cambridge

Nguyen TD, Sørensen AJ (2009) Setpoint chasing for thruster-assisted position mooring. *IEEE J Ocean Eng* 34(4):548–558

Nguyen TD, Sørensen AJ, Quek ST (2007) Design of hybrid controller for dynamic positioning from calm to extreme sea conditions. *Automatica* 43(5):768–785

Pettersen KY, Fossen TI (2000) Underactuated dynamic positioning of a ship – experimental results. *IEEE Trans Control Syst Technol* 8(4):856–863

Ruth E, Smogeli ØN, Perez T, Sørensen AJ (2009) Anti-spin thrust allocation for marine vessels. *IEEE Trans Control Syst Technol* 17(6):1257–1269

Sælid S, Jenssen NA, Balchen JG (1983) Design and analysis of a dynamic positioning system based on Kalman filtering and optimal control. *IEEE Trans Autom Control* 28(3):331–339

Smogeli ØN, Sørensen AJ (2009) Antispin thruster control for ships. *IEEE Trans Control Syst Technol* 17(6):1362–1375

Smogeli ØN, Sørensen AJ, Minsaas KJ (2008) The concept of anti-spin thruster control. *Control Eng Pract* 16(4):465–481

Strand JP, Fossen TI (1999) Nonlinear passive observer for ships with adaptive wave filtering. In: Nijmeijer H, Fossen TI (eds) New directions in nonlinear observer design. Springer, London, pp 113–134

Strand JP, Sørensen AJ, Fossen TI (1998) Design of automatic thruster assisted position mooring systems for ships. *Model Identif Control* 19(2):61–75

Sørensen AJ (2005) Structural issues in the design and operation of marine control systems. *Annu Rev Control* 29(1):125–149

Sørensen AJ (2011) A survey of dynamic positioning control systems. *Annu Rev Control* 35:123–136.

Sørensen AJ, Smogeli ØN (2009) Torque and power control of electrically driven marine propellers. *Control Eng Pract* 17(9):1053–1064

Sørensen AJ, Strand JP (2000) Positioning of small-waterplane-area marine constructions with roll and pitch damping. *Control Eng Pract* 8(2):205–213

Sørensen AJ, Dukan F, Ludvigsen M, Fernandez DA, Candeloro M (2012) Development of dynamic positioning and tracking system for the ROV Minerva. In: Roberts G, Sutton B (eds) Further advances in unmanned marine vehicles. IET, London, pp 113–128. Chapter 6

Dynamical Social Networks

Chiara Ravazzi¹ and Anton Proskurnikov^{2,3}

¹Institute of Electronics, Computer and Telecommunication Engineering (IEIT), National Research Council of Italy (CNR), Turin, TO, Italy

²Department of Electronics and Telecommunications, Politecnico di Torino, Turin, TO, Italy

³Institute for Problems in Mechanical Engineering, Russian Academy of Sciences, St. Petersburg, Russia

Abstract

The classical field of social network analysis (SNA) considers societies and social groups as *networks*, assembled of social actors (individuals, groups or organizations) and relationships among them, or social ties. From the systems and control perspective, a social network may be considered as a complex dynamical system where an actor's attitudes, beliefs, and behaviors related to them evolve under the influence of the other actors. As a result of these local interactions, complex dynamical behaviors arise that depend on both individual characteristics of actors and the structural properties of a network. This entry provides basic concepts and theoretical tools elaborated to study dynamical social networks. We focus

on (a) structural properties of networks and (b) dynamical processes over them, e.g., dynamics of opinion formation.

Keywords

Social network · Dynamical network ·
Centrality measure · Opinion formation

Introduction

Pioneer works of Moreno and Jennings (1938) have opened up a new interdisciplinary field of study, which is nowadays called social network analysis (SNA) (Freeman 2004). SNA has anticipated and inspired, to a great extent, the general theory of complex networks (Newman 2003) developed in the recent decades. Many important structural properties of networks, e.g., centrality measures, cliques, and communities, have arisen as characteristics of social groups and have been studied from a sociological perspective (Freeman 2004). On a parallel line of research, *dynamics* over social networks have been studied that are usually considered as processes of information diffusion over networks (Del Vicario et al. 2016; Arnaboldi et al. 2016) and evolution of individual opinions, attitudes, beliefs, and actions related to them under social influence (Friedkin 2015; Proskurnikov and Tempo 2017, 2018).

The theory of dynamical social networks (DSNs) combines the two aforementioned lines of research and considers a social network as a dynamical system, aiming at understanding the interplay between the network's structural properties and behaviors of dynamical processes over the network. The ultimate and long-standing goal is to develop the theory of *temporal* social networks, where both vertices (standing for social actors) and edges (standing for social relations) can emerge and disappear.

In this entry, we provide a brief overview of the main concepts and tools, related to the theory of DSN, focusing on its system- and control-theoretic aspects.

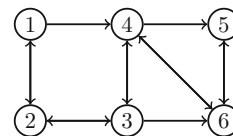
Mathematical Representation of a Social Network

Mathematically, a social network is naturally represented by a directed *graph*, that is, a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ constituted by a finite set of *nodes* (or *vertices*) $\mathcal{V}$ and a set of (directed) *arcs* (or *edges*) $\mathcal{E}$ (Fig. 1). The nodes represent social actors (individuals or organizations), and the arcs stand for relations (influence, dependence, similarity, etc.).

The structure of arcs can be encoded by the adjacency matrix $A = (a_{ij})$, whose entry is $a_{uv} = 1$ if $(u, v) \in \mathcal{E}$ and $a_{uv} = 0$ otherwise. More generally, it is often convenient to assign heterogeneous *weights* to the arcs that can quantify the relations' strengths (Liu 2012). In this case, a *weighted* adjacency matrix $W = (w_{ij})$ is introduced, where $w_{uv} \neq 0$ if and only if $(u, v) \in \mathcal{E}$. A triple $(\mathcal{V}, \mathcal{E}, W)$ is referred to as a *weighted* (or *valued*) graph. In some problems (e.g., structural balance theory (Cartwright and Harary 1956; Marvel et al. 2011; Facchetti et al. 2011)), it is convenient to distinguish positive and negative relationships (trust/distrust, friendship/enmity, cooperation/competition, etc.) assigning thus positive and negative weights to the arcs; such a graph is said to be *signed*.

Structural Properties of Social Networks

At the dawn of SNA, it was realized that a social group is more than a sum of individual actors and cannot be examined apart from the structure of social relationships or *ties*. Numerous characteristics have been proposed in the literature to



Dynamical Social Networks, Fig. 1 Example of a directed graph with six nodes: $\mathcal{V} = \{1, \dots, 6\}$ and $\mathcal{E} = \{(1, 2), (1, 4), (2, 1), (2, 3), (3, 2), (3, 4), (3, 6), (4, 3), (4, 5), (4, 6), (5, 6), (6, 4), (6, 5)\}$

quantify the structure of social ties and social actors' positions within a network. Many of these characteristics prove to be useful in analysis of general large-scale networks (Newman 2003).

Density Characteristics

Many large-scale social networks are *sparse* in the sense that the number of the (directed) arcs is much smaller than the maximal possible number n^2 , where $n = |\mathcal{V}|$ is the number of nodes (also referred to as the network size). See, e.g., SNAP dataset collection (Leskovec and Sosič 2016) and The Colorado Index of Complex Networks at <https://icon.colorado.edu/#!/networks>. These social networks can be easily visualized using the software GraphViz (Ellson et al. 2001) or Gephi <https://gephi.org/users/download/>. Another example (Bokuniewicz and Shulman 2017) is the structure of Twitter networks of marketing organizations with a very sparse adjacency matrix. A natural coefficient to quantify the network's sparsity property is the *network density* $\rho = |\mathcal{E}|/n^2$.

Another way to quantify the sparsity of the network is to consider the *degree* of a node. If, from sociological perspective, an agent is influenced from few friends, then the in-degree, i.e., the number of edges incoming to a specific node, is low compared to the size of the network. As a consequence, the corresponding row in the adjacency matrix is sparse, i.e., with few nonzero elements. Many real-world networks are "scale-free" networks (Newman 2003) with a degree distribution of the form $p(k) \propto k^{-\gamma}$ with $\gamma \in (2, 3)$ (here $p(k)$ is the proportion of nodes having in-degree $k = 0, 1, 2, \dots$); it is remarkable that similar distributions were discovered in the early works on sociometry (Moreno and Jennings 1938).

Many social networks are featured by the presence of few clusters or *communities* (Arnaboldi et al. 2016; Newman 2003). Social actors within a community are densely connected, whereas relationships between different communities are sparse. These types of networks are described by an influence matrix that can be decomposed as a sum of a low-rank matrix and a sparse matrix. The sparseness, scale-freeness, and clus-

tering are common properties that are regularly exploited in data storage (Tran 2012), reconstruction of influence networks (Ravazzi et al. 2018), and identification of communities in large-scale social networks (Blondel et al. 2008). Sparsity is also related to structural controllability (Liu et al. 2011): in some sense, sparse networks are the most difficult to control, whereas dense networks can be controlled via a few driver nodes.

Centrality Measures, Resilience, and Structural Controllability

The problem of identifying the most "important" (influential) nodes in the social network dates back to the first works on sociometry in the 1930s. At a "local" level, the influence of a node can be measured by its degree, that is, the number of social actors influenced by the corresponding individuals. However, a person that is influential within his/her small community need not be a global opinion leader. To distinguish globally influential nodes, various *centrality measures* have been introduced (Friedkin 1991; Freeman 2004; Newman 2003), among which the most important are *closeness*, *betweenness*, and *eigenvector centralities*.

In a connected network, the closeness centrality relates to how long it will take to spread information from a node u to all other nodes in the network. Mathematically,

$$c_u = \frac{1}{\sum_{v \in \mathcal{V} \setminus \{u\}} d_{uv}}, \quad \forall u \in \mathcal{V}.$$

where d_{uv} is the length of the shortest path between u and v .

Betweenness of node u (Freeman 1977) is defined as

$$b_u = \sum_{j, k \in \mathcal{V}, j \neq k \neq u} \frac{|S_u(j, k)|}{|S(j, k)|}$$

where $S(j, k)$ denotes the set of shortest paths from j to k and $S_u(j, k)$ the set of shortest paths from j to k that contain the node u . Hence, the more shortest paths pass through node u , the higher is its betweenness. Similarly, the *edge betweenness* can be introduced (Newman

2003) that measures the number of shortest paths containing an edge. Nodes and edges of high betweenness serve as “bridges” between other nodes.

Eigenvector centrality measures the influence of an individual in a social network by means of the leading eigenvector π^* of a suitable weighted adjacency matrix M , i.e., the scores of the nodes π_u^* are found from the equations

$$\pi_u^* = \frac{1}{\lambda} \sum_{v \in \mathcal{V}} M_{uv} \pi_v^* \quad (1)$$

where λ is the leading eigenvalue of M . This idea leads to Bonacich, Katz, and Freeman centrality measures (Friedkin 1991) and the PageRank (Brin and Page 1998). The main principle of eigenvector centrality is to assign high scores to nodes whose neighbors are highly ranked.

In particular, PageRank is computed as in (1) defining

$$M = (1 - m)A + \frac{m}{n} \mathbf{1}\mathbf{1}^\top,$$

where m is a scalar parameter, usually set to 0.15, n is the graph’s size and A is the adjacency matrix of a graph renormalized to be row stochastic (i.e., $a_{ij} = 1/d_i$ if i and j are connected, where d_i stands for the degree of node i).

Considering the simple graph in Fig. 1, it should be noticed that different centrality measures may produce very different ranking (see Tables 1 and 2).

Identification of the most influential nodes may be also considered as an analysis of network’s *resilience* properties, that is, its ability

Dynamical Social Networks, Table 1 Centrality measures of nodes in the graph of Fig. 1

Node	Out-degree	Closeness	Betweenness	PageRank
1	2	15	0.833	0.061
2	2	11	0.5	0.085
3	3	9	2.166	0.122
4	3	7	3.666	0.214
5	1	9	0	0.214
6	2	7	0.5	0.302

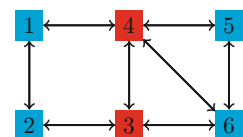
Dynamical Social Networks, Table 2 Ranking of nodes in the graph of Fig. 1 according to different centrality measures

Centrality measure	Ranking
Degree	(3,4,1,2,6,5)
Closeness	(1,2,3,5,4,6)
Betweenness	(4,3,1,2,6,5)
PageRank	(6,4,5,3,2,1)

to counteract various malicious attacks. This problem is especially important in online social media where attacks on the “most central” nodes allow to disseminate fake news, rumors, and other sorts of disinformation. At the same time, users of such media can manipulate their ranks (e.g., by adding fictitious profiles and/or connections), which leads to another important problem: how does the PageRank change under perturbations of a network, e.g., how large values of the rank can one obtain by activating some “hidden” links (Csáji et al. 2014). As has been shown in Como and Fagnani (2015), families of graphs with “fast-mixing” properties, such as expander graphs, are more resilient to localized perturbations than “slow-mixing” ones that exhibit information diffusion bottlenecks. The mixing-time properties of graphs are related to robustness of stochastic matrices against perturbations (Como and Fagnani 2015).

The aforementioned resilience problems are closely related to a problem of *controlling* centrality measures in *weighted graphs*, addressed in Nicosia et al. (2012). It has been shown that a relatively small “control set” of nodes can produce an *arbitrary* centrality vector by assigning influence weights to their neighbors. To compute the minimal controlling set is a difficult problem, reducing to the NP-hard problem of identifying the minimal *dominating set* in the graph, that is, the set of nodes such that any node is either dominating or adjacent to at least one dominating node (Fig. 2).

Dynamical Social Networks, Fig. 2 The minimal dominating set $S = \{3, 4\}$ of two nodes

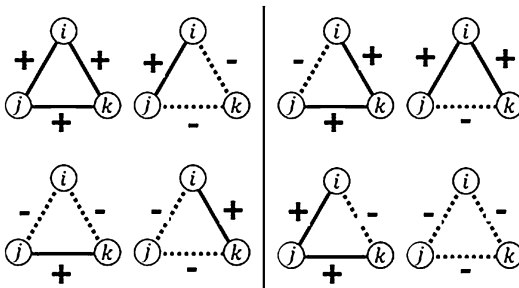


Structural Balance in Signed Networks

Structural balance is an important property of a social network with signed weights on the arcs, standing for positive and negative relations among individuals (trust/distrust, cooperation/competition, friendship/enmity, etc.). The concept of structural balance originates from the works of Heider (1946), further developed by Cartwright and Harary (1956). As Heider argued (Heider 1946), “an attitude towards an event can alter the attitude towards the person who caused the event, and, if the attitudes towards a person and an event are similar, the event is easily ascribed to the person.” The same applies to attitudes toward people and other objects. Heider’s theory predicts the emergence of a *balanced* state where positive and negative relations among individuals are in harmony with their attitudes to people, things, ideas, situations, etc. If actor *A* likes actor *B* in the balanced state, then *A* shares with *B* a positive or negative appraisal of any other individual *C*; vice versa, if *A* dislikes *B*, then *A* disagrees with any *B*’s appraisal of *C*.

In other words, if a social network corresponds to a complete signed graphs where each actor has a positive or negative appraisal of all other actors, then it tends to a balanced state satisfying four simple axioms (Fig. 3):

1. A friend of my friend is my friend.
2. A friend of my enemy is my enemy.
3. An enemy of my friend is my enemy.
4. An enemy of my enemy is my friend.



Dynamical Social Networks, Fig. 3 Balanced (left) vs. imbalanced (right) triads

The violation of these rules leads to tensions and cognitive dissonances that have to be somehow resolved; the description of such tension resolving mechanisms is beyond Heider’s theory, and their mathematical modeling is a topic of ongoing research.

If the signed complete graph is balanced, then its node can be divided into two opposing factions, where members of the same factions are “friends” and every two actors from different factions are “enemies” (if all relations are positive, then one of the factions is empty). This property can be used as a definition of the structural balance in a more general situation where the graph is not necessarily complete (Cartwright and Harary 1956); a strongly connected graph is decomposed into two opposing factions if and only if the product of signs along any path connecting two nodes *i* and *j* is the same and depends only on the pair (*i*, *j*) (equivalently, the product of all signs over a semicycle is positive). Experimental studies reveal that many large-scale networks are close to the balanced state (Facchetti et al. 2011).

Dynamical Processes over Social Networks

Spread of information (including fake news and rumors), evolution of attitudes, beliefs and actions related to them, and strengthening and weakening of social ties are examples of dynamic processes unfolding over social networks. In this entry, we give a brief overview of the most “mature” results on social dynamics obtained in the systems and control literature and the related mathematical concepts.

Opinions and Models of Opinion Formation

As discussed in Friedkin (2015), an individual’s opinion is his/her cognitive orientation toward some object, e.g., particular issue, event, action, or another person. Opinions can stand for displayed attitudes or subjective certainties of belief. Mathematically, opinions are usually modeled as either elements of a discrete set (e.g., the decision

to support some action to withstand it, the name of a presidential election candidate to vote for) or scalar and vector quantities, belonging to a subset of the Euclidean space.

One of the simplest models with discrete opinions is known as the *voter model* (Clifford and Sudbury 1973). Each node of a network is associated to an actor (“voter”) whose opinion is a binary value (0 or 1). At each step, a random voter is selected who replaces his/her opinion by the opinion of a randomly chosen neighbor in the graph. Other examples include, but are not limited to, Granovetter’s threshold model (Granovetter 1978) (an individual supports some action if some proportion of his/her neighbors support it), Schelling model or spatial segregation (Schelling 1971) (the graph stands for a network of geographic locations, and an opinion is the preferred node to live), Axelrod’s model of culture dissemination (Axelrod 1997) (an opinion stands for a set of cultural traits), and the Ising model of phase transition adapted to social behaviors (Sznajd-Weron and Sznajd 2000). The aforementioned models are usually examined by tools of advanced probability theory and statistical physics (Castellano et al. 2009).

Control-theoretic studies on opinion dynamics have primarily focused on models with real-valued (“continuous”) opinions, which can attain a continuum of values. We consider only microscopic or agent-based models of opinion formation, portraying evolution of individual opinions and described by systems of differential or recurrent equations, whose dimension is proportional to the size of the network (number of social actors). As the number of actors tends to infinity, the dynamics transforms into a macroscopic (statistical, Eulerian, fluid-based, mean-field) model that portrays the evolution of opinion *distribution* (a density function or a probability measure) and is usually described by a partial differential or integral equation (Canuto et al. 2012).

Models of Rational Consensus: Social Power

A simple model of opinion formation was proposed by French (1956), examined by Harary (1959), and generalized by DeGroot (1974) as

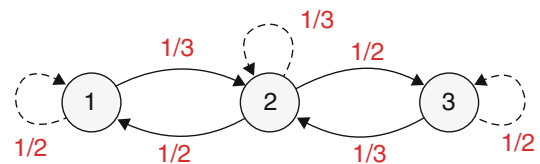
an iterative procedure to reach a consensus. Each social actor in a network keeps a scalar or vector opinion $x_i \in \mathbb{R}^d$ (e.g., a vector of subjective probabilities (DeGroot 1974; Wai et al. 2016)). At each stage of the opinion evolution, the opinions are simultaneously recalculated based on a simple rule of iterative averaging

$$x_i(t+1) = \sum_{j=1}^n w_{ij} x_j(t) \quad \forall i = 1, \dots, n, \quad t = 0, 1, \dots \quad (2)$$

Here n stands for the number of social actors, and $W = (w_{ij})$ is a stochastic (A $n \times n$ matrix W is (row-)stochastic if all its entries are nonnegative $w_{ij} \geq 0$, and each row sums up to 1, i.e., $\sum_{j=1}^n w_{ij} = 1$ for each $i = 1, \dots, n$.) matrix of *influence weights*. The matrix W should be compatible with the graph of a social network, that is, an actor allocates influence weight to the adjacent actors (including him/herself if the node has a self-arc). The influence can be thought of as a finite resource each individual distributes to self and the others (Friedkin 2015). In the original French’s model, this distribution is uniform: if a node has out-degree m , each neighboring node gets the weight $1/m$, e.g., the graph structure in Fig. 4 induces the following rule of opinion update (for simplicity, we assume that opinions are scalar)

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \end{bmatrix} = \begin{pmatrix} 1/2 & 1/2 & 0 \\ 1/3 & 1/3 & 1/3 \\ 0 & 1/2 & 1/2 \end{pmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}.$$

A continuous-time counterpart of model (2) was proposed by Abelson (1964)



Dynamical Social Networks, Fig. 4 An example of the French model with $n = 3$ agents

$$\dot{x}_i(t) = \sum_{j=1}^n a_{ij}(x_j(t) - x_i(t)), \quad i = 1, \dots, n, \\ t \geq 0. \quad (3)$$

The coefficients $a_{ij} \geq 0$ play the role of infinitesimal influence weights; the matrix $A = (a_{ij})$ is not necessarily stochastic yet should be compatible with the graph of the social network: $a_{ij} > 0$ if and only if $(i, j) \in \mathcal{E}$.

The most typical behavior of the French-DeGroot and the Abelson model is *consensus* of the opinions, that is, their convergence to a common value $x_i(t) \xrightarrow{t \rightarrow \infty} x_*$ that depends on the initial condition. For this reason, these dynamical systems are often referred to as the *consensus protocols* and have been thoroughly studied in the literature on multi-agent systems (see the survey in Proskurnikov and Tempo 2017, 2018); first consensus criteria have been found by Harary (1959) and Abelson (1964). Consensus is established, for instance, whenever the graph of the social network contains a globally reachable node, and, in the discrete-time case, this node has a self-arc.

It is remarkable, however, that the original goal of French's work was not concerned with consensus but rather aimed at finding numerical characteristics of *social power*, that is, an individual's capability to influence the group's ultimate decisions. The consensus is established in the French DeGroot model if and only if the matrix W has a unique left eigenvector p corresponding to eigenvalue 1 such that (stochastic matrices satisfying this property are known as fully regular or SIA (stochastic indecomposable aperiodic) matrices; such a matrix serves as a transition matrix of some regular (ergodic) Markov chains (Proskurnikov and Tempo 2017). The vector p is nonnegative in view of the Perron-Frobenius theorem and stands for the unique stationary distribution of the corresponding Markov chain.)

$$p^\top W = p^\top, \quad \sum_{i=1}^n p_i = 1.$$

Using (2), one shows that

$$\begin{aligned} \sum_i p_i x_i(k+1) &= \sum_j \left(\sum_i p_i w_{ij} \right) x_j(k) \\ &= \sum_j p_j x_j(k) = \sum_i p_i x_i(k) \\ &= \dots = \sum_i p_i x_i(0), \end{aligned}$$

and passing to the limit as $k \rightarrow \infty$, one finds the consensus value x_* as follows:

$$x_* = \left(\sum_i p_i \right) x_* = \sum_i p_i x_i(0).$$

The component p_i can be thus considered as the influence of actor i 's opinion on the final opinion of the group. This measure of influence, or *social power*, of an individual is similar to the eigenvector centrality (computed for the normalized adjacency matrix).

Most of the properties of the French-DeGroot models retain their validity in the case of asynchronous *gossip-based* communication (Fagnani and Zampieri 2008) where at each stage of the opinion iteration, two randomly chosen actors interact, whereas the remaining actors' opinions remain unchanged. A gossip-based model portrays spontaneous interactions between people in real life and can be considered as a model of opinion formation over a random temporal (time-varying) graph.

From Consensus of Opinions to Community Cleavage

Whereas the French-DeGroot and the Abelson model predict consensus, real social groups often exhibit disagreement and clustering of opinions, even when the graph is strongly connected. Thus a realistic dynamic model of opinion formation should be able to explain both consensus and disagreement and help to disclose conditions that prevent the individuals from reaching a consensus. To find such models is a long-standing challenging problem in mathematical sociology referred to as Abelson's "diversity puzzle" or the

problem of *community cleavage* (Friedkin 2015; Proskurnikov and Tempo 2017).

Most of the opinion cleavage models studied in control theory stem from the French-DeGroot and the Abelson models. The three main classes of such models are:

1. Models with stubborn individuals
2. Homophily-based (bounded confidence) models
3. Models with antagonistic interactions

Models with Stubborn Individuals

The **first** class of models explains disagreement by stubbornness or “zealotry” of some individuals who are resistant to opinion assimilation. In the French-DeGroot model (2), such an actor assigns the maximal weight $w_{ii} = 1$ to him/herself being unaffected by the others’ opinions ($w_{ij} = 0$ for $j \neq i$), so that his/her opinion remains unchanged $x_i(t+1) = x_i(t) = \dots = x_i(0)$. Similarly, in the Abelson model, a stubborn individual assigns zero influence weights $a_{ij} = 0$ to all individuals, so that $\dot{x}_i(t) \equiv 0$. In the presence of two or more stubborn actors with different opinions, there is no consensus in the group, and the opinions split into several clusters (in the generic situation, all steady opinions are different).

A natural extension of the French-DeGroot model with stubborn individuals is the Friedkin-Johnsen model (Friedkin 2015), being an important and indispensable part of the social influence network theory (SINT) (Friedkin and Johnsen 2011). In the Friedkin-Johnsen model, actors are allowed to be “partially” stubborn. Namely, along with the stochastic matrix of influence weights $W = (w_{ij})$, a set of *susceptibility* coefficients $0 \leq \lambda_{11}, \dots, \lambda_{nn} \leq 1$ (or, equivalently, a diagonal matrix $0 \leq \Lambda \leq I_n$) is introduced that measure how strong is the influence of the actor’s initial opinion on all subsequent opinions:

$$x_i(t+1) = \lambda_{ii} \sum_j w_{ij} x_j(t) + (1 - \lambda_{ii}) x_i(0),$$

$$\forall i = 1, \dots, n \quad \forall t = 0, 1, \dots$$

The French-DeGroot model appears as a special case of the Friedkin-Johnsen system where none of the actors is anchored at his/her initial opinion ($\lambda_{ii} = 1$ for all i). If $\lambda_{ii} < 1$ for some i , then the matrix AW is typically Schur stable (has the spectral radius < 1). Assuming for simplicity that the opinions $x_i(t)$ are scalars, the vector $x(t) = (x_1(t), \dots, x_n(t))^T$ they constitute converges to

$$x(\infty) = \lim_{t \rightarrow \infty} x(t) = Vx(0),$$

$$V = (I - AW)^{-1}(I - A).$$

The matrix V (“control matrix” (Friedkin 2015)) is stochastic, but its rows are different. The average steady opinion of the group is thus found as

$$\frac{1}{n} \sum_{j=1}^n x_j(\infty) = \sum_j c_j x_j(0), \quad c_j = \frac{1}{n} \sum_{i=1}^n v_{ij}.$$

The vector $c = (c_1, \dots, c_n)$ is a natural generalization of French’s social power and measures the influence of the actors’ initial opinions on the average opinion of the group; it can thus also be thought of as a centrality measure. As discussed in Proskurnikov and Tempo (2017), in the case where $\Lambda = \alpha I$, $\alpha \in (0, 1)$, this centrality measure (first introduced in Friedkin 1991) is nothing else than the conventional PageRank.

Homophily-Based Models

The principle of homophily can be formulated as “birds of a feather fly together” (McPherson et al. 2001). Social actors tend to interact with like-minded individuals and assimilate their opinions easier than an opinion of a dissimilar person. This principle is prominently illustrated by models of bounded confidence, similar to the French-DeGroot and the Abelson model yet allowing the matrix (w_{ij}) (respectively, (a_{ij})) to depend on the system’s state. The Hegselmann-Krause model (Hegselmann and Krause 2002) is an extension of (2), stipulating that $w_{ij}(x_i, x_j) = 0$ if $|x_i - x_j| \geq \varepsilon$, where $\varepsilon > 0$ is an actor’s *confidence bound*. A gossip-based version of this model is due to Deffuant et al. (2000). A detailed discussion of bounded confidence models

(which, e.g., may contain partially stubborn individuals) and their convergence properties is available at the survey (Proskurnikov and Tempo 2018).

Antagonistic Interactions

Abelson (1964) suggested that one of the reasons for disagreement can be “boomerang” (reactance, anticonformity) effect: an attempt to convince other people can make them to adopt an opposing position. In other words, social actors do not always bring their opinions closer, and the convex-combination mechanisms of the models (2), (3) have to be generalized to allow repulsion between their opinions. This idea is in harmony with Heider’s theory of structural balance, predicting that individuals disliking each other should have opposite positions on every issue. Although the presence of negative ties in opinion formation models has not been secured experimentally, the mathematical theory of networks with positive and negative ties is important since such networks arise in abundance in biology and economics.

A natural extension of the French-DeGroot model allowing negative ties is known as the “discrete-time Altafini model” (Liu et al. 2017) (historically, its continuous-time counterpart was first proposed by Altafini; see the survey in Shi et al. (2019)) and deals with a system (2), where the weights w_{ij} can be positive and negative; however, their absolute values constitute a stochastic matrix ($|w_{ij}|$). It appears that a structurally balanced graph leads to bimodal polarization (or bipartite consensus) of the opinions: the actors in the two opposing factions converge on the two opposite opinions x_* and $(-x_*)$, where x_* depends on the initial opinion distribution. An imbalanced strongly connected graph induces, however, a degenerate behavior where all opinions converge to 0. In both situations, the opinions remain bounded. More sophisticated dynamic models have been proposed recently (Shi et al. 2019) that are able to explain clustering of opinions over graphs without structural balance property.

Inference of Networks’ Structures from Opinion Dynamics

Data-driven inference of a network’s characteristics using observations of some dynamical processes over the network is a topic of an active research in statistics, physics, and signal processing. In social networks, a natural dynamical process is opinion formation, which, as has been already mentioned, is closely related to centrality measures and other structural properties of a social graph.

The existing works on identification of opinion formation processes are primarily focused on linear models such as the French-DeGroot model (Wai et al. 2016) or the Friedkin-Johnsen model and its extensions (Ravazzi et al. 2018). Two different methods to infer the network’s structure are the *finite-horizon* and the *infinite horizon* identification procedures. In the finite-horizon approach, the opinions are observed for T subsequent rounds of conversation. Then, if enough observations are available, the parameters of the model (e.g., the matrix of influence weights $W = (w_{ij})$) can be estimated as the matrix best fitting the dynamics for $0 \leq k \leq T$, by using classical identification techniques.

The infinite-horizon approach, instead, performs the estimation based on the observations of the initial and final opinions’ profiles. One of the first works on the inference of the network topology is Wai et al. (2016), in which the authors consider French-DeGroot models with stubborn agents. This identification procedure has been adopted also in Ravazzi et al. (2018) to estimate the influence matrix of the Friedkin-Johnsen model, under the assumption that the matrix is sparse (i.e., agents are influenced by few friends). In these works, the estimation problem is non-convex, and deterministic or stochastic relaxation techniques are used to approximate it by semidefinite programs.

Dynamics of Social Graphs

Up to now, we have considered dynamics *over* social networks paying no attention to dynamics of networks themselves. Two examples where the network’s graph can be thought of as a temporal (time-varying) graphs are the aforementioned

models with gossip-based interactions (one may suppose that the graph of a network is random and contains a single arc at each step) and models with bounded confidence (if the opinions of two individuals are distant, the link connecting them may be considered as missing).

Even more sophisticated dynamical models describe the dynamics of *social power* under reflected appraisals (Jia et al. 2015; Ye et al. 2018), which is related to the French-DeGroot and the Friedkin-Johnsen model yet formally does not require to introduce opinions. It is assumed that social actors participate in infinitely many rounds of discussion, and, upon finishing each round, are able to find their social powers p_i (or, more generally, centralities c_i generated by the Friedkin-Johnsen model). The social power (an “appraisal” of an individual by the others) modifies the influence weights the individual assigns to the others in the next round, which, in turn, produces a new vector of social power (or Friedkin-Johnsen centralities).

Another important class of dynamical models describe the evolution of positive and negative ties leading to the structural balance. Whereas the balance is predicted by Heider’s theory, it does not provide any explicit description of the mechanisms balancing the network. Several mathematical models have been proposed in the literature (Marvel et al. 2011; Traag et al. 2013) to portray the dynamics of structural balance.

Cross-References

- [Consensus of Complex Multi-agent Systems](#)
- [Dynamic Graphs, Connectivity of](#)
- [Graphs for Modeling Networked Interactions](#)
- [Markov Chains and Ranking Problems in Web Search](#)

Bibliography

Abelson RP (1964) Mathematical models of the distribution of attitudes under controversy. In Frederiksen N, Gulliksen H (eds) Contributions to mathematical

- psychology. Holt, Rinehart & Winston, Inc, New York, pp 142–160
- Arnaboldi V, Conti M, Gala ML, Passarella A, Pezzoni F (2016) Ego network structure in online social networks and its impact on information diffusion. *Comput Commun* 76:26–41
- Axelrod R (1997) The dissemination of culture: a model with local convergence and global polarization. *J Conflict Resolut* 41:203–226
- Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. *J Stat Mech* 2008(10):P10008
- Bokuniewicz JF, Shulman J (2017) Influencer identification in twitter networks of destination marketing organizations. *J Hosp Tour Technol* 8(2): 205–219
- Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. *Comput Netw ISDN Syst* 30(1):107–117
- Canuto C, Fagnani F, Tilli P (2012) An Eulerian approach to the analysis of Krause’s consensus models. *SIAM J Control Optim* 50:243–265
- Cartwright D, Harary F (1956) Structural balance: a generalization of Heider’s theory. *Psychol Rev* 63:277–293
- Castellano C, Fortunato S, Loreto V (2009) Statistical physics of social dynamics. *Rev Mod Phys* 81:591–646
- Clifford P, Sudbury A (1973) A model for spatial conflict. *Biometrika* 60(3):581–588
- Como G, Fagnani F (2015) Robustness of large-scale stochastic matrices to localized perturbations. *IEEE Trans Netw Sci Eng* 2(2):53–64
- Csáji BC, Jungers RM, Blondel VD (2014) Pagerank optimization by edge selection. *Discrete Appl Math* 169(C):73–87
- Deffuant G, Neau D, Amblard F, Weisbuch G (2000) Mixing beliefs among interacting agents. *Adv Complex Syst* 3:87–98
- DeGroot MH (1974) Reaching a consensus. *J Am Stat Assoc* 69(345):118–121
- Del Vicario M, Bessi A, Zollo F, Petroni F, Scala A, Caldarelli G, Eugene Stanley H, Quattrociocchi W (2016) The spreading of misinformation online. *Proc Natl Acad Sci* 113(3):554–559
- Ellson J, Gansner E, Koutsofios L, North S, Gordon Woodhull, Short Description, and Lucent Technologies. (2001) *Graphviz: open source graph drawing tools*. In: *Lecture notes in computer science*. Springer, Berlin, Heidelberg, pp 483–484
- Facchetti G, Iacono G, Altafini C (2011) Computing global structural balance in large-scale signed social networks. *PNAS* 108(52):20953–20958
- Fagnani F, Zampieri S (2008) Randomized consensus algorithms over large scale networks. *IEEE J Sel Areas Commun* 26(4):634–649
- Freeman LC (1977) A set of measures of centrality based upon betweenness. *Sociometry* 40:35–41
- Freeman LC (2004) The development of social network analysis. A study in the sociology of science. Empirical Press, Vancouver

- French Jr JRP (1956) A formal theory of social power. *Psychol Rev* 63:181–194
- Friedkin NE (1991) Theoretical foundations for centrality measures. *Am J Sociol* 96(6):1478–1504
- Friedkin N (2015) The problem of social control and coordination of complex systems in sociology: a look at the community cleavage problem. *IEEE Control Syst Mag* 35(3):40–51
- Friedkin NE, Johnsen EC (2011) *Social influence network theory*. Cambridge University Press, New York
- Granovetter M (1978) Threshold models of collective behavior. *Am J Sociol* 83(6):1420–1443
- Harary F (1959) A criterion for unanimity in French's theory of social power. In: Cartwright D (ed) *Studies in social power*. University of Michigan Press, Ann Arbor, pp 168–182
- Hegselmann R, Krause U (2002) Opinion dynamics and bounded confidence, models, analysis and simulation. *J Artif Soc Soc Simul* 5(3):2
- Heider F (1946) Attitudes and cognitive organization. *J Psychol* 21:107–122
- Jia P, Mirtabatabaei A, Friedkin NE, Bullo F (2015) Opinion dynamics and the evolution of social power in influence networks. *SIAM Rev* 57(3):367–397
- Leskovec J, Sosič R (2016) Snap: a general-purpose network analysis and graph-mining library. *ACM Trans Intell Syst Technol (TIST)* 8(1):1
- Liu B (2012) *Sentiment analysis and opinion mining*. Morgan & Claypool Publishers, San-Rafael
- Liu Y-Y, Slotine J-J, Barabasi A-L (2011) Controllability of complex networks. *Nature* 473:167–173
- Liu J, Chen X, Basar T, Belabbas MA (2017) Exponential convergence of the discrete- and continuous-time Altafini models. *IEEE Trans Autom Control* 62(12):6168–6182
- Marvel SA, Kleinberg J, Kleinberg RD, Strogatz SH (2011) Continuous-time model of structural balance. *PNAS* 108(5):1771–1776
- McPherson M, Smith-Lovin L, Cook JM (2001) Birds of a feather: homophily in social networks. *Annu Rev Sociol* 27(1):415–444
- Moreno JL, Jennings HH (1938) Statistics of social configurations. *Sociometry* 1(3/4):324–374
- Newman ME (2003) The structure and function of complex networks. *SIAM Rev* 45(2):167–256
- Nicosia V, Criado R, Romance M, Russo G, Latora V (2012) Controlling centrality in complex networks. *Sci Rep* 2:218
- Proskurnikov AV, Tempo R (2017) A tutorial on modeling and analysis of dynamic social networks. Part I. *Annu Rev Control* 43:65–79
- Proskurnikov AV, Tempo R (2018) A tutorial on modeling and analysis of dynamic social networks. Part II. *Annu Rev Control* 45:166–190
- Ravazzi C, Tempo R, Dabbene F (2018) Learning influence structure in sparse social networks. *IEEE Trans Control Netw Syst* 5(4):1976–1986
- Schelling TC (1971) Dynamic models of segregation. *J Math Sociol* 1(2):143–186
- Shi G, Altafini C, Baras J (2019) Dynamics over signed networks. *SIAM Rev* 61(2):229–257
- Sznajd-Weron K, Sznajd J (2000) Opinion evolution in closed community. *Int J Mod Phys C* 11(06):1157–1165
- Traag VA, Van Dooren P, De Leenheer P (2013) Dynamical models explaining social balance and evolution of cooperation. *Plos One* 8(4):e60063
- Tran D (2012) Data storage for social networks. A socially aware approach. *SpringerBriefs in Optimization*, Springer, New York
- Wai H-T, Scaglione A, Leshem A (2016) Active sensing of social networks. *IEEE Trans Signal Inf Process Netw* 2:406–419
- Ye M, Liu J, Anderson BDO, Yu C, Başar T (2018) Evolution of social power in social networks with dynamic topology. *IEEE Trans Autom Control* 63(11):3793–3808

Dynamics and Control of Active Microcantilevers

Michael G. Ruppert¹ and S. O. Reza Moheimani²

¹The University of Newcastle, Callaghan, NSW, Australia

²The University of Texas at Dallas, Richardson, TX, USA

Abstract

The microcantilever is a key precision mechatronic component of many technologies for characterization and manipulation of matter at the nanoscale, particularly in the atomic force microscope. When a cantilever is operated in a regime that requires the direct excitation and measurement of its resonance frequencies, appropriate instrumentation and control is crucial for high-performance operation. In this entry, we discuss integrated cantilever actuation and present the cantilever transfer function model and its properties. As a result of using these active cantilevers, the ability to control the quality factor in order to manipulate the cantilever tracking bandwidth is demonstrated.

Keywords

Atomic force microscopy · Flexible structures · Piezoelectric actuators and sensors · Vibration control · Q control · Negative imaginary systems

Introduction

The microcantilever is a precision mechatronic device and an integral component of a myriad of systems which have collectively enabled the steady progress in nanotechnology over the past three decades. It is a key component and sits at the heart of a variety of instruments for manipulation and interrogation at the nanoscale, such as atomic force microscopes, scanning probe lithography systems, high-resolution mass sensors, and probe-based data storage systems. Despite its apparent simplicity, the fast dynamics, particularly when interacting with nonlinear forces arising at the micro- and nanoscale, have motivated significant research on the design, identification, instrumentation, and control of these devices.

One of the most important applications of the microcantilever is in atomic force microscopy (AFM) Binnig et al. (1986). In dynamic AFM, the cantilever is actively driven at its fundamental resonance frequency while a sample is scanned underneath a sharp tip located at its extremity. As the tip interacts with the sample, changes occur in the oscillation amplitude, phase, and/or frequency which relate to surface features and material properties. By controlling the vertical interaction between the tip and the sample with a feedback controller, 3D images with atomic resolution can be obtained Giessibl (1995).

Examples of cantilevers used for AFM are shown in Fig. 1. The most conventional approach is to use a passive rectangular cantilever with an extremely sharp tip at the end etched from single-crystal silicon as shown in Fig. 1a. In order to oscillate the cantilever, a piezo-acoustic bulk actuator is typically used which is clamped to the chip body Bhushan (2010). While this approach is simple and effective, it leads to highly distorted frequency responses with numerous structural

modes of the mounting system appearing in the frequency response as can be seen in Fig. 1c. This renders the identification and subsequent analysis of the actual cantilever dynamics exceedingly difficult and model-based controller design nearly impossible.

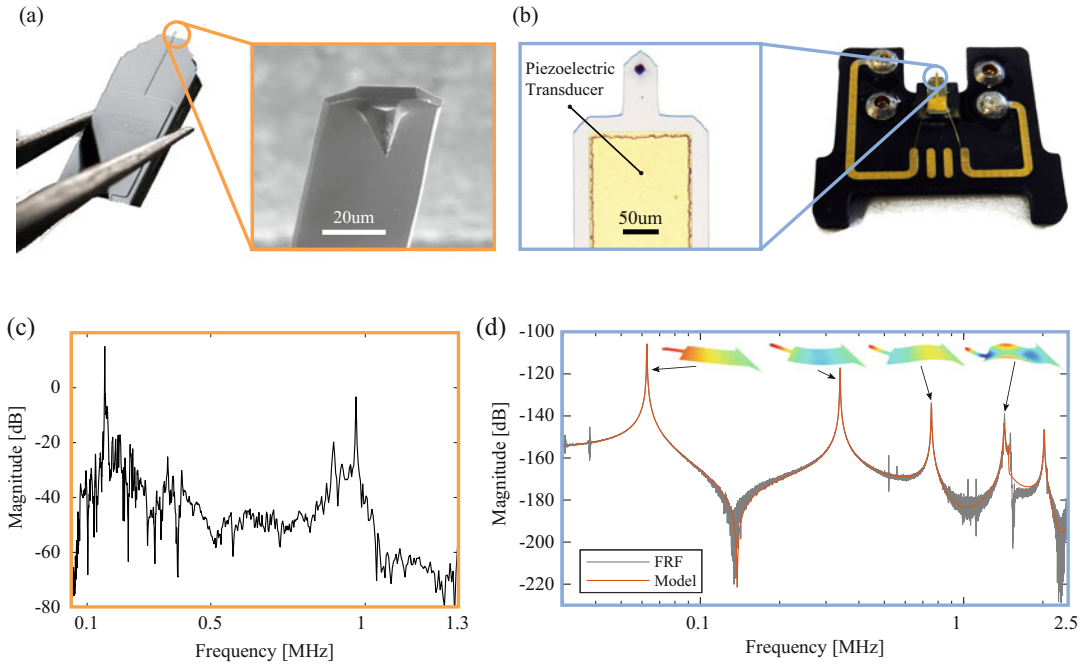
In contrast, active cantilevers with integrated actuation and sensing on the chip level provide several distinct advantages over conventional cantilever instrumentation Rangelow et al. (2017). Most importantly, these include clean frequency responses, the possibility of down-scaling, parallelization to cantilever arrays, as well as the absence of optical interferences. While a number of integrated actuation methods have been developed over the years to replace the piezo-acoustic excitation, only electro-thermal and piezoelectric actuation can be directly integrated on the cantilever chip. An example of such a cantilever is shown in Fig. 1b which features an integrated piezoelectric transducer which can be used for actuation, deflection sensing, or both Ruppert and Moheimani (2013). As a consequence, this type of cantilever yields a clean frequency response, such as shown in Fig. 1d, for which frequency domain system identification yields very good models suitable for controller design.

Microcantilever Dynamics

Active microcantilevers allow the utilization of second-order transfer function models to accurately model their dynamics. Moreover, these models have favorable properties for robust vibration control which will be discussed in the following.

Cantilever Model

When a voltage is applied to the electrodes of the piezoelectric layer of a cantilever such as in Fig. 1b, the transfer function from actuator voltage $V(s)$ to cantilever deflection $D(s)$ of the first n flexural modes can be described by a sum of second-order modes Moheimani and Fleming (2006)



Dynamics and Control of Active Microcantilevers, Fig. 1 (a) Image of a commercially available passive AFM microcantilever (Asylum Research High Quality AFM probe, image courtesy of Oxford Instruments Asylum Research, Santa Barbara, CA) and sharp tip (User:MaterialsScientist, Wikimedia Commons, CC BY-SA 3.0). (b) Image of an AFM cantilever with

integrated piezoelectric transducer (Bruker, DMASP) for integrated actuation/sensing. (c) Frequency response measured with the AFM optical beam deflection sensor of a standard base-excited passive cantilever. (d) Frequency response and estimated transfer function model (1) of a cantilever with integrated piezoelectric actuation

$$G_{dv}(s) = \frac{D(s)}{V(s)} = \sum_{i=1}^n \frac{\alpha_i \omega_i^2}{s^2 + \frac{\omega_i}{Q_i} s + \omega_i^2} + d, \quad \alpha_i \in \mathbb{R} \quad (1)$$

where each term is associated with a specific vibrational mode shape (shown in the inset of Fig. 1d), quality (Q) factor Q_i , natural frequency ω_i , and gain α_i . The feedthrough term d takes into account the modeling error associated with limiting the sum to modes within the measurement bandwidth.

For the first eigenmode of the cantilever ($i = 1$), the poles of (1) are given by

$$p_{1,2} = -\frac{\omega_1}{2Q_1} \pm j\omega_1 \sqrt{1 - \frac{1}{4Q_1^2}}. \quad (2)$$

The real part of the eigenvalues characterizes the amplitude transient response time or cantilever tracking bandwidth as

$$f^c = \frac{f_1}{2Q_1}. \quad (3)$$

Example 1 For a standard tapping-mode microcantilever (e.g., Budget Sensors, TAP190-G) with a fundamental resonance frequency of $f_1 = 190$ kHz and a natural Q factor of 400, the tracking bandwidth is only approximately $f^c = 238$ Hz.

System Property

An important property of the cantilever transfer function for Q factor control (Q control) is its negative imaginary (NI) nature. In general, the transfer function G of a system is said to be NI if it is stable and satisfies the condition Petersen

and Lanzon (2010)

$$j[G(j\omega) - G^*(j\omega)] \geq 0 \quad \forall \omega > 0. \quad (4)$$

Moreover, G is strictly negative imaginary (SNI) if (4) holds with a strict inequality sign. It is straightforward to show that the first mode of $G_{dv}(s)$ is NI since $\alpha_1, \omega_1, Q_1 > 0$. If additionally, collocated actuators and sensors are available, this property can be extended to multiple modes of the cantilever Ruppert and Yong (2017). The benefit of this structural property results from the negative imaginary lemma which states that the positive feedback interconnection of an SNI plant and an NI controller is robustly stable if and only if the DC-gain condition $G(0)K(0) < 1$ is satisfied Lanzon and Petersen (2008). In other words, even in the case where the resonance frequency, modal gain, and/or damping of a cantilever mode change, as long as the NI property and DC-gain condition hold, the closed-loop system remains stable.

Microcantilever Control

In dynamic-mode AFM, the output of the vertical feedback (z-axis) controller resembles the surface topography only when the cantilever is oscillating in steady state. Therefore, a fast decaying transient response is highly desirable. Reduction of the cantilever Q factor using an additional feedback loop enables it to react faster to sample surface features. This, in turn, allows the gain of the regulating z-axis controller to be increased which improves the cantilever's ability to track the surface topography.

Q Control Using Positive Position Feedback

The positive position feedback (PPF) controller of the form

$$K_{PPF}(s) = \sum_{i=1}^m \frac{\gamma_{c,i}}{s^2 + 2\zeta_{c,i}\omega_{c,i}s + \omega_{c,i}^2}, \quad (5)$$

with $\gamma_{c,i}$, $\zeta_{c,i}$, and $\omega_{c,i}$ being the tunable controller parameters is an example of a model-based controller which can be used to control the Q factor of m modes of the cantilever. From its structure, it can be verified that it fulfills the negative imaginary property if $\gamma_{c,i}, \zeta_{c,i}, \omega_{c,i} > 0$, that is, robust stability is achieved if such a controller is used to dampen a negative imaginary mode of the cantilever.

The closed-loop system of the cantilever first mode ($n = 1$) in positive feedback with the PPF controller ($m = 1$) (5) can be written as

$$\ddot{z} + \underbrace{\begin{bmatrix} \frac{\omega_1}{Q_1} & 0 \\ 0 & 2\zeta_c\omega_c \end{bmatrix}}_{E=E^T} \dot{z} + \underbrace{\begin{bmatrix} \omega_1^2 & -\gamma_c\alpha_1\omega_1^2 \\ -\gamma_c\alpha_1\omega_1^2 & \omega_c^2 - \gamma_c d\gamma_c \end{bmatrix}}_{K=K^T} z = 0. \quad (6)$$

In order to show stability of the closed loop (6), E and K need to be positive definite. While $E > 0$ is given due to the NI property of plant and controller, for K to be positive definite, (6) can be rewritten into an LMI in the variables γ_c and ω_c^2 using the Schur complement as

$$\begin{bmatrix} \omega_1^2 & -\gamma_c\alpha_1\omega_1^2 & 0 \\ -\gamma_c\alpha_1\omega_1^2 & \omega_c^2 & \gamma_c \\ 0 & \gamma_c & d^{-1} \end{bmatrix} > 0. \quad (7)$$

Using the constraint (7), a feasible set of controller parameters can be calculated to initialize the controller design Boyd et al. (1994). This is demonstrated in the next subsection.

Controller Design

An effective approach to controller design is to perform a pole-placement optimization Ruppert and Moheimani (2016). This method places the real part of the closed-loop poles $\Re(p_{i,cl})$ at a desired location $p_{i,d} = \frac{\omega_1}{-2Q_1^*}$ by selecting a desired effective Q factor Q_1^* and finding a solution to the optimization problem

$$\begin{aligned} \min_{\gamma_c, \zeta_c, \omega_c} \quad & J(\gamma_c, \zeta_c, \omega_c) \\ \text{s.t.} \quad & \begin{bmatrix} \omega^2 & -\alpha\omega_1^2\gamma_c & 0 \\ -\gamma_c\alpha\omega_1^2 & \omega_c^2 & \gamma_c \\ 0 & \gamma_c & d^{-1} \end{bmatrix} > 0. \end{aligned} \quad (8)$$

Here, the cost function

$$\begin{aligned} J(\gamma_c, \zeta_c, \omega_c) = & \sum_{i=1}^4 (p_{i,d} - \Re(p_{i,cl}))^2 \\ & + \sum_{i=3}^4 |p_{i,cl} - p_{i-2,cl}|^2 \end{aligned} \quad (9)$$

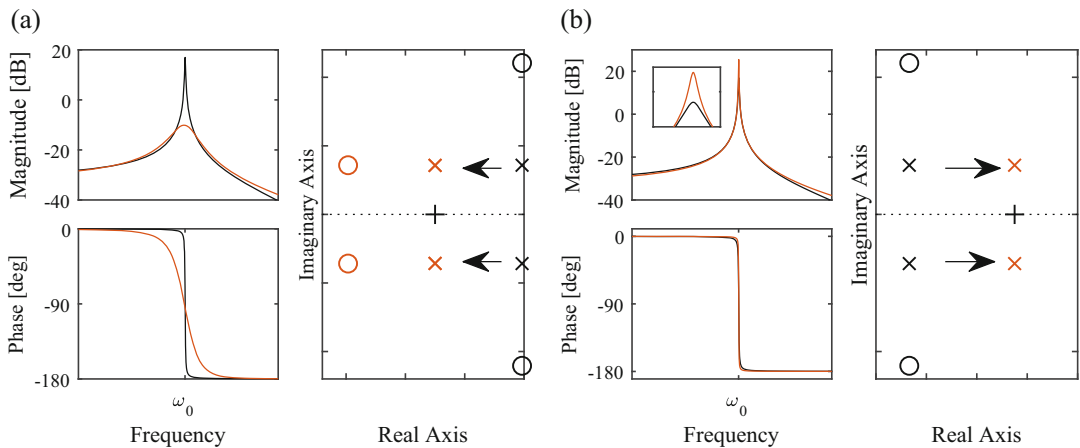
penalizes the difference between the desired and actual pole locations and between poles that correspond to the same open-loop poles. Notice that the objective function is nonlinear in the optimization variables γ_c , ζ_c , and ω_c^2 , leading to a non-convex optimization problem. However in practice, initialization of the optimization problem with a feasible set calculated from (7) results in good convergence.

Example 2 Figure 2 demonstrates the effectiveness of this approach for reducing and increasing the Q factor of the first mode. In Fig. 2a, the Q factor was reduced from 268 to 10 by placing the closed-loop poles deeper into the left-half

plane, and in Fig. 2b the procedure shifts the closed-loop poles closer to the imaginary axis to yield a Q factor of 720. Increasing the Q factor may be desirable in applications where the cantilever is naturally heavily damped, for example, when the AFM cantilever is used for imaging in liquid.

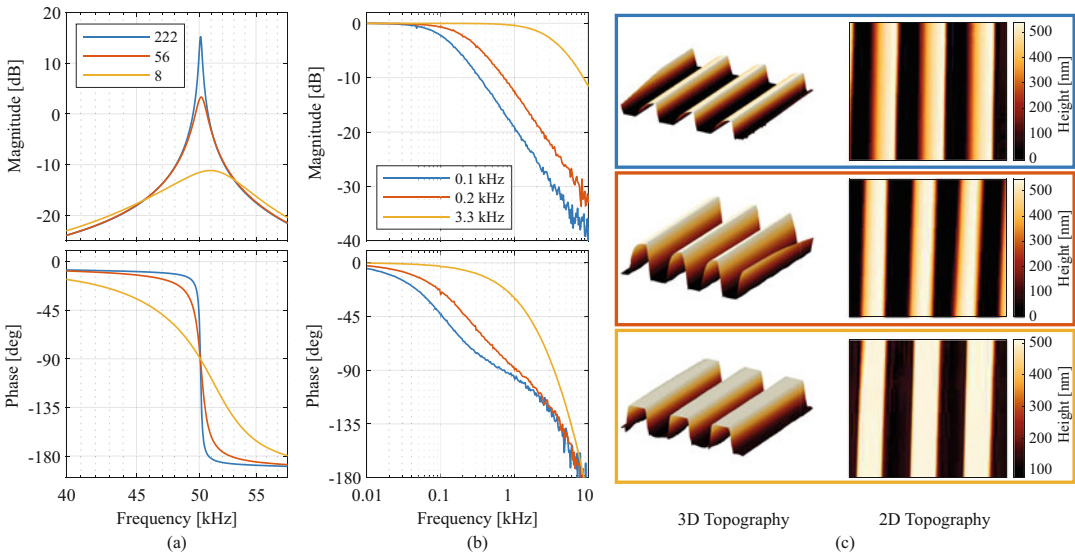
Application to Atomic Force Microscopy

In order to demonstrate the improvement in cantilever tracking bandwidth, we report a high-speed tapping-mode AFM experiment in constant-height mode. In this mode, sample features entirely appear in the cantilever amplitude image and any imaging artifacts are due to insufficient cantilever bandwidth. Using the model-based Q controller presented in the previous section, the Q factor of the fundamental resonance of the piezoelectric cantilever is reduced to as low as $Q_1^* = 8$, resulting in a measured tracking bandwidth of 3.3 kHz, as shown in Fig. 3a, b. With the native Q factor of $Q = 222$, this is more than an order of magnitude faster than the native tracking bandwidth of approximately 100 Hz. When imaging an AFM calibration grating (NT-MDT TGZ3), the increase in bandwidth manifests itself



Dynamics and Control of Active Microcantilevers, Fig. 2 (a) Q factor reduction and (b) Q factor enhancement: Bode plot of the open-loop (—) and closed-loop

system (—) and pole-zero map of desired pole location (+), open-loop (x,o), and closed-loop (x,o) system



Dynamics and Control of Active Microcantilevers, Fig. 3 (a) Frequency response of the DMAPS cantilever in open loop (blue) and for various closed loops to reduce the Q factor (stated in the legend). (b) Tracking bandwidths of the DMAPS cantilever determined via drive

amplitude modulation for the frequency responses shown in (a) and color coded accordingly. (c) Resulting constant-height AFM images in open loop and for various closed loops, color coded accordingly

as better tracking of the rectangular surface features, as is visible in Fig. 3c. The controller was implemented on a Anadigm AN221E04 Field-Programmable Analog Array interfaced with a NT-MDT NTEGRA AFM which was operated at a scanning speed of 1 mm/s.

Summary and Future Directions

When a cantilever with integrated actuation is used for tapping-mode atomic force microscopy (AFM), the negative imaginary nature of its transfer function model allows robust resonant controllers to be used to regulate the quality (Q) factor of the fundamental resonance mode. This is of advantage, as it allows an increase in cantilever tracking bandwidth and therefore imaging speed. With the continuing development of AFM techniques, which make use of the excitation and detection of not only one but multiple eigenmodes of the cantilever, the ability to control these modes and their responses to excitation is believed to be the key to unraveling the true potential of these multifrequency AFM methods.

Cross-References

- [Control for Precision Mechatronics](#)
- [Control Systems for Nanopositioning](#)

Recommended Reading

For more details on the modeling and control of vibrations using piezoelectric transducers, we point the interested reader to the book Moheimani and Fleming (2006).

Bibliography

- Bhushan B (2010) Springer handbook of nanotechnology. Springer, Berlin/Heidelberg
- Binnig G, Quate CF, Gerber C (1986) Atomic force microscope. *Phys Rev Lett* 56:930–933
- Boyd S, Feron E, Ghaoui LE, Balakrishnan V (1994) Linear matrix inequalities in system and control theory. Society for Industrial and Applied Mathematics, Philadelphia
- Giessibl FJ (1995) Atomic resolution of the silicon (111)-(7 × 7) surface by atomic force microscopy. *Science* 267(5194):68–71

- Lanzon A, Petersen I (2008) Stability robustness of a feedback interconnection of systems with negative imaginary frequency response. *IEEE Trans Autom Control* 53(4):1042–1046
- Moheimani SOR, Fleming AJ (2006) Piezoelectric transducers for vibration control and damping. Springer, London
- Petersen I, Lanzon A (2010) Feedback control of negative-imaginary systems. *IEEE Control Syst Mag* 30(5): 54–72
- Rangelow IW, Ivanov T, Ahmad A, Kaestner M, Lenk C, Bozchalooi IS, Xia F, Youcef-Toumi K, Holz M, Reum A (2017) Review article: Active scanning probes: a versatile toolkit for fast imaging and emerging nanofabrication. *J Vac Sci Technol B* 35(6):06G101
- Ruppert MG, Moheimani SOR (2013) A novel self-sensing technique for tapping-mode atomic force microscopy. *Rev Sci Instrum* 84(12):125006
- Ruppert MG, Moheimani SOR (2016) Multimode Q control in tapping-mode AFM: enabling imaging on higher flexural eigenmodes. *IEEE Trans Control Syst Technol* 24(4):1149–1159
- Ruppert MG, Yong YK (2017) Note: guaranteed collocated multimode control of an atomic force microscope cantilever using on-chip piezoelectric actuation and sensing. *Rev Sci Instrum* 88(8):086109

Economic Model Predictive Control

David Angeli

Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Dipartimento di Ingegneria dell'Informazione, University of Florence, Florence, Italy

Abstract

Economic model predictive control (EMPC) is a variant of model predictive control aimed at maximization of system's profitability. It allows one to explicitly deal with hard and average constraints on system's input and output variables as well as with nonlinearity of dynamics. We provide basic definitions and concepts of the approach and highlight some promising research directions.

Keywords

Constrained systems · Dynamic programming · Profit maximization

Introduction

Most control tasks involve some kind of economic optimization. In classical linear quadratic (LQ) control, for example, this is cast

as a trade-off between control effort and tracking performance. The designer is allowed to settle such a trade-off by suitably tuning weighting parameters of an otherwise automatic design procedure.

When the primary goal of a control system is profitability rather than tracking performance, a suboptimal approach has often been devised, namely, a hierarchical separation is enforced between the economic optimization layer and the dynamic real-time control layer.

In practice, while set points are computed by optimizing economic revenue among all equilibria fulfilling the prescribed constraints, the task of the real-time control layer is simply to drive (basically as fast as possible) the system's state to the desired set-point value.

Optimal control or LQ control may be used to achieve the latter task, possibly in conjunction with model predictive control (MPC), but the actual economics of the plant are normally neglected at this stage.

The main benefits of this approach are twofold:

1. Reduced computational complexity with respect to infinite-horizon dynamical programming
2. Stability robustness in the face of uncertainty, normally achieved by using some form of robust control in the real-time control layer

The hierarchical approach, however, is suboptimal in two respects:

1. First of all, given nonlinearity of the plant's dynamics and/or nonconvexity of the functions characterizing the economic revenue, there is no reason why the most profitable regime should be an equilibrium.
2. Even when systems are most profitably operated at equilibrium, transient costs are totally disregarded by the hierarchical approach, and this may be undesirable if the time constants of the plant are close enough to the time scales at which set point's variations occur.

Economic model predictive control seeks to remove these limitations by directly using the economic revenue in the stage cost and by the formulation of an associated dynamic optimization problem to be solved online in a receding horizon manner. It was originally developed by Rawlings and co-workers, in the context of linear control systems subject to convex constraints as an effective technique to deal with infeasible set points (Rawlings et al. 2008) (in contrast to the classical approach of redesigning a suitable quadratic cost that achieves its minimum at the closest feasible equilibrium). Preserving the original cost has the advantage of slowing down convergence to such an equilibrium when the transient evolution occurs in a region where the stage cost is better than at steady state. Stability and convergence issues are at first analyzed, thanks to convexity and for the special case of linear systems only. Subsequently Diehl introduced the notion of rotated cost (see Diehl et al. 2011) that allowed a Lyapunov interpretation of stability criteria and paved the way for the extension to general dissipative nonlinear systems (Angeli et al. 2012).

Economic MPC Formulation

In order to describe the most common versions of economic MPC, assume that a discrete-time finite-dimensional model of state evolution is available for the system to be controlled:

$$x^+ = f(x, u) \quad (1)$$

where $x \in X \subset \mathbb{R}^n$ is the state variable, $u \in U \subset \mathbb{R}^m$ is the control input, and $f: X \times U \rightarrow X$ is a continuous map which computes the next state value, given the current one and the value of the input. We also assume that $\mathbb{Z} \subset X \times U$ is a compact set which defines the (possibly coupled) state/input constraints that need to hold pointwise in time:

$$(x(t), u(t)) \in \mathbb{Z} \quad \forall t \in \mathbb{N}. \quad (2)$$

In order to introduce a measure of economic performance, to each feasible state/input pair $(x, u) \in \mathbb{Z}$, we associate the instantaneous net cost of operating the plant at that state when feeding the specified control input:

$$\ell(x, u) : \mathbb{Z} \rightarrow \mathbb{R}. \quad (3)$$

The function ℓ (which we assume to be continuous) is normally referred to as stage cost and together with actuation and/or inflow costs should also take into account the profits associated to possible output/outflows of the system. Let (x^*, u^*) denote the best equilibrium/control input pair associated to (3) and (2), namely,

$$\begin{aligned} \ell(x^*, u^*) &= \min_{x, u} \ell(x, u) \\ &\text{subject to} \\ &(x, u) \in \mathbb{Z} \\ &x = f(x, u) \end{aligned} \quad (4)$$

Notice that, unlike in tracking MPC, it is not assumed here that

$$\ell(x^*, u^*) \leq \ell(x, u) \quad \forall (x, u) \in \mathbb{Z}. \quad (5)$$

This is, technically speaking, the main point of departure between economic MPC and tracking MPC.

As there is no natural termination time to operation of a system, our goal would be to optimize the infinite-horizon cost functional:

$$\sum_{t \in \mathbb{N}} \ell(x(t), u(t)) \quad (6)$$

possibly in an average sense (or by introducing some discounting factor to avoid infinite costs) and subject to the dynamic/operational constraints (1) and (2).

To make the problem computationally more tractable and yet retain some of the desirable economic benefits of dynamic programming, (6) is truncated to the following cost functional:

$$J(\mathbf{z}, \mathbf{v}) = \sum_{k=0}^{N-1} \ell(z(k), v(k)) + V_f(z(N)) \quad (7)$$

where $\mathbf{z} = [z(0), z(1), \dots, z(N)] \in X^{N+1}$, $\mathbf{v} = [v(0), v(1), \dots, v(N-1)] \in U^N$ and $V_f : X \rightarrow \mathbb{R}$ is a terminal weighting function whose properties will be specified later.

The virtual state/control pair $(\mathbf{z}^*, \mathbf{v}^*)$ at time t is the solution (which for the sake of simplicity we assume to be unique) of the following optimization problem:

$$\begin{aligned} & V(x(t)) = \min_{\mathbf{z}, \mathbf{v}} J(\mathbf{z}, \mathbf{v}) \\ & \text{subject to} \\ & z(k+1) = f(z(k), v(k)) \\ & (z(k), v(k)) \in \mathbb{Z} \\ & \text{for } k \in \{0, 1, \dots, N-1\} \\ & z(0) = x(t), z(N) \in \mathbb{X}_f. \end{aligned} \quad (8)$$

Notice that $z(0)$ is initialized at the value of the current state $x(t)$. Thanks to this fact, $\mathbf{z}^*$ and $\mathbf{v}^*$ may be seen as functions of the current state $x(t)$. At the same time, $z(N)$ is constrained to belong to the compact set $\mathbb{X}_f \subset X$ whose properties will be detailed in the next paragraph.

As customary in model predictive control, a state-feedback law is defined by applying the first virtual control to the plant, that is, by letting $u(t) = v^*(0)$ and restating, at the subsequent time instant, the same optimization problem from initial state $x(t+1)$ which, in the case of exact match between plant and model, can be computed as $f(x(t), u(t))$.

In the next paragraph, we provide details on how to design the “terminal ingredients” (namely, V_f and $\mathbb{X}_f$) in order to endow the basic algorithm (8) with important features such as recursive feasibility and a certain degree of average performance and/or stability.

Hereby it is worth pointing out how, in the context of economic MPC, it makes sense to treat, together with pointwise-in-time constraints, asymptotic average constraints on specified input/output variables. In tracking applications, where the control algorithm guarantees asymptotic convergence of the state to a feasible set point, the average asymptotic value of all input/output variables necessarily matches that of the corresponding equilibrium/control input pair. In economic MPC, the asymptotic regime resulting in closed loop may, in general, fail to be an equilibrium; therefore, it might be of interest to impose average constraints on system’s inflows and outflows which are more stringent than those indirectly implied by the fulfillment of (2). To this end, let the system’s output be defined as

$$y(t) = h(x(t), u(t)) \quad (9)$$

with $h(x, u) : \mathbb{Z} \rightarrow \mathbb{R}^p$, a continuous map, and consider the convex compact set $\mathbb{Y}$. We may define the set of asymptotic averages of a bounded signal y as follows:

$$\begin{aligned} \text{Av}[y] &= \left\{ \eta \in \mathbb{R}^p : \exists \{t_n\}_{n=1}^{\infty} : t_n \rightarrow \infty \text{ as } n \rightarrow \infty \right. \\ &\quad \left. \text{and } \eta = \lim_{n \rightarrow \infty} \left(\frac{\sum_{k=0}^{t_n-1} y(k)}{t_n} \right) \right\} \end{aligned}$$

Notice that for converging signals, or even for periodic ones, $\text{Av}[y]$ always is a singleton but may fail to be such for certain oscillatory regimes. An asymptotic average constraint can be expressed as follows:

$$\text{Av}[y] \subseteq \mathbb{Y} \quad (10)$$

where y is the output signal as defined in (9).

Basic Theory

The main theoretical results in support of the approach discussed in the previous paragraph are discussed below. Three fundamental aspects are treated:

- Recursive feasibility and constraint satisfaction
- Asymptotic performance
- Stability and convergence

Feasibility and Constraints

The departing point of most model predictive control techniques is to ensure recursive feasibility, namely, the fact that feasibility of the problem (8) at time 0 implies feasibility at all subsequent times, provided there is no mismatch between the true plant and its model (1). This is normally achieved by making use of a suitable notion of control invariant set which is used as a terminal constraint in (8). Economic model predictive control is not different in this respect, and either one of the following set of assumptions is sufficient to ensure recursive feasibility:

1. Assumption 1: Terminal constraint

$$\mathbb{X}_f = \{x^*\} \quad V_f = 0$$

2. Assumption 2: Terminal penalty function

There exists a continuous map $\kappa : \mathbb{X}_f \rightarrow U$ such that

$$\begin{aligned} (x, \mathbb{K}(x)) &\in \mathbb{Z} & \forall x \in \mathbb{X}_f \\ f(x, \mathbb{K}(x)) &\in \mathbb{X}_f & \forall x \in \mathbb{X}_f \end{aligned}$$

The following holds:

Theorem 1 *Let $x(0)$ be a feasible state for (8), and assume that either Assumption 1 or 2 holds. Then, the closed-loop trajectory $x(t)$ resulting from receding horizon implementation of the feedback $u(t) = v^*(0)$ is well defined for all $t \in \mathbb{N}$ (i.e., $x(t)$ is a feasible initial state of (8) for all $t \in \mathbb{N}$), and the resulting closed-loop variables $(x(t), u(t))$ fulfill the constraints in (2).*

The proof of this Theorem can be found in Angeli et al. (2012) and Amrit et al. (2011), for instance. When constraints on asymptotic averages are of interest, the optimization problem (8) can be augmented by the following constraints:

$$\sum_{k=0}^{N-1} h(z(k), v(k)) \in \mathbb{Y}_t \quad (11)$$

provided $\mathbb{Y}_t$ is recursively defined as

$$\mathbb{Y}_{t+1} = \mathbb{Y}_t \oplus \mathbb{Y} \oplus \{-h(x(t), u(t))\} \quad (12)$$

where $\oplus$ denotes Pontryagin's set sum. ($A \oplus B := \{c : \exists a \in A, \exists b \in B : c = a + b\}$) The sequence is initialized as $\mathbb{Y}_0 = N\mathbb{Y} \oplus \mathbb{Y}_{00}$ where $\mathbb{Y}_{00}$ is an arbitrary compact set in $\mathbb{R}^p$ containing 0 in its interior. The following result can be proved.

Theorem 2 *Consider the optimization problem (8) with additional constraints (11), and assume that $x(0)$ is a feasible initial state. Then, provided a terminal equality constraint is adopted, the closed-loop solution $x(t)$ is well defined and feasible for all $t \in \mathbb{N}$, and the resulting closed-loop variable $y(t) = h(x(t), u(t))$ fulfills the constraint (10).*

Extending average constraints to the case of economic MPC with terminal penalty function is possible but outside the scope of this brief tutorial. It is worth mentioning that the set $\mathbb{Y}_{00}$ plays the role of an initial allowance that is shrunk or expanded as a result of how close are closed-loop output signals to the prescribed region. In particular, $\mathbb{Y}_{00}$ can be selected a posteriori (after computation of the optimal trajectory) just for $t = 0$, so that the feasibility region of the algorithm is not affected by the introduction of average asymptotic constraints.

Asymptotic Average Performance

Since economic MPC does not necessarily lead to converging solutions, it is important to have bounds which estimate the asymptotic average performance of the closed-loop plant. To this end, the following dissipation inequality is needed for the approach with terminal penalty function:

$$V_f(f(x, \mathbb{K}(x))) \leq V_f(x) - \ell(x, \mathbb{K}(x)) + \ell(x^*, u^*) \quad (13)$$

which shall hold for all $x \in \mathbb{X}_f$. We are now ready to state the main bound on the asymptotic performance:

Theorem 3 Let $x(0)$ be a feasible state for (8), and assume that either Assumption 1 or Assumption 2 together with (13) holds. Then, the closed-loop trajectory $x(t)$ resulting from receding horizon implementation of the feedback $u(t) = v^*(0)$ is well defined for all $t \in \mathbb{N}$ and fulfills

$$\limsup_{T \rightarrow +\infty} \frac{\sum_{t=0}^{T-1} \ell(x(t), u(t))}{T} \leq \ell(x^*, u^*). \quad (14)$$

The proof of this fact can be found in Angeli et al. (2012) and Amrit et al. (2011). When periodic solutions are known to outperform, in an average sense, the best equilibrium/control pair, one may replace terminal equality constraints by periodic terminal constraints (see Angeli et al. 2012). This leads to an asymptotic performance at least as good as that of the solution adopted as a terminal constraint. See Angeli et al. (2016) for extensions to the case of time-varying costs.

Stability and Convergence

It is well known that the cost-to-go $V(x)$ as defined in (8) is a natural candidate Lyapunov function for the case of tracking MPC. In fact, the following estimate holds along solutions of the closed-loop system:

$$V(x(t+1)) \leq V(x(t)) - \ell(x(t), u(t)) + \ell(x^*, u^*). \quad (15)$$

This shows, thanks to inequality (5), that $V(x(t))$ is nonincreasing. Owing to this, stability and convergence can be easily achieved under mild additional technical assumptions. While property (15) holds for economic MPC, both in the case of terminal equality constraint and terminal penalty function, it is no longer true that (5) holds. As a matter of fact, x^* might even fail to be an equilibrium of the closed-loop system, and hence, convergence and stability cannot be expected in general.

Intuitively, however, when the most profitable operating regime is an equilibrium, the average performance bound provided by Theorem 3 seems to indicate that some form of stability or convergence to x^* could be expected. This is

true under an additional dissipativity assumption which is closely related to the property of optimal operation at steady state.

Definition 1 A system is strictly dissipative with respect to the supply function $s(x, u)$ if there exists a continuous function $\lambda : X \rightarrow \mathbb{R}$ and $\rho : X \rightarrow \mathbb{R}$ positive definite with respect to x^* such that for all x and u in $X \times U$, it holds:

$$\lambda(f(x, u)) \leq \lambda(x) + s(x, u) - \rho(x). \quad (16)$$

The next result highlights the connection between dissipativity of the open-loop system and stability of closed-loop economic MPC.

Theorem 4 Assume that either Assumption 1 or Assumption 2 together with (13) holds. Let the system (1) be strictly dissipative with respect to the supply function $s(x, u) = \ell(x, u) - \ell(x^*, u^*)$ as from Definition 1 and assume there exists a neighborhood of feasible initial states containing x^* in its interior. Then provided V is continuous at x^* , x^* is an asymptotically stable equilibrium with basin of attraction equal to the set of feasible initial states.

See Angeli et al. (2012) and Amrit et al. (2011) for proofs and discussions. Convergence results are also possible for the case of economic MPC subject to average constraints. Details can be found in Müller et al. (2014).

Hereby it is worth mentioning that finding a function satisfying (16) (should one exist) is in general a hard task (especially for nonlinear systems and/or nonconvex stage costs); it is akin to the problem of finding a Lyapunov function, and therefore general construction methods do not exist. Let us emphasize, however, that while existence of a storage function λ is a sufficient condition to ensure convergence of closed-loop economic MPC, formulation and resolution of the optimization problem (8) can be performed irrespectively of any explicit knowledge of such function. Also, we point out that existence of λ as in Definition 1 and Theorem 4 is only possible if the optimal infinite-horizon regime of operation for the system is an equilibrium.

Summary and Future Directions

Economic model predictive control is a fairly recent and active area of research with great potential in those engineering applications where economic profitability is crucial rather than tracking performance.

The technical literature is rapidly growing in application areas such as chemical engineering (see Heidarinejad 2012) or power systems engineering (see Hovgaard et al. 2010; Müller et al. 2014) where system's output is in fact physical outflows which can be stored with relative ease.

We only dealt with the basic theoretical developments and would like to provide pointers to interesting recent and forthcoming developments in this field:

- Generalized terminal constraints: possibility of enlarging the set of feasible initial states by using arbitrary equilibria as terminal constraints, possibly to be updated on line in order to improve asymptotic performance (see Fagiano and Teel 2012; Müller et al. 2013).
- Economic MPC without terminal constraints: removing the need for terminal constraints by taking a sufficiently long control horizon is an interesting possibility offered by standard tracking MPC. This is also possible for economic MPC at least under suitable technical assumptions as investigated in Grüne (2012, 2013). Another related line of investigation has recently focused on the link between Turnpike Properties and design of closed-loop Economic MPC algorithms (Faulwasser and Bonvin 2015)
- Generalized optimal regimes of operation (beyond steady state) are considered in Dong and Angeli (2018a)
- The basic developments presented in the previous paragraph only deal with systems unaffected by uncertainty. This is a severe limitation of initial approaches and, as for the case of tracking MPC, a great deal of research in this area is currently being developed, see Bayer et al. (2014) and Dong and Angeli (2018b). In particular, both deterministic and stochastic uncertainties are of interest, and tube-based or scenario based solutions.

Cross-References

- [Model Predictive Control in Practice](#)
- [Nominal Model-Predictive Control](#)
- [Stochastic Model Predictive Control](#)

Recommended Reading

Papers Amrit et al. (2011), Angeli et al. (2011, 2012), Diehl et al. (2011), Müller et al. (2014, 2015) and Rawlings et al. (2008) set out the basic technical tools for performance and stability analysis of EMPC. To readers interested in the general theme of optimization of system's economic performance and its relationship with classical turnpike theory in economics, please refer to Rawlings and Amrit (2009). Potential applications of EMPC are described in Hovgaard et al. (2010), Heidarinejad (2012), and Ma et al. (2011) while Rawlings et al. (2018) describes one of the first large-scale industrial deployment of the technique on Stanford University's new heating/chilling station, controlling the HVAC of 155 campus buildings. Rawlings et al. (2012) and Angeli and Müller (2018) are (respectively) a concise and a recent survey on the topic. Fagiano and Teel (2012) and Grüne (2012, 2013) deal with the issue of relaxation or elimination of terminal constraints, while Müller et al. (2013) explore the possibility of adaptive terminal costs and generalized equality constraints. For readers interested on a detailed technical monograph, see Faulwasser et al. (2018), while a recent book, with special emphasis on chemical process applications is Ellis et al. (2017). Finally, Chapter 2 of Rawlings et al. (2017), contains a section devoted to Economic MPC in the context of classical MPC.

Bibliography

- Amrit R, Rawlings JB, Angeli D (2011) Economic optimization using model predictive control with a terminal cost. *Annu Rev Control* 35:178–186
- Angeli D, Müller M (2018) Economic model predictive control: some design tools and analysis techniques. In: Rakovic S, Levine WS (eds) *Handbook of model predictive control*. Birkhauser, Cham, pp 145–167

- Angeli D, Amrit R, Rawlings JB (2011) Enforcing convergence in nonlinear economic MPC. In: Paper presented at the 50th IEEE conference on decision and control and european control conference (CDC-ECC), Orlando, 12–15 Dec 2011
- Angeli D, Amrit R, Rawlings JB (2012) On average performance and stability of economic model predictive control. *IEEE Trans Autom Control* 57:1615–1626
- Angeli D, Casavola A, Tedesco F (2016) Theoretical advances on economic model predictive control with time-varying costs. *Ann Rev Control* 41:218–224
- Bayer FA, Müller MA, Allgöwer F (2014) Tube-based robust economic model predictive control. *J Process Control* 24:1237–1246
- Diehl M, Amrit R, Rawlings JB (2011) A Lyapunov function for economic optimizing model predictive control. *IEEE Trans Autom Control* 56:703–707
- Dong Z, Angeli D (2018a) Analysis of economic model predictive control with terminal penalty functions on generalized optimal regimes of operation. *Int J Robust Nonlinear Control* 28:4790–4815
- Dong Z, Angeli D (2018b) Tube-based robust economic model predictive control on dissipative systems with generalized optimal regimes of operation. In: Paper presented at the 2018 IEEE conference on decision and control, Miami
- Ellis M, Liu J, Christofides PD (2017) Economic model predictive control: theory, formulations and chemical process applications. In: *Advances in industrial control*. Springer, Amsterdam
- Fagiano L, Teel A (2012) Model predictive control with generalized terminal state constraint. In: Paper presented at the IFAC conference on nonlinear model predictive control, Noordwijkerhout, 23–27 Aug 2012
- Faulwasser T, Bonvin D (2015) On the design of economic NMPC based on approximate turnpike properties. In: Paper presented at the 54th IEEE conference on decision and control, Osaka, Dec 2015
- Faulwasser T, Grüne L, Müller M (2018) Economic nonlinear model predictive control. *Found Trends Syst Control* 5:1–98
- Grüne L (2012) Economic MPC and the role of exponential turnpike properties. *Oberwolfach Rep* 12:678–681
- Grüne L (2013) Economic receding horizon control without terminal constraints. *Automatica* 49:725–734
- Heidarnejad M (2012) Economic model predictive control of nonlinear process systems using Lyapunov techniques. *AIChE J* 58:855–870
- Hovgaard TG, Edlund K, Bagterp Jorgensen J (2010) The potential of economic MPC for power management. In: Paper presented at the 49th IEEE conference on decision and control, Atlanta, 15–17 Dec 2010
- Ma J, Joe Qin S, Li B et al (2011) Economic model predictive control for building energy systems. In: Paper presented at the IEEE innovative smart grid technology conference, Anaheim, 17–19 Jan 2011
- Müller MA, Angeli D, Allgöwer F (2013) Economic model predictive control with self-tuning terminal weight. *Eur J Control* 19:408–416
- Müller MA, Angeli D, Allgöwer F, Amrit R, Rawlings J (2014) Convergence in economic model predictive control with average constraints. *Automatica* 50:3100–3111
- Müller MA, Angeli D, Allgöwer F (2015) On necessity and robustness of dissipativity in economic model predictive control. *IEEE Trans Autom Control* 60:1671–1676
- Rawlings JB, Amrit R (2009) Optimizing process economic performance using model predictive control. In: Magni L, Raimondo DM, Allgöwer F (eds) *Nonlinear model predictive control. Lecture notes in control and information sciences*, vol 384. Springer, Berlin, pp 119–138
- Rawlings JB, Bonne D, Jorgensen JB et al (2008) Unreachable setpoints in model predictive control. *IEEE Trans Autom Control* 53:2209–2215
- Rawlings JB, Angeli D, Bates CN (2012) Fundamentals of economic model predictive control. In: Paper presented at the IEEE 51st annual conference on decision and control (CDC), Maui, 10–13 Dec 2012
- Rawlings JB, Mayne DQ, Diehl MM (2017) *Model predictive control: theory, design, and computation*. Nob Hill Publishing, Madison
- Rawlings JB, Patel NR, Risbeck MJ et al (2018) Economic MPC and real-time decision making with application to large-scale HVAC energy systems. *Comput Chem* 114:89–98

EKF

► Extended Kalman Filters

Electric Energy Transfer and Control via Power Electronics

Fred Wang

University of Tennessee, Knoxville, TN, USA

Abstract

Power electronics and their applications for electric energy transfer and control are introduced. The fundamentals of the power electronics are presented, including the commonly used semiconductor devices and power converter circuits. Different types of power electronics controllers for electric power generation, transmission and distribution, and consumption are described. The advantages of power electronics over

traditional electromechanical or electromagnetic controllers are explained. The future directions for power electronics application in electric power systems are discussed.

Keywords

Power electronics · Electric energy transfer · Electric energy control

Introduction

Modern society runs on electricity or electric energy. The electric energy generally must be transferred before consumption since the energy sources, such as thermal power plants, hydro dams, and wind farms, are often some distances away from the loads. In addition, electric energy needs to be controlled as well since the energy transfer and use often require electricity in a form different from the raw form generated at the source. Examples are the voltage magnitude and frequency. For long-distance transmission, the voltage needs to be stepped up at the sending end to reduce the energy loss along the lines and then stepped down at the receiving end for users; for many modern consumer devices, DC voltage is needed and obtained through transforming the 50 or 60 Hz utility power. Note that electric energy transfer and control is often used interchangeably with the electric power transfer and control. This is because the modern electric power systems have very limited energy storage, and the energy generated must be consumed at the same time. Even with its growing use, the energy storage is mainly connected to the system to either act as load or source, and its power transfer is part of the system power transfer.

Since the beginning of the electricity era, electric energy transfer and control technologies have been an essential part of electric power systems. Many types of equipment were invented and applied for these purposes. The commonly used equipment includes electric transmission and distribution lines, generators, transformers, switchgears, inductors or reactors, and capacitor banks. The traditional equipment has limited

control capability. Many cannot be controlled at all or can only be connected or disconnected with mechanical switches, others with limited range, such as transformers with tap changers. Even with fully controllable equipment such as generators, the control dynamics is relatively slow due to the electromechanical or electro magnetic nature of the controller.

Power electronics are based on semiconductor devices. These devices are derivatives from transistors and diodes used in microelectronic circuits with the additional large power handling capability. Due to their electronic nature, power electronic devices are much more flexible and faster than their electromechanical or electromagnetic counterparts for electric energy transfer and control. Since the advent of power electronics in the 1950s, they have steadily gained ground in power systems applications. Today, power electronics controllers are an important part of equipment for electric energy transfer and control. Their roles are growing rapidly with the continuous improvement of the power electronics technologies.

Fundamentals of Power Electronics

Different from semiconductor devices in microelectronics, the power electronic devices only act as switches for desired control functions, such that they incur minimum losses when they are either on (closed) or off (open). As a result, the power electronics controllers are basically switching circuits. The semiconductor switches are therefore the most important elements of the power electronics controllers. Since the 1950s, many different types of power semiconductor switches have been developed and can be selected based on the applications.

The performance of the power semiconductor devices is mainly characterized by their voltage and current ratings, conduction or on-state loss, as well as the switching speed (or switching frequency capability) and associated switching loss. Main types of power semiconductor devices are listed with their symbols and state-of-the-

Electric Energy Transfer and Control via Power Electronics, Table 1 Commonly use Si-based power semiconductor devices and their ratings

Types	Symbol	Voltage	Current	Switching frequency
Power diodes		Max 80 kV, typical <10 kV	10 kA	Various
Thyristor		Max 8 kV	4.5 kA	AC line frequency
GTO		Max 10 kV	6.5 kA	<500 Hz
Power MOSFET		Max 4.5 kV, typical <600 V	1.6 kA	10s of kHz to MHz
IGBT		Max 6.5 kV, typical >600 V	2.4 kA	1 kHz to 10s of kHz
IGCT		Max 10 kV, typical >4.5 kV	6.5 kA	<2 kHz

E

art rating and frequency range shown in Table 1 (Rahimo and Klaka 2009).

- Power diode – a two-terminal device with similar characteristics to diodes used in microelectronics but with higher voltage and power ratings.
- Thyristor – also called SCR (silicon-controlled rectifier). Unlike diode, thyristor is a three-terminal device with an additional gate terminal. It can be turned on by a current pulse through gate but can only be turned off when the main current goes to zero with external means. Thyristor has low conduction loss but slow switching speed.
- GTO – stands for gate-turn-off thyristor. GTO can be turned on similarly as a regular thyristor and can also be turned off with a large negative gate current pulse. GTO has been largely replaced by IGBT and IGCT due to its complex gate driving needs and slow switching speed.
- Power BJT – similar to bipolar transistor for microelectronics and requires a sustained gate current to turn on and off. It has been replaced by IGBT and power MOSFET with simpler gate signals and faster switching speed.
- Power MOSFET – similar to metal-oxide semiconductor field effect transistor for microelectronics and can be turned on and off with a gate voltage signal. It is the fastest device available but has relatively

high conduction loss and relatively low voltage/power ratings.

- IGBT – stands for insulated-gate bipolar transistor. Unlike regular BJT, it can be turned on and off with a gate voltage like MOSFET. It has relatively low conduction loss and fast switching speed. IGBT is becoming the workhorse of the power electronics for high-power applications.
- IGCT – stands for integrated gate-commutated thyristor. It is basically a GTO with an integrated gate drive circuit allowing a hard-driven turnoff. It therefore has faster switching speed than regular GTO but slower than IGBT.

Except for diodes, all other devices above can be turned on and/or off through a gate signal, so they are active switches, while diodes are called passive switch.

Note that devices in Table 1 are all based on Si. Since 1950s, Si power semiconductor devices have gone through many generations of development and are approaching material theoretical limitations in terms of blocking voltage, operation temperature, and conduction and switching characteristics. In recent years, semiconductor devices based on wide bandgap (WBG) (energy bandgap significantly higher than 1 eV) materials have progressed rapidly that promises to revolutionize next-generation power electronics converters. Compared with the Si (energy bandgap around 1 eV) devices,

WBG devices feature high breakdown electric field, low specific on-resistance, fast switching speed, and high junction temperature capability. These characteristics are beneficial for efficiency, high power and voltage capability, size, and cost of power electronics converters, which are all important for power electronics applications as energy transfer controllers in power systems.

The WBG devices under rapid development and commercialization include silicon carbide (SiC) and gallium nitride (GaN) devices, with SiC mainly targeting high-voltage high-power (600 V, kilowatts or above) applications and GaN for low-voltage low-power (600 V, kilowatts or below) applications (Millan et al. 2014; Wang et al. 2015; Wang and Zhang 2016).

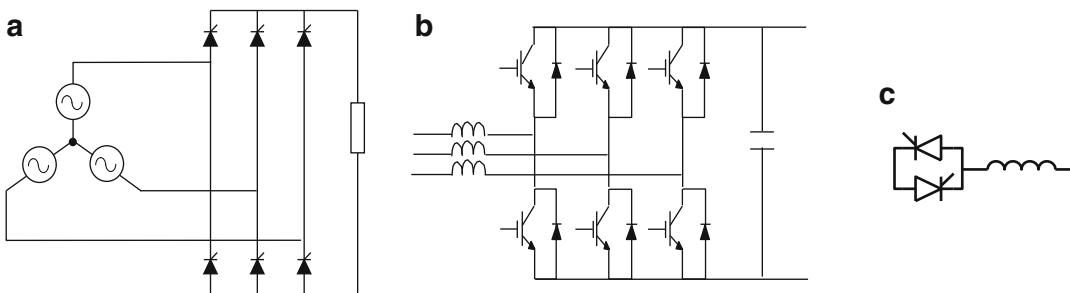
Similar to Si semiconductors, WBG semiconductors can be used to realize various power devices. The more mature SiC-based devices already include diodes (both Schottky diodes and PIN diodes), MOSFETs, IGBTs, and thyristors with the voltage range from 400 V to 22.6 kV. In some cases, SiC junction gate FETs (JFETs) and bipolar junction transistors (BJTs) are also available and used. The low-voltage (from 400 to 1700 V) SiC devices are already commercially available. Among them, the current rating per die approaches 100 A, and with multiple dies in parallel, state-of-the-art SiC power modules on market can deliver hundreds of amperes of current. The high-voltage SiC devices (3.3 kV and above) are generally still in development stages and with small current rating per die; however, their further development and commercialization

are expected in the near future and will help to improve electric energy transfer controllers.

With different types of power semiconductors, many power electronics circuits have been developed. Based on their functions, they can be classified as:

- Rectifier – rectifiers convert AC to DC. Depending on AC sources, rectifiers can be three-phase or single-phase; depending on device types, they can be passive (diode-based), phase-controlled (thyristor-controlled), or actively switched.
- Inverter – inverters convert DC to AC. They again can be three-phase or single-phase. Inverters generally require active switching devices.
- DC-DC converter – also called chopper, DC-DC converter converts one DC voltage level to another. Sometimes it also contains a magnetic isolation. DC-DC converter can have unidirectional or bidirectional power flow and generally requires active switching devices.
- AC-AC converter – directly converts one AC to another, either only the voltage magnitude or both magnitude and frequency. The former can also be called AC switch, and the latter can be called frequency changer. Active devices are needed for these types of converters.

There are a variety of converter topologies for each type of the converters listed above. The most commonly used basic topologies for power system applications are shown in Fig. 1.



Electric Energy Transfer and Control via Power Electronics, Fig. 1 Commonly used basic power electronics converter topologies (only one phase shown for the AC

switch). (a) Thyristor-based rectifier. (b) Voltage source inverter (VSI). (c) Bidirectional AC switch

These basic topologies can be expanded through paralleling or series of devices and/or converters to achieve higher current and voltage ratings. Other variations such as multilevel converters are also popular for high-voltage applications using lower-voltage rating devices.

It should be noted that passive components, i.e., inductors and capacitors, are essential parts of power electronics converters. In fact, power electronics converters transfer or control the electric energy by storing it temporarily in inductors or capacitors while reformat the original voltage or current waveform through switching actions. The other key function of the passives is filtering the harmonics caused by switching.

Power Electronics Controller Types for Energy Transfer and Control

For almost all traditional non-power electronics equipment for electric energy transfer and control, there can be corresponding power electronics-based counterpart, often with better controllability. However, power electronics equipment can be more expensive and therefore only used when it provides better overall performance and cost benefits. In other cases, only power electronics equipment can achieve the required control functions.

The power electronics controllers can be categorized as for energy generation, delivery, and consumption. For generation, the thermal or hydro generators both use synchronous machines with excitation windings on the rotor, which require DC current. A thyristor-based rectifier, called exciter, is generally used for this purpose. Wind turbine generators usually use a back-to-back VSI to interface to the AC grid, and PV solar sources use a DC-DC converter cascaded with a VSI.

Power electronics controllers for transmission and distribution controllers include the so-called flexible AC transmission systems (FACTS) and high-voltage DC transmission (HVDC) (Hingorani and Gyugyi 2000). Some of the more commonly used controllers and their functions and circuit topologies are listed in Table 2 (Wang et al. 2003).

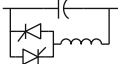
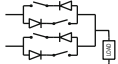
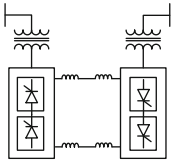
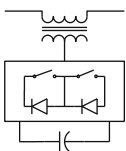
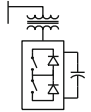
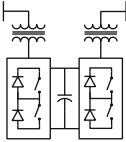
The main power electronics controllers for loads include variable speed motor drives, electronic ballast for florescent lights and power supplies for LED, various power supplies for computer, IT and other electronic loads, and chargers for electric vehicles. The percentage of power electronics controlled loads in power systems has been steadily increasing. Power electronics can generally result in improved performance and efficiency.

Future Directions

Power electronics have progressed steadily since the invention of thyristors in the 1950s. The progress is in all aspects, semiconductor devices, passives, circuits, control, and system integration, leading to converter systems with better performance, higher efficiency, higher power density, higher reliability, and lower cost. Because of these progresses, the power electronics applications in power systems have become more and more widespread. However, in general, power electronics controllers are still not sufficiently cost-effective, reliable, or efficient. Many improvements are needed and expected, especially in the following areas:

- Semiconductor devices – Devices used today are almost exclusively based on silicon. The emerging devices based on wide bandgap materials such as SiC and GaN are expected to revolutionize power electronics with their capabilities of higher voltage, lower loss, faster switching speed, higher temperature, and smaller size.
- Power electronics converters – More cost-effective and reliable converters will be developed as a result of better devices, passive components, and circuit structures. Modular, distributed, and hybrid with non-power electronics approaches are expected to result in overall better benefits.
- Enhanced functions – Power electronics controllers can be designed to have multiple functions in the system. For example, wind and PV solar inverters can provide reactive power to the grid in addition

Electric Energy Transfer and Control via Power Electronics, Table 2 Commonly used power electronics controllers for transmission and distribution

Controller	One-line configuration	System functions	Control principle	Basic PE function
SVC – Static Var Compensator with Thyristor-Controlled Reactor and Capacitor		Stability enhancement Voltage regulation & VAR compensation	VAR control through varying L & C in shunt connection	Controlled bi-directional AC switch
TCSC – Thyristor-Controlled Series Capacitor		Power flow control Stability enhancement Fault current limiting	Power & VAR control through varying C & L in series connection	Controlled bi-directional AC switch
SSTS – Solid State Transfer Switch		Power supply transfer for reliability and power quality	On and off control	Controlled bi-directional AC switch
HVDC (Classic)		System interconnection Power flow control Stability enhancement	Power control through back-to-back converters in shunt connections	Bi-directional AC/DC current source converter
SSSC – Static Series Synchronous compensator		Power flow control Stability enhancement	VAR control through voltage control in series connection	Bi-directional AC/DC voltage source converter
STATCOM – Static Synchronous Compensator		Stability enhancement Voltage regulation & VAR compensation	VAR control through current control in shunt connection	Bi-directional AC/DC voltage source converter
HVDC (Voltage Source)		System interconnection Power flow control Stability enhancement	Power and Var control through back-to-back converters in shunt connections	Bi-directional AC/DC voltage source converter

to transferring real energy. Today, power electronics controllers are mostly locally controlled. With better measurement and communication technologies, they may be controlled over a wide area for supporting the system-level functions. With more system-level functions, power electronics converters also need to be designed to consider grid operating conditions.

- New applications – The new applications for future power system include DC grid based on multiterminal HVDC and energy storage. Critical technologies include cost-effective

and efficient DC transformers and DC circuit breakers. Power electronics will play key roles in these technologies. The emerging WBG device technologies will help enable these new applications.

Cross-References

- [Cascading Network Failure in Power Grid Blackouts](#)

- ▶ [Coordination of Distributed Energy Resources for Provision of Ancillary Services: Architectures and Algorithms](#)
- ▶ [Lyapunov Methods in Power System Stability](#)
- ▶ [Power System Voltage Stability](#)
- ▶ [Small Signal Stability in Electric Power Systems](#)
- ▶ [Time-Scale Separation in Power System Swing Dynamics: Singular Perturbations and Coherency](#)

Bibliography

- Hingorani NG, Gyugyi L (2000) Understanding facts – concepts and technology of flexible AC transmission systems. IEEE Press, New York
- Millan J, Godignon P, Perpina X, Perez-Tomas A, Rebollo J (2014) A survey of wide bandgap power semiconductor devices. *IEEE Trans Power Electron* 29(5): 2155–2163
- Rahimo M, Klaka S (2009) High voltage semiconductor technologies. In: 13th European conference on power electronics and applications, EPE'09. Barcelona, pp 1–10
- Wang F, Zhang Z (2016) Overview of silicon carbide technology: device, converter, system, and application. *CPSS Trans Power Electron Appl* 1(1):13–32
- Wang F, Rosado S, Boroyevich D (2003) Open modular power electronics building blocks for utility power system controller applications. In: IEEE PESC, Acapulco, pp 1792–1797, 15–19 June 2003
- Wang F, Zhang Z, Ericson T, Raju R, Burgos R, Boroyevich D (2015) Advances in power conversion and drives for shipboard systems. *Proc IEEE* 103(12):2285–2311

Emergency Building Control

Veronica A. Adetola
Pacific Northwest National Laboratory,
Richland, WA, USA

Abstract

The number and frequency of emergency events in buildings are increasing. Rapid situational awareness and effective control require greater integration and coordination between diverse building systems. This entry

provides an overview of emergency events and response in buildings and highlights some future research directions.

Keywords

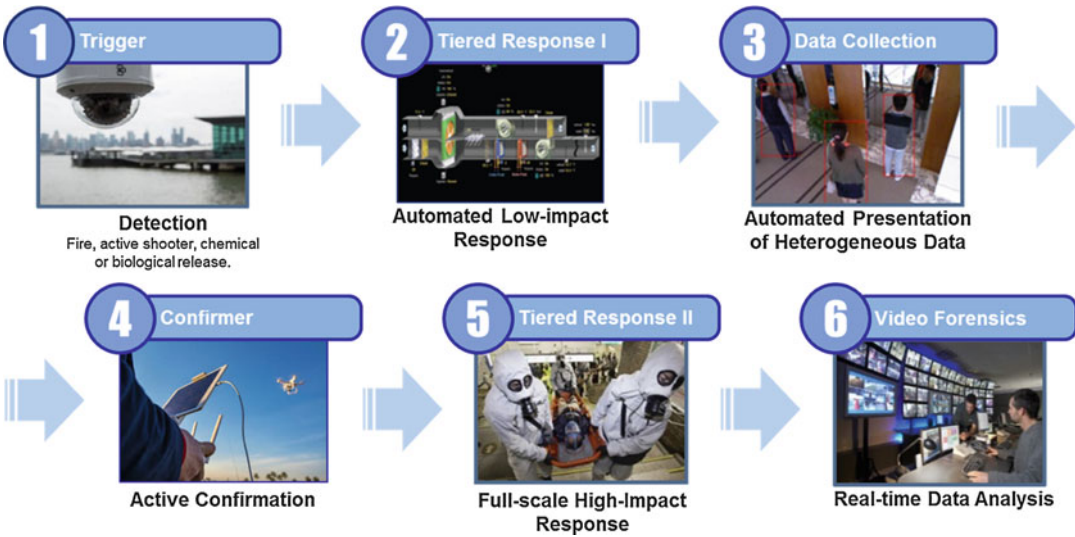
Emergency response · Hazardous contaminants · Cyber security · Power outage · Mitigation · Building management system (BMS)

E

Introduction

There is no universally accepted definition or classification of an emergency situation. For this entry, building emergency is defined as an event or circumstance that can cause death or significant injuries to the building occupants, cause physical damage, or significantly disrupt the building operations. Such events include fire outbreak, hazardous agent release, active shooter incidents, power outages, cyber-attacks, and severe weather. The building sector has made significant strides in deploying safety and response systems for traditional events such as fire. While these systems are heavily regulated, response strategies for the newer and increasing threats such as active shooters, biological/chemical threats, and cyber-attacks are still at a nascent stage with significant gaps and opportunities. An effective response requires a multilayer strategy to allow for threat assessment and to assure the highest possible levels of security and safety.

Figure 1 shows an exemplary multi-tiered response strategy for building emergencies. The decision options depend on the available information and may have a range of impacts on the building and its occupants, from low and localized impact (e.g., redirecting airflows in the case of hazardous agent release) to high impact (e.g., building shutdown and remediation). The automated low-impact response provides immediate action, upon the detection of a threat or occurrence of an emergency event, and mitigates impact. The low-impact response is usually followed by additional data collection and threat confirmation. If the threat is



Emergency Building Control, Fig. 1 Tiered response reduces impact while threat is assessed. Adapted from (Wagner et al. 2018)

verified, a high-impact response is implemented. The following sections discuss some of the emergency events in buildings and automated low-impact response strategies.

Physical Attack: Hazardous Agents

The number of hazardous (i.e., chemical or biological) agent attacks in indoor environments is growing. Examples include the 2001 anthrax attacks in the United States and the recent nerve agent attacks in Malaysia and Salisbury, England. Typical commercial buildings have no system to actively mitigate these threats; they rely primarily on restricted access for security and onsite video cameras for post-event forensics (Wagner et al. 2018). Commercially available sensors exist for monitoring chemical and biological threats (e.g., *TACBIO 2*, *FLIR IBAC 2*, *Polaron F10*), and there are ongoing efforts such as the Department of Homeland Security's (DHS) SenseNet program to develop reliable low-cost sensors for real-time detection of biological threats in indoor environments. (<https://www.dhs.gov/science-and-technology/detection-and-diagnostics>) The advancements in smart connected buildings, along with these sensor technologies, can be leveraged to

provide an automated initial response to the release of a hazardous agent and mitigate impact prior to the arrival of first responders.

Recent work has demonstrated the potential of using building infrastructure for effective low-impact response (Wagner et al. 2018). The study, based on a validated CONTAM (*CONTAM* is a software tool developed by NIST for modeling multi-zone airflow and contaminant transport in buildings.) model of an office building, shows that increasing the HVAC ventilation to the affected area resulted in a 70% reduction in the contaminant spread to the adjacent rooms (measured 30 min after release). The most effective response depends on the hazardous agent type (chemical or biological), source location (indoor or outdoor), and concentration. Different automated building HVAC control actions can be enacted following a detection to minimize exposure to such agents during the initial release. This may include increasing air exchange rates between outdoor and affected indoor space, shutting down the HVAC system that serves the affected zones to confine the contaminant, and pressurizing the unaffected zones with outdoor air. For a building that is equipped with a chemical or biological filtration system, an immediate response to an outdoor release

will be to over-pressurize the building with the treated air, thereby preventing contaminated outdoor air from entering the building. While these responses will help slow the spread of contaminants in the building, the occupants must be monitored for thermal comfort and O₂ deprivation, and the response strategy must preserve the reliability of the HVAC equipment (e.g., limit air intake during subfreezing weather conditions). Selecting the best response for a given scenario calls for an advanced control strategy that effectively coordinates the use of building HVAC controls and other assets (e.g., access control) while respecting the building's safe operational constraints.

Cyber-Attack

As buildings are becoming more intelligent and (Internet) connected, their resiliency to cyber-attacks becomes vital. The current generation of Building Management Systems (BMS) are designed for functionality with little attention to their software and physical components security. The vulnerabilities in BMS can be exploited to create an emergency situation. An attacker can remotely turn off ventilation to a chemical room, allowing harmful fumes to spread in a building or turn off the lighting or ventilation system at the most inconvenient time to destabilize the occupants or cause an unplanned evacuation. The emerging grid-interactive technologies increase the building system's exposure, providing newer entry points for hackers to disrupt both the building and grid operations. Examples include an attacker who alters demand reduction event data (level or price) to mislead customers, resulting in sudden disruption of utility operations. Experts have asserted that a large-scale coordinated and simultaneous attack on grid-edge devices (including buildings) could have severe implications for reliable power grid services. The recent vulnerability assessments (<https://securityledger.com/2016/02/ibm-research-calls-out-smart-building-risks/>) and cyber security incidents (<https://ics-cert.us-cert.gov/>) have spurred

cyber-aware campaigns and development of advanced cyber intrusion and response methods.

There are ongoing efforts to protect industrial control systems including BMS from cyber-attacks. The popular information technology-based security designs (such as antivirus and patching policy, encryption, cryptographic mechanisms) can only provide assurance that cyber-attacks will be less likely to succeed but do not provide immunity against new and unknown attacks. Adverse cyber events must be treated as both security and resiliency issues. (*NIST Special Publication 800–160 Vol. 2 (DRAFT)*.) To achieve attack resiliency, the BMS needs to be continuously monitored for complete situational awareness and targeted response. Some of the existing schemes for cyber intrusion detection in buildings include approaches that use machine learning (Tonejc et al. 2016) and rule-based approaches (Pan et al. 2014) to detect anomalies in the BMS communication network. While there is a handful literature on anomaly detection for building systems, combining communication network and physical system data could enable accurate differentiation between cyber-attacks and equipment or operational faults. Such distinction is needed to ensure appropriate control response to cyber threats and inform near-term deployment of security hardening measurements. The control options may include automated isolation and restoration (Jones et al. 2018), reconfigurable and adaptive control methods that modify the control algorithm to reflect the compromised building state and maintain essential performance level.

Power Outage Management

Power outages in buildings are either directly caused by weather-inflicted faults (e.g., high wind, snowstorm, hurricanes, etc.) or indirectly by equipment failures, operational errors, or adversarial activities. Local power supplies (e.g., onsite diesel generators, renewable energy resources, and storage) are usually utilized as emergency backups to power critical loads during the outage. To maximize operational

efficiency, minimize cost, and prolong the coverage time of the local power system, an advanced priority-driven control algorithm can be deployed to optimally dispatch the available local energy assets and effectively manage the changing load priorities in the building. Although power outages are stochastic events, the extreme weather-induced outages can be predicted (in a probabilistic manner) by correlating multiple sources of data including historical outage and weather data from diverse sources, e.g., land-based sensor measurement stations, radar, satellite, and national lightning detection network (Kezunovic et al. 2018). Such predictive capability would enable proactive controls that optimally plan and respond to events. A distributed generation asset can be actively managed through supply and load shifting. Grid power can be accumulated in actual or virtual storage (e.g., building pre-cooling) prior to the event and be optimally discharged during the outage event.

Emergency Egress

Many emergency situations (e.g., fire) in buildings call for immediate evacuation. In some countries, directional signage provides guidance for evacuation. Recently, intelligent egress systems have been developed that use occupancy information in conjunction with information from other building systems (e.g., fire safety and elevator systems) to determine the best path for evacuation. These systems are designed to maximize the occupants' safety while minimizing evacuation time by diverting evacuees to less-congested paths suggested by optimal routing algorithms (Desmet and Gelenbe 2014; Nguyen et al. 2019). For state-of-the-art buildings, real-time information of building occupancy can also be provided to first responders (such as fire fighters and facility personnel) to inform response tactics and accelerate search and rescue. Early approaches for occupancy estimation include a hybrid approach where sensor (e.g., motion sensor, video camera) data are combined with a dynamic model of people movement

to estimate occupancy using extended Kalman filter (Tomastik et al. 2008). Recent methods are leveraging smart wearable devices, ubiquitous availability of connectivity, and advancements in artificial intelligence. Indoor occupancy estimation and localization technologies are becoming prevalent and affordable for emergency evacuation and other normal operational benefits (Labeodan et al. 2015; Zafari et al. 2017).

Summary and Future Direction

The rising number of attacks against occupied commercial buildings in the last two decades has increased the urgency of protecting the building facilities, users, and (critical) operations from threats. New sensors are being developed to detect life safety threats. These threat sensors can be integrated with existing building control systems to enable rapid situation awareness and effective response to attacks. The Building Management System (BMS) serves as a cost-effective platform to enable the combined prowess of different building subsystems for threat detection and mitigation. For example, a fire or hazardous agent detection system can trigger an HVAC control system to reduce/shutdown ventilation to affected areas and/or inform a security system to release certain access doors for use as escape routes and/or stop passengers in an elevator at a safe level (e.g., ground level). This platform can also be used to enhance response to other emergency events such as hazardous contaminants, grid outages, and extreme weather. Future R&D is needed for holistic solutions that effectively coordinate the operation of the diverse building assets, threat confirmers, and notification strategies to maximize life safety and provide information to first responders.

Cyber-resilient control methods integrate cyber-attack detection and diagnostics with control strategies to minimize the attack impact on the building system performance. Timely and high detection accuracy becomes necessary for effective attack mitigation. In some cases, intelligent attackers may keep the attack process hidden and undetectable by closely following

the expected behavior of the system. Resilient control methods that achieve robustness to such undetectable stealthy anomaly are equally important. Future research is needed to explore the applicability of advance defense mechanisms for building domain. Examples include moving target defense mechanisms (Giraldo et al. 2019; Kanellopoulos and Vamvoudakis 2019) for deceiving and impeding a stealthy attacker and game theoretic methods (Bakker et al. 2019) for mitigating the influence of persistent attacker.

Cross-References

- [Building Energy Management System](#)
- [Cyber-Physical Security](#)
- [Human-Building Interaction \(HBI\)](#)

Recommended Reading

Paper (Gelenbe and Wu 2013) provides a survey and research opportunities for evacuation support systems. For the general theme of cyber security of building, please refer to Wendzel et al. (2017).

Bibliography

- Bakker C, Bhattacharya A, Chatterjee S, Vrabie, D (2019) Learning and information manipulation: repeated hypergames for cyber-physical security. *IEEE Control Syst Lett*. Early Access. <https://doi.org/10.1109/LCSYS.2019.2925681>
- Desmet A, Gelenbe E (2014) Capacity based evacuation with dynamic exit signs. In: *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp 332–337
- Gelenbe E, Wu F-J (2013) Future research on cyber-physical emergency management systems. *Future Internet* 5(3):336–354
- Giraldo J, Cardenas A, Sanfelice RG (2019) A moving target defense to detect stealthy attacks in cyber-physical systems. In: *Proceedings of the American Control Conference (ACC)*, July 2019
- Jones CB, Carter C, Thomas Z (2018) Intrusion Detection & Response using an Unsupervised Artificial Neural Network on a Single Board Computer for Building Control Resilience. *Proceedings of the 2018 Resilience Week (RWS)*, Denver CO
- Kanellopoulos A, Vamvoudakis KG (2019) A moving target defense control framework for cyber physical

- systems. *IEEE Trans Autom Control*. Early Access. <https://doi.org/10.1109/TAC.2019.2915746>
- Kezunovic M, Obradovic Z, Dokic T, Roychoudhury S (2018) Systematic framework for integration of weather data into prediction models for the electric grid outage and asset management applications. In: *Proceedings of the 51st Hawaii International Conference on System Sciences*
- Labeodan T, Zeiler W, Boxem G, Zhao Y (2015) Occupancy measurement in commercial office buildings for demand-driven control applications – a survey and detection system evaluation. *Energ Buildings* 93:303–314
- Nguyen V, Nguyen H, Huynh Q, Venkatasubramanian N, Kim K (2019) A scalable approach for dynamic evacuation routing in large smart buildings. In: *IEEE International Conference on Smart Computing (SMART-COMP)*. <https://doi.org/10.1109/SMARTCOMP.2019.00065>
- Pan Z, Hariri S, Al-Nashif Y (2014) Anomaly based intrusion detection for building automation and control networks. In: *Proceedings of IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, pp 72–77
- Tomastik R, Narayanan S, Banaszuk A, Meyn S (2008) Model-based real-time estimation of building occupancy during emergency egress. In: *Pedestrian and Evacuation Dynamics*, pp 215–224
- Tonejc J, Guttus S, Kobekova A, Kaur J (2016) Machine learning methods for anomaly detection in BACnet networks. *J Univ Comput Sci* 22(9):1203–1224
- Wagner T, Taylor R, Reeve H (2018) Leveraging intelligent building infrastructure for event response. In: *Proceedings of International High Performance Buildings Conference, Purdue*
- Wendzel S, Tonejc J, Kaur J, Kobekova A (2017) Cyber-security of smart buildings. In: *Security and privacy in cyber-physical systems*. Chapter 16, pp 327–35. Wiley. <https://doi.org/10.1002/9781119226079.ch16>
- Zafari F, Gkelias A, Leung K (2017) A survey of indoor localization systems and technologies. *IEEE Commun Surv Tutorials*. <https://arxiv.org/pdf/1709.01015.pdf>

Engine Control

Luigi del Re¹ and Carlos Guardiola²

¹Johannes Kepler Universität, Linz, Austria

²Universitat Politècnica de València, Valencia, Spain

Abstract

Engine control is the enabling technology for efficiency, performance, reliability, and clean-

liness of modern vehicles equipped with a combustion engine for a wide variety of uses and users. It has also a paramount importance for many other engine applications like power plants. Engines are essentially chemical reactors, and the core task of engine control consists in preparing and starting the reaction (mixing the reactants and igniting the mixture) while the reaction itself is not controlled. The technical challenge derives from the combination of high complexity, wide range of conditions of use, performance requirements, significant time delays, and several constraints. In practice, engine control is to a large extent feed-forward control, feedback loops being used either for low-level control or for updating the feed forward. Industrial engine control is based on very complex structures calibrated experimentally, but there is a growing interest for model-based control with stronger feedback action, supported by the breakthrough of new computational and communication possibilities, as well as by the introduction of new sensors.

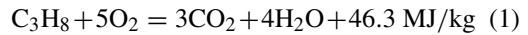
Keywords

Compression ignition · Emissions · Exhaust aftertreatment · Internal combustion engines · Spark ignition

Introduction

Even though electrification is becoming more important, most vehicles are moved by internal combustion engines (ICEs), and at least for some segments, like heavy duty, this is not expected to change for some time, mainly due to the energy density of fuels and to their well-established supply chain. ICEs provide traction power not only for road vehicles but also for off-road vehicles, ships, planes, and in some cases trains. The ICE is also a main element of alternative power plants as hybrid electrical vehicles (HEV) or may act as enablers for others, as in the case of their application as range extenders for plug-in hybrid electrical vehicles (PHEV).

The key function of an ICE is the conversion of chemical energy into mechanical energy, basically by oxidation, e.g., in the case of propane:



The chemical energy is first transformed into heat and converted by the ICE into mechanical energy through a thermodynamic cycle (Heywood 1988). The key task of engine control (Kiencke and Nielsen 2005) is to make sure that the reactants (fuel and oxygen) meet in the right proportion (“mixture formation”) and that the combustion is timely started (or “ignited”) to deliver the required torque at the engine crankshaft. Despite the apparent simplicity, the engine control has additional multiple tasks, the main ones being the following:

- To operate the system in an efficient way, adapting the engine operation to the varying ambient and use conditions while ensuring a smooth and agreeable torque delivery (or “driveability”).
- To keep the toxic by-products of the combustion below given threshold values, according to the certification levels required by the local legislations.
- To keep other undesired emissions (the so-called NVH - noise, vibration, and harshness) below given threshold values.
- To protect engine subsystems and limit the fatigue of components.
- To diagnose the operation of the system and to detect faults that could be a risk for the user or the environment.

It must be noticed that, beyond of the engine itself, the engine control is in charge to control, monitor, and coordinate many peripheral elements and subsystems. Relevant examples are the exhaust aftertreatment system, whose operation constraints the engine combustion settings, or the thermal management system. In a parallel way and significantly in vehicular applications, engine control must be coordinated with other powertrain and vehicular systems and with a plethora of

torque, braking, and driving assistance functions. The complexity of the vehicle architecture has exploded in recent years, and the propagation of the subsystem constraints for a safe operation to the supervisory layer represents one of the major challenges, as it also is to ensure the driveability of the system as a whole, which must be evaluated by subjective criteria. All these requirements have to be met for all vehicles of a production model in spite of production variability and under all relevant operating conditions, including all drivers, fuel specifications, road, traffic, and weather/altitude conditions.

Target Systems

The ICE may be classified according to its combustion process. While several processes are known, most of commercially available engines rely on two radically opposite concepts: spark-ignited (SI) and compression-ignited (CI) engines. The former, also referred as Otto engines, uses a nearly stoichiometric premixed charge of air and a low-reactivity fuel, and the combustion is triggered by an electrical spark, which allows an accurate control of the combustion phasing. The latter, known also as diesel engines, injects a high-reactivity fuel in a previously compressed inert cylinder charge, resulting in a diffusion flame with a low global equivalence ratio (i.e., fuel-to-air ratio). Figure 1 schematizes the operation of these two engine variants for the case of the more popular four-stroke implementation, which completes an engine cycle for every two turns of the crankshaft.

Figure 1 shows the basic setup of an ICE as CI and SI. In both cases, the main components of an ICE are air path, fuel path, combustion chamber, and exhaust path.

Basic air path layout is similar in SI and CI engines. While throttle valves are now present on both types, the nearly stoichiometric requirement for most conditions of SI engines implies that the air flow is frequently actively reduced by throttling, different from CI engines in which throttles are used only under special conditions.

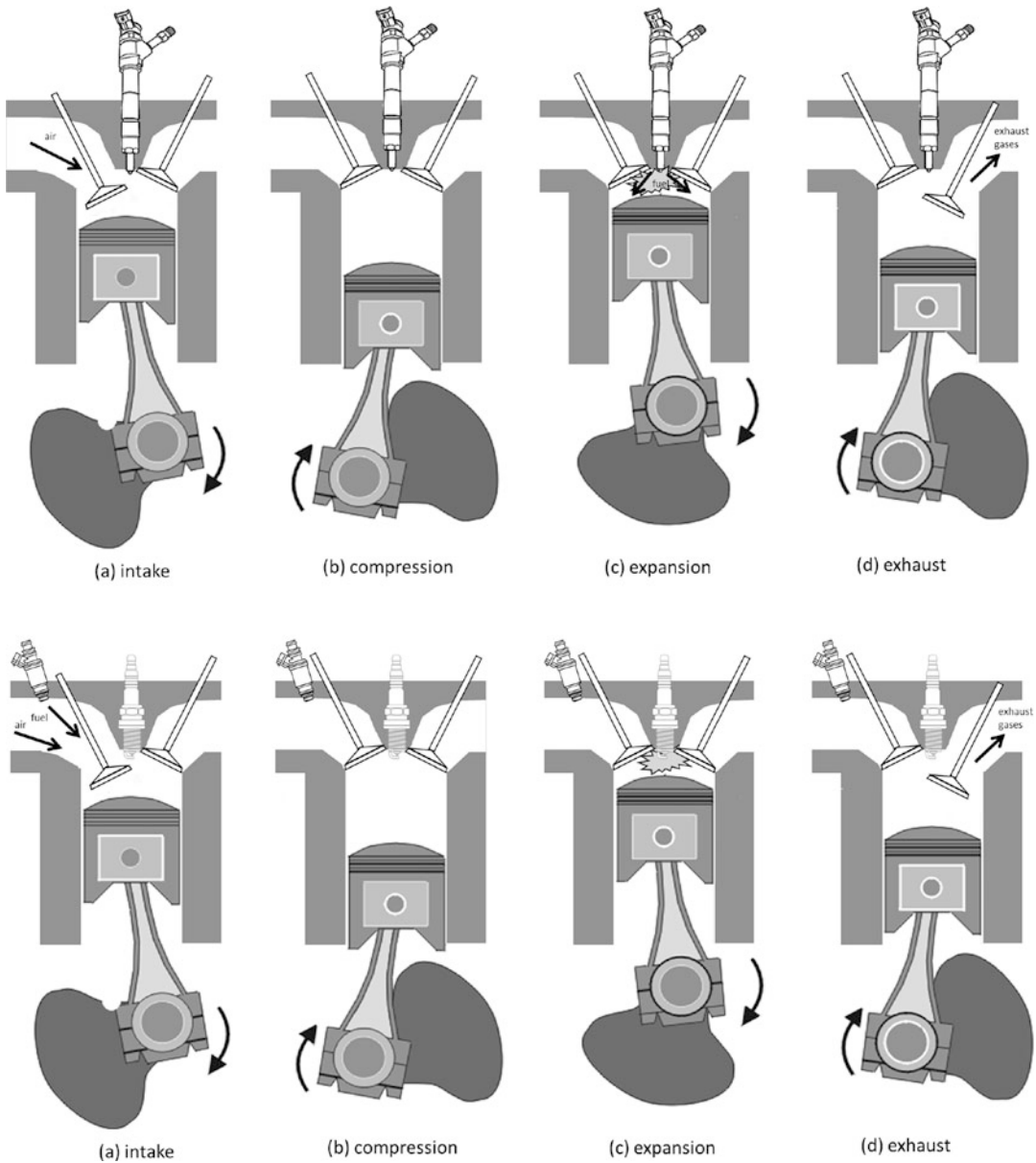
Turbochargers are now widespread for both automotive CI and SI applications, as well as exhaust gas recirculation (EGR) circuits and, to a small extent, variable timing systems for the intake and exhaust valves of the combustion chamber.

CI combustion, which relies in a nonhomogeneous diffusion flame, requires a fuel with a high tendency to autoignition (diesel, oil, esters, etc.). In state-of-the-art engines, fuel is injected through the use of high-pressure injection systems (known as common rail injection systems), which allows a huge flexibility in the number, timing, and duration of the injection events. Differently, SI combustion is characterized by a flame front in a homogeneous mix and the use of high-octane fuels (gasoline, natural gas, liquefied petroleum gas, alcohols, etc.). Traditionally, the mixture was prepared outside of the combustion chamber, but most new SI engines (denoted as gasoline direct injection, GDI) are fitted with direct fuel injection systems similar to those of CI engines, which allow the mixture to be prepared in the cylinder and provide a certain degree of mix stratification.

Aftertreatment system layout depends on the engine combustion, since pollutant characteristics and oxygen excess in the exhaust gases restrict the selection of the aftertreatment elements. Three-way catalysts (TWC) are the widespread solution for SI engines, and in some cases, a gasoline particulate filter (GPF) is added; for CI engines, there is no universal solution, but a combination of diesel oxidation catalyst (DOC), diesel particulate filter (DPF), and selective catalytic reduction (SCR) has proven to be effective.

Roughly speaking, ICEs exhibit three time scales:

- Since ignition, combustion, exhaust, and intake processes at each cylinder occur once every thermodynamic cycle, many variables present significant variations along the cycle; for the case of a κ -cylinder four-stroke engine, the basic frequency is $\kappa \frac{n}{2}$ (which equals a 100 Hz for a four-cylinder engine at 3000 rpm) (Guardiola et al. 2020a).



Engine Control, Fig. 1 Operation of four-stroke CI (*top*) and SI (*bottom*) engines

- Some of the actuators must be synchronized with the cycle process (i.e., *crank angle synchronous*, as opposed to *time synchronous*) and allow a cycle-to-cycle control of the system, as it is the case of the fuel injection and spark ignition. On the other hand, air path dynamics are typically in the range of 0.5–5 Hz, due to mass accumulation in manifolds and turbocharger inertia.
- Finally, thermal processes of the engine and aftertreatment system are significantly slower, remarkably below 0.1 Hz, and many times with characteristic times of several minutes.

The Control Tasks

Engine systems are highly optimized; their control is no exception. The basic control task they address can be defined as the minimization of the average fuel consumption while providing the required torque and respecting the constraints on emissions (i.e., nitrogen oxides and dioxides (NO_x) and particulate matter (PM)), noise, temperature, etc. The legislators in different countries have defined test procedure, including a specified speed profile and corresponding emission limits. Figure 3 shows the progressive reduction of the limits and the speed profile used to assess this value.

Even if fuel consumption is not limited by law at a vehicle level (since only fleet averages must be satisfied), the associated key control problem can be stated as an optimal constrained control problem (besides the respect of instantaneous limits on actuators and states, e.g. maximum speed

of the turbocharger):

$$\min_{u(t)} \int_0^{t_{\max}} \dot{q}_f dt, \quad (2)$$

so that

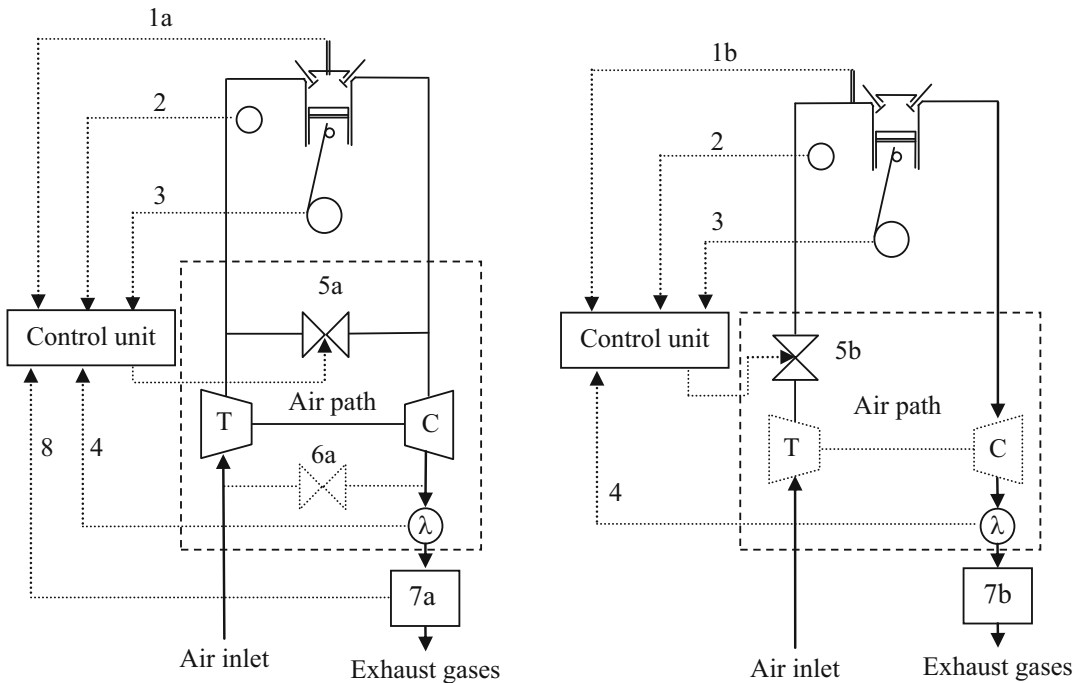
$$v(t) = v_{\text{dem}}(t) \pm \Delta v \quad (3)$$

and

$$\int_0^{t_{\max}} \dot{q}_i dt \leq Q_i \quad (4)$$

where $u(t)$ are all available control inputs, $t_{\max}$ is the duration of the considered driving cycle (e.g., the WLTC cycle in Fig. 3), $v_{\text{dem}}(t)$ is the corresponding speed, Δv is the speed tolerance, $\dot{q}_f$ is the instantaneous fuel consumption, $\dot{q}_i$ is the each limited quantity (e.g., NO_x), and Q_i is the corresponding limit for the whole test. Notice that different countries use different certification cycles

E



Engine Control, Fig. 2 Basic system scheme of CI (left) and SI (right) engines: 1a Control of the injector opening; 1b injection premixing with air; 2 measurement of the engine temperature; 3 measurement of the engine rotational speed; 4 measurement of oxygen concentration in

the exhaust gases; 5a EGR valve; 5b throttle valve; 6a low-pressure EGR valve; 7a diesel exhaust aftertreatment (DOC, SCR, DPF); 7b SI engine aftertreatment (three-way catalyst); 8 SCR dosing control

(e.g., FTP cycle series in the United States), and the new engines are tested using the so-called real drive emissions (RDE), in which a sample vehicle is tested in real traffic conditions. This is a significant change in the certification approach, since the driving profile is not known beforehand: in RDE approach, only speed and acceleration metrics and general boundary conditions must be met for a driving profile to be considered as a valid test.

In practice, as already mentioned, other criteria must be considered as well, like NVH, and the control must be robust with respect to several parameters. For instance, most engines have several cylinders, and their combustion properties (including the properties of the injectors) are not exactly equal. On one side, engine control functions detect different torque production in the single cylinders and automatically compensate them; on the other side, some safety margins must be used to make sure that the real emissions will not pass the limit values. Even though this can be easily stated as an optimal problem as in Fig. 2 to Fig. 4, it is never solved in this way: on one side because there is no sufficiently good model which could be used to find the solution and on the other side in view of the enormous complexity.

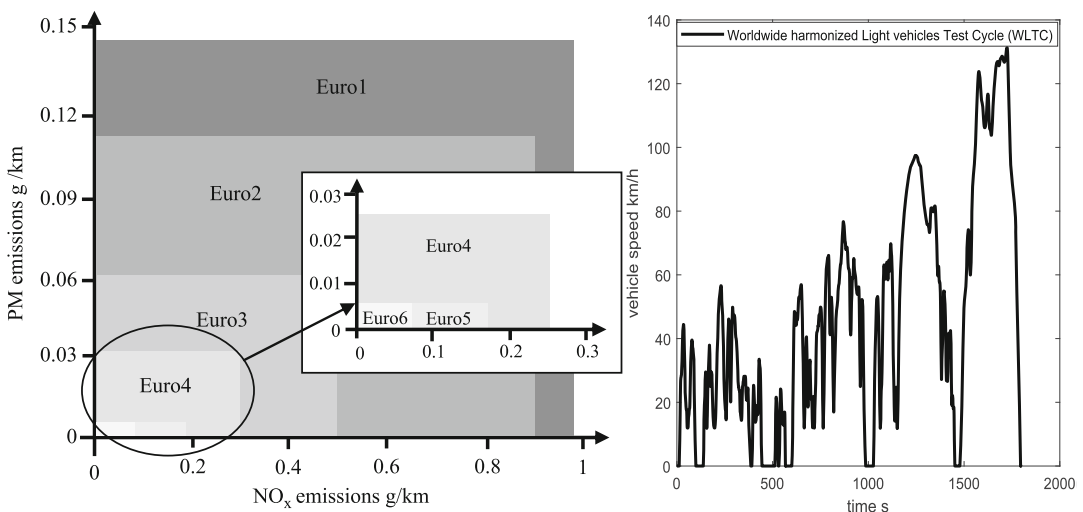
In practice, different simpler subproblems are solved separately by heuristic descriptions –

typically maps – which can be tuned experimentally and even automatically (Schoggl et al. 2002); a modern engine control unit (ECU) can include up to 40,000 labels (parameters or maps). The resulting structure is mainly feed forward, with feedback loops typically used for control of actuators, primarily calibrated under laboratory conditions but with adaptation loops designed to correct parameters to take in account production and wear effects. Figure 4 shows an engine test bench setup with the engine control unit (ECU) and a calibration system.

While there is a wide agreement on the limits of this heuristic approach, there is no simple way out, as translating the actual ECU into a model-based structure would require an enormous effort. However, model-based parts are included, e.g., the wall-wetting compensation in the SI engines. In the following, we review some key parts of the combustion control.

Air Path Control

The main source of oxygen for the reaction of Eq. (1) is ambient air which contains about 21% oxygen. The engine – essentially a volumetric air pump – aspires air flow roughly proportional to the cylinder volume and the revolution speed of the engine. The amount of



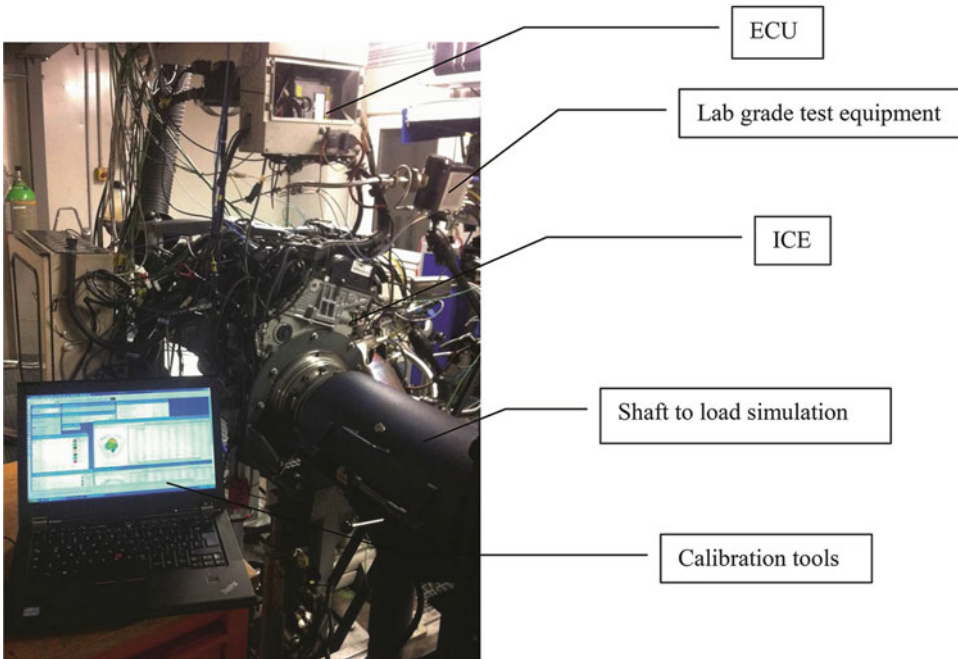
Engine Control, Fig. 3 Left: different steps of limits of emissions per km as defined by the European Union (Euro 1 introduced in 1991 and Euro 6 from 2014). Right: New European Driving Cycle (NEDC)

gas entering the combustion chamber (referred as the cylinder charge), however, will depend also on temperature and pressure. This charge can be reduced by throttling (i.e., adding an additional flow resistance between the air intake and the combustion chamber, standard in SI engines) or increased by compressing the fresh air, most commonly by turbocharging. A turbocharger consists essentially of a turbine, which transforms part of the enthalpy of the exhaust gas into mechanical power, and a compressor, driven by this power to compress the fresh air on its way to the combustion chamber, thus increasing both its density and temperature. Turbocharger operation is typically controlled either directly (for instance, with variable vane angles) or indirectly, by bypass valves (known as waste gates) which deviate the gas flows in parallel to the turbine.

Variable valve timing (VVT) refers to several technologies aiming to modify valve timing and/or the valve lift. Some electrohydraulic systems already in the market allow a high flexibility of the valve timing with a cycle-to-cycle response time, while others rely on slower actuation over

camshaft relative phase or valve actuation chain. Some of the advantages of the fully flexible VVT systems are the control of the engine charge without intake manifold throttling, the effective compression ratio control, or the application of alternative engine cycles as Miller and Atkinson concepts through the modification of the compression and intake stroke. Finally, they may also serve for the cycle-to-cycle control of the cylinder charge.

Fresh air dilution may be used for controlling cylinder charge composition: if fresh ambient air is mixed with exhaust gases, resulting mix will have an oxygen concentration below 21% and an increased percentage of inert gases as CO₂ and moisture. This provides advantages in terms of NO_x reduction as it depends on adiabatic flame temperature which is strongly correlated with oxygen concentration. Therefore, in some engines, especially in CI engines, part of the combusted gases is recirculated to the combustion chamber (“exhaust gas recirculation,” EGR), as an active method for NO_x control. While EGR is typically realized at high pressures (as in Fig. 1), it may also be realized on the low pressure side



Engine Control, Fig. 4 Light-duty engine test bench with ECU and calibration system (JKU laboratory)

of turbine and compressor. Typically, the air path includes some coolers designed to increase gas density.

Air path control is designed to track dynamical references, for instance, the total fresh air mass flow (MAF) entering the cylinder and the corresponding manifold air pressure (MAP), but also other quantities are possible. The references are typically generated by the calibration engineers on the basis of tests. The control inputs of the air path are mostly the turbine (and possibly compressor) steering angle, the EGR, and – if available – throttle(s) set points. Most commonly used sensors include a mass flow meter (hot film sensor), rather slow and dynamically not reliable; pressure and temperature sensors; and sometimes the turbocharger speed as well. The actual position of the involved valves – throttle, turbine, or EGR – is usually measured, thus allowing cascade control and correcting the actuator hysteresis.

Fuel Path Control

The fuel path delivers the correct amount of fuel for the reaction (1). In almost every ICE, a rail is filled with fuel at a given pressure (from a few up to 400 bar for SI to about 3000 bar for CI), from which the required amount of fuel is injected into the cylinder. The injection can occur inside the combustion chamber (as for CI and GDI engines) or near the intake valve (“port injection”) for standard SI engines.

The injection amount corresponds roughly to the desired torque but is always constrained taking into account the available oxygen mass. In SI engines with three-way catalyst, the fuel injection is given by the stoichiometric condition, i.e., the ratio between the fuel quantity which can be perfectly burnt and the injected fuel, the so-called λ -factor, is exactly 1 ($\lambda > 1$ is called lean mixture or air excess and is common for diesel and GDI engines but prevents the use of a TWC). λ control uses an oxygen sensor in the exhaust to determine the actual λ for instantaneous control and if appropriate slowly corrects the injection tables. In CI and GDI, the maximum fuel injection is limited to prevent smoke formation, typically by tables, even though λ control can be and is partly used (Amstutz and del Re 1995). Manufacturing

discrepancies and aging are compensated by the use of the λ control, while individual cylinder balancing is usually performed by the analysis of the variation of the engine speed along the combustion cycle.

In SI engines with port injection, the liquid fuel is injected near the inlet valve and is expected to vaporize due to the local temperature and pressure conditions. During load changes, however, it can happen that part of the fuel is not vaporized, remains on the duct wall (“wall wetting”), and vaporizes at a later time, leading in both cases to a deviation from the expected values (Turin and Geering 1995), which must be compensated by the injection control.

Direct injection systems allow a more direct control of the injection event and usually allow several injections per cylinder and cycle. Injection in CI engines is typically split in a main injection for torque, a pilot injection for NVH control, and sometimes also a postinjection for emission control or regeneration of aftertreatment devices. In GDI engines, injection may allow an active control of the combustion mode: if the injection is performed during the intake stroke, a homogeneous mix is formed, and the combustion is like in conventional SI engines; if the injection, sometimes split in a couple of shorter injections, is delayed, the mix may be stratified with advantages in terms of fuel efficiency. This latter strategy needs a careful coordination of the injection and ignition control.

Ignition

Once the combustion chamber is filled, the combustion can be started. In traditional SI and GDI, combustion is started by a spark, leading to a flame front which propagates through the whole combustion chamber. Because of differences in the flame propagation, these engines are affected by significant cycle-to-cycle variability. Spark timing and energy are used for reducing this variability. In challenging situations, multiple consecutive sparks may be used; very few SI engines have a second spark plug to better control the ignition process.

In addition to the cycle-to-cycle variability, faulty cycles may occur. Misfires may be related with λ very different from 1 or problems in the

spark plugs. Misfires must be detected as they significantly impact pollutant emissions. Usual detection strategy relies in analyzing the in-cycle variation of the crankshaft acceleration, which allows isolating faulty cycles and detects persistent faults.

Under some circumstances, e.g., high temperature, an undesired autoignition (“knock”) can occur with potentially catastrophic consequences for the engine durability and also unconventional NVH. To cope with this, SI engines have vibration sensors whose output is used to modify the engine operation, in particular the spark timing, to prevent knock occurrence. In such engines, the spark timing is continuously varied for each cycle: it is slightly advanced with respect to the previous cycle if no knock is detected, while it is delayed if previous cycle was classified as a knocking cycle. Such simple control allows to adapt the engine to the fuel quality and the varying ambient and use conditions.

In CI engines, there is no direct actuator for the ignition, since the combustion is a consequence of the self-ignition of the fuel as it is injected. Thus, injection settings and charge oxygen concentration crucially impact the combustion timing and the pollutant formation. While in-cylinder pressure sensors can provide valuable information about the combustion, most engines rely on a collection of lookup tables for adapting the settings to the use and operative conditions.

Aftertreatment

Pollutant formation is the undesirable side effect of combustion. In a perfect combustion, as in (1), the combustion will only produce water and CO_2 (which is not considered as a pollutant in the classical sense as it is the natural product of a complete combustion); however, due to partial combustion or side reactions, additional species may occur. Incomplete fuel burnt yields to hydrocarbons (HC), carbon monoxide (CO), and particulate matter (PM), which is the aggregation of soot and heavy HC from fuel or lubricant partial combustion. Even if much effort is spent on reducing their formation, this is almost never sufficient for reaching regulated limits, so additional aftertreatment equipment is used. Table 1 gives an

overview over the most common aftertreatment systems as well as over their control aspects.

Modern automotive engines usually combine a series of systems for the control of the more relevant pollutant species. For the case of the SI engines, operating in nearly stoichiometric combustion, three-way catalysts (TWC) provide a compact and cost-effective solution for the joint control of HC, CO, and NO_x . The catalyst is able to reduce NO_x and, at the same time, oxidizes CO and HC; it also has the capacity of storing a certain amount of excess oxygen, so it can control short excursions of the systems out of the specified control reference. For TWC to be effective, it requires $\lambda \approx 1$.

For the case of CI engines, which operate with excess oxygen, aftertreatment is more challenging. While diesel oxidation catalysts (DOC) are able to oxidize CO and HC making use of the oxygen in the exhaust, NO_x needs an active approach. Selective catalytic reduction (SCR) uses an injection of a mix of water and urea for generating ammonia (NH_3) as reductor agent for abating the NO_x . Note that a low value of the urea dosing will yield NO_x emissions, while an excess will imply the existence of ammonia slip. Optimal urea dosing is the main difficulty for SCR: not only the trade-off between NH_3 slip and NO_x conversion must be considered, but also the urea consumption must be kept controlled for avoiding unscheduled refilling. Closed-loop control is needed for fulfilling the regulation and diagnosis, and NO_x sensors are already in production. SCR systems may be controlled despite the existence of several chemical species in the exhaust and the cross sensitivity of NO_x sensors to NH_3 slip (Guardiola et al. 2020b).

An important problem of the current aftertreatment technology is the need to reach a certain temperature for the catalyst to be effective, known as light-off temperature (Guardiola et al. 2018b). In order to avoid excessive emissions during cold phases of the engine operation, some elements may be included with storing capacities. For example, DOC is usually able to store a certain amount of HC at low temperature; the HCs are later burnt with exhaust excess oxygen when the catalyst meets the light-off

Engine Control, Table 1

Main exhaust
aftertreatment systems

System	Purpose	Control targets
Three-way catalyst	Reduction of HC, CO, and NO _x by more than 98%	Achieve fast and maintain operating temperature and keep $\lambda = 1$
Oxidation catalyst	Reduction of HC and CO, partly of PM	Achieve fast and maintain operating temperature and keep $\lambda > 1$
Particulate filter	Traps PM	Check trap state and regenerate by increasing exhaust temperature for short time if needed
NO _x lean trap	Traps NO _x	Estimate trap state and shift combustion to CO rich when required
Selective catalytic reduction	Reduces NO _x	Estimate required quantity of additional reactant (urea) and dose it

temperature. Lean NO_x traps (LNT) are able to store NO_x at low temperature, which are later released when the SCR reaches the light-off temperature or alternatively reduced using excess CO and HC generated by modifications on the air-to-fuel ratio control during the so-called regeneration phase. It must be noticed that a same catalyst brick can be designed for combining several of the aforementioned functionalities.

Finally, PM is usually filtered through particulate filters in CI (DPF) and GDI (GPF) engines. Particulate filters accumulate the generated PM and, depending on the engine kind, must be actively regenerated when a given amount of particulates is stored. In some engines, exhaust temperature is sufficient for a periodic burnt of the accumulated PM. For the case of modern diesel engines, DPF regeneration makes use of delayed postinjections producing excess HC which are burnt at the DOC rising the exhaust temperature above 600 °C. Closed-loop control of the exhaust temperature is usually implemented for controlling these regeneration events, which are usually scheduled according to a combination of models and pressure drop measurements. Note that regenerating a filter with an excessive amount of particulates may result in excessive temperature, destroying the filter itself or more commonly other catalysts in the exhaust line (as the SCR).

Cranking and Idle Speed

Initially, the engine is cranked by the starter until a relatively low speed, and then, injection starts bringing the engine to the minimum operational speed. This operation can be performed without difficulty in favorable conditions (warm engine and ambient conditions around the standard), but it can be challenging in other situations as it can lead to emission spikes which could not be withhold by the exhaust aftertreatment system. Thus, a specific strategy is necessary that is robust enough so that a proper engine start occurs even when the conditions are not favorable. This action is further conditioned by the reduced power of typical engine starters, which are not able to rotate the engine at a high speed (in general, half of the idle speed is usually not exceeded).

During the cranking process, special control actions are set into place with the sole purpose of obtaining combustion in the first cycles; once rotational speed is above a given threshold, engine stall is unlikely, and warm-up process may start. Some of the strategies articulated during these first cycles include nonconventional fueling strategies able to compensate for the very cold intake and cylinder walls, use of a train of sparks in SI engines, and wide open throttle valve.

Normally, an ICE is expected to provide a torque to the driveline, speed being the result of the balance between it and the load. In idle control however, no torque is transmitted to the driveline: the engine is declutched, and it is expected

to remain stable in spite of possible changes of local loads (like cabin climate control or engine ancillary systems). This is essentially a robust control problem (Hrovat and Sun 1997).

In many modern engines, idling is avoided as much as possible, and the engine is stopped when no torque is demanded. This strategy, denoted as stop-start, requires frequent cranking of the engine and usually needs a better specification of the starter. With a correct thermal management, controlling the position of the crankshaft at the engine stop, and the right injection settings, modern systems are able to achieve combustion in the first cycle for avoiding misfires resulting in emission and NVH level increase.

Thermal Management

Despite being one of the main loss flows in the ICE, heat has become a precious resource during engine operation. Specific actions are to be implemented during warm-up phase or when extra heat flow is required for aftertreatment regeneration. Engine heat is also required for other purposes (like defrosting of windshields in cold climates). In this sense, stop-start adds additional complexity from the thermal management point of view, as it is the integration of the engine into HEV powertrains, resulting in the engine to be stop for extended periods. Electrical heated systems can add an extra degree of freedom and can be used for accelerating the light-off temperature of different aftertreatment devices, but they heavily impact overall efficiency.

All main properties of engines are strongly affected by their temperature, which depends on the varying load conditions. Engine operation is typically optimal for a relatively narrow temperature range; the same is even more critical for the exhaust aftertreatment system. On the other hand, control settings must be significantly modified when the engine is operating far from its reference temperature. Thus, the engine control system has two main tasks: bringing the engine and the exhaust aftertreatment system as fast as possible into the target temperature range and taking into account deviation from this target. The first task is performed both by control of the cooling circuit and by specific

combustion-related measures, the second one by taking the measured or estimated temperature as input for the controllers.

A number of measures are implemented during the warm-up operation of the engine: valves in the cooling system are set to avoid heat transfer to the ambient, and specially designed combustion modes with excess exhaust heat – even if impacting fuel efficiency – are set. Note that if the aftertreatment is not able to store the pollutants while the catalysts are below the light-off temperature, this will critically impact pollutant emission levels; this is the case of SI engines simply relying on a TWC.

New Trends

Over the past decades, the importance of the engine control system has been increasing enormously, and this trend is expected to continue. The importance of electronic control in the engine will be reinforced by the following:

- Increase in the number of electronically controlled systems: more and more systems are managed by the control unit, and it is foreseeable that in the future, the number of electronically controlled actuators will increase further. Electrification is steadily reaching elements that have historically been mechanically actuated, such as the lubricant pump or the coolant pump, adding new degrees of freedom. In parallel with the grow in actuators, the list of available sensors is steadily growing.
- The increase in engine flexibility requirements, which must be able to adapt to different operating modes and possibly different fuels, all with increased robustness and self-diagnostic capacity.
- Increased coordination among the different vehicle systems. Within this framework and in relation to the layered structure of the control software, it is possible that the control of different components is carried out in a single controller or, more likely, in distributed systems that share high-level information through data buses.

- Increased coordination among vehicles, through infrastructure-to-vehicle (I2V) and vehicle-to-vehicle (V2V) communication, which may allow for traffic management actions, improved security, cooperative adaptive cruise control (CACC), or even self-guided vehicle caravan systems (referred in literature as “platooning”).

Against this background, it is expected that both the current control system structure and the calibration process associated with it will undergo sustained changes. In order to simplify the calibration process and to allow optimal control of the system, two lines of evolution (not mutually exclusive) can be observed:

- Development of model-based control systems, which are based on the existence of a model (physical or mathematical) that describes the operation of the engine. The model can be used to derive the laws of control, or it can be used as a prediction model within a predictive control system (del Re et al. 2010). The improvements in this approach reside, on the one hand, in reducing the number of tests required during calibration and, on the other one, in the possibility of deriving optimal control laws based on the prediction provided by the model. The first advantage is especially evident when using physical models (as mean value engine models, MVEM), since the model contains a description of the physical phenomena and therefore needs less identification effort (Alberer et al. 2012). Some model-based controls have already found their way into the ECU, and the academy has shown in several occasions that model-based control is able to achieve better performance, but it has not yet been shown how this could comply with other industrialization requirements. In many cases, models are included in the ECU for inferring some of the variables that are not directly measured (i.e., “virtual sensors”), but are not integrated in predictive controls (Guardiola et al. 2017).
- Use of adaptive systems so that the control system is able to improve its performance

based on measurements provided by the sensors and through certain learning laws. An adaptive system would allow to absorb manufacturing variability and the effects of drift and aging and adapting to real-use conditions (Guardiola et al. 2016). In addition, it could be used to adapt models in real time, which can be utilized for use in the control system or for engine diagnostics. Data connectivity makes it possible to manage this learning in a cloud environment and access data of different units (as opposed to embedded ECU approach); artificial intelligence techniques can be used for processing and clustering the huge data stream at a fleet level. Over-the-air ECU firmware update is already a possibility, so embedded control unit software can be updated as new information is available. While unsupervised learning for control system stands as a difficulty because of safety concerns, significant advancements are expected in this direction. Also regulation bodies are skeptical in view of possible abuse to circumvent tests, as shown by recent events involving a major manufacturer.

Cross-References

- [Powertrain Control for Hybrid-Electric and Electric Vehicles](#)
- [Transmissions](#)

Bibliography

- Alberer D et al (2012) Identification for automotive systems. Springer, London
- Amstutz A, del Re L (1995) EGO sensor based robust output control of EGR in diesel engines. *IEEE Trans Control Syst Technol* 3(1):39–48
- del Re L et al (2010) Automotive model predictive control. Springer, Berlin
- Guardiola C et al (2016) Adaptive calibration for reduced fuel consumption and emissions. *Proc Inst Mech Eng Part D J Automob Eng* 230(14):2002–2014
- Guardiola C et al (2017) Cycle by cycle NO_x model for diesel engine control. *Appl Thermal Eng* 110:1011–1020

- Guardiola C et al (2018a) An analysis of the in-cylinder pressure resonance excitation in internal combustion engines. *Appl Energy* 228:1272–1279
- Guardiola C et al (2018b) An on-board method to estimate the light-off temperature of diesel oxidation catalysts. *Int J Engine Res*. 1468087418817965
- Guardiola C et al (2020a) Closed-loop control of a dual-fuel engine working with different combustion modes using in-cylinder pressure feedback. *Int J Engine Res* 21(3):484–496
- Guardiola C et al (2020b) Model-based ammonia slip observation for SCR control and diagnosis. *IEEE/ASME Trans Mechatron*. <https://doi.org/10.1109/TMECH.2020.2974341>
- Guzzella L, Onder C (2010) Introduction to modeling and control of internal combustion engine systems, 2nd edn. Springer, Berlin
- Heywood J (1988) Internal combustion engines fundamentals. McGraw-Hill, New York
- Hrovat D, Sun J (1997) Models and control methodologies for IC engine idle speed control design. *Control Eng Pract* 5(8):1093–1100
- Kiencke U, Nielssen L (2005) Automotive control systems: for engine, driveline, and vehicle, 2nd edn. Springer, New York
- Payri F et al (2015) A challenging future for the IC engine: new technologies and the control role. *Oil Gas Sci Technol* 70(1):15–30. *Rev. IFP Energies nouvelles*
- Schoggli P et al (2002) Automated EMS calibration using objective driveability assessment and computer aided optimization methods. *SAE Trans* 111(3):1401–1409
- Turin RC, Geering HP (1995) Model-reference adaptive A/F-ratio control in an SI engine based on Kalman-filtering techniques. In: *Proceedings of the American Control Conference*, vol 6, Seattle

Estimation and Control over Networks

Vijay Gupta
Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

Estimation and control of systems when data is being transmitted across nonideal communication channels has now become an important research topic. While much progress has been made in the area over the last few years, many open problems still remain. This entry summarizes some

results available for such systems and points out a few open research directions. Two popular channel models are considered – the analog erasure channel model and the digital noiseless model. Results are presented for both the multichannel and multisensor settings.

Keywords

Analog erasure channel · Digital noiseless channel · Networked control systems · Sensor fusion

Introduction

Networked control systems refer to systems in which estimation and control is done across communication channels. In other words, these systems feature data transmission among the various components – sensors, estimators, controllers, and actuators – across communication channels that may delay, erase, or otherwise corrupt the data. It has been known for a long time that the presence of communication channels has deep and subtle effects. As an instance, an asymptotically stable linear system may display chaotic behavior if the data transmitted from the sensor to the controller and the controller to the actuator is quantized. Accordingly, the impact of communication channels on the estimation/control performance and design of estimation/control algorithms to counter any performance loss due to such channels have both become areas of active research.

Preliminaries

It is not possible to provide a detailed overview of all the work in the area. This entry attempts to summarize the flavor of the results that are available today. We focus on two specific communication channel models – analog erasure channel and the digital noiseless channel. Although other channel models, e.g., channels that introduce delays or additive noise, have been consid-

ered in the literature, these models are among the ones that have been studied the most. Moreover, the richness of the field can be illustrated by concentrating on these models.

An analog erasure channel model is defined as follows. At every time step k , the channel supports as its input a real vector $i(k) \in \mathbf{R}^t$ with a bounded dimension t . The output $o(k)$ of the channel is determined stochastically. The simplest model of the channel is when the output is determined by a Bernoulli process with probability p . In this case, the output is given by

$$o(k) = \begin{cases} i(k-1) & \text{with probability } 1-p \\ \phi & \text{otherwise,} \end{cases}$$

where the symbol ϕ denotes the fact that the receiver does not obtain any data at that time step and, importantly, recognizes that the channel has not transmitted any data. The probability p is termed the erasure probability of the channel. More intricate models in which the erasure process is governed by a Markov chain, or by a deterministic process, have also been proposed and analyzed. In our subsequent development, we will assume that the erasure process is governed by a Bernoulli process.

A digital noiseless channel model is defined as follows. At every time step k , the channel supports at its input one out of 2^m symbols. The output of the channel is equal to the input. The symbol that is transmitted may be generated arbitrarily; however, it is natural to consider the channel as supporting m bits at every time step and the specific symbol transmitted as being generated according to an appropriately design quantizer. Once again, additional complications such as delays introduced by the channel have been considered in the literature.

A general networked control problem consists of a process whose states are being measured by multiple sensors that transmit data to multiple controllers. The controllers generate control inputs that are applied by different actuators. All the data is transmitted across communication channels. Design of control inputs when multiple controllers are present, even without the

presence of communication channels, is known to be hard since the control inputs in this case have dual effect. It is, thus, not surprising that not many results are available for networked control systems with multiple controllers. We will thus concentrate on the case when only one controller and actuator is present. However, we will review the known results for the analog erasure channel and the digital noiseless channel models when (i) multiple sensors observe the same process and transmit information to the controller and (ii) the sensor transmits information to the controller over a network of communication channels with an arbitrary topology.

An important distinction in the networked control system literature is that of one-block versus two-block designs. Intuitively, the one-block design arises from viewing the communication channel as a perturbation to a control system designed without a channel. In this paradigm, the only block that needs to be designed is the receiver. Thus, for instance, if an analog erasure channel is present between the sensor and the estimator, the sensor continues to transmit the measurements as if no channel is present. However, the estimator present at the output of the channel is now designed to *compensate* for any imperfections introduced by the communication channel. On the other hand, in the two-block design paradigm, both the transmitter and the receiver are designed to optimize the estimation or control performance. Thus, if an analog erasure channel is present between the sensor and the estimator, the sensor can now transmit an appropriate function of the information it has access to. The transmitted quantity needs to satisfy the constraints introduced by the channel in terms of the dimensions, bit rate, power constraints, and so on. It is worth remembering that while the two-block design paradigm follows in spirit from communication theory where both the transmitter and the receiver are design blocks, the specific design of these blocks is usually much more involved than in communication theory. It is not surprising that in general performance with two-block designs is better than the one-block designs.

Analog Erasure Channel Model

Consider the usual LQG formulation. A linear process of the form

$$x(k+1) = Ax(k) + Bu(k) + w(k),$$

with state $x(k) \in \mathbf{R}^d$ and process noise $w(k)$ is controlled using a control input $u(k) \in \mathbf{R}^m$. The process noise is assumed to be white, Gaussian, zero mean, with covariance Σ_w . The initial condition $x(0)$ is also assumed to be Gaussian and zero mean with covariance Π_0 . The process is observed by n sensors, with the i -th sensor generating measurements of the form

$$y_i(k) = C_i x(k) + v_i(k),$$

with the measurement noise $v_i(k)$ assumed to be white, Gaussian, zero mean, with covariance $\Sigma_{v_i}^i$. All the random variables in the system are assumed to be mutually independent. We consider two cases:

- If $n = 1$, the sensor communicates with the controller across a network consisting of multiple communication channels connected according to an arbitrary topology. Every communication channel is modeled as an analog erasure channel with possibly a different erasure probability. The erasure events on the channels are assumed to be independent of each other, for simplicity. The sensor and the controller then form two nodes of a network each edge of which represents a communication channel.
- If $n > 1$, then every sensor communicates with the controller across an individual communication channel that is modeled as an analog erasure channel with possibly a different erasure probability. The erasure events on the channels are assumed to be independent of each other, for simplicity.

The controller calculates the control input to optimize a quadratic cost function of the form

$$J_K = E \left[\sum_{k=0}^{K-1} \left(x^T(k) Q x(k) + u^T(k) R u(k) \right) + x^T(K) P_K x(K) \right].$$

All the covariance matrices and the cost matrices Q , R , and P_K are assumed to be positive definite. The pair (A, B) is controllable and the pair (A, C) is observable, where C is formed by stacking the matrices C_i 's. The system is said to be stabilizable if there exists a design (within the specified one-block or two-block design framework) such that the cost $\lim_{K \rightarrow \infty} \frac{1}{K} J_K$ is bounded.

A Network of Communication Channels

We begin with the case when $N = 1$ as mentioned above. The one-block design problem in the presence of a network of communication channels is identical to the one-block design as if only one channel were present. This is because the network can be replaced by an “equivalent” communication channel with the erasure probability as some function of the reliability of the network. This can lead to poor performance, since the reliability may decrease quickly as the network size increases. For this reason, we will concentrate on the two-block design paradigm.

The two-block design paradigm permits the nodes of the network to process the data prior to transmission and hence achieve much better performance. The only constraint imposed on the transmitter is that the quantity that is transmitted is a causal function of the information that the node has access to, with a bounded dimension. The design problem can be solved using the following steps. The first step is to prove that a separation principle holds if the controller knows the control input applied by the actuator at every time step. This can be the case if the controller transmits the control input to the actuator across a perfect channel or if the control input is transmitted across an analog erasure channel but the actuator can transmit an acknowledgment to the controller. For simplicity, we assume that the controller transmits the control input to the actu-

ator across a perfect channel. The separation principle states that the optimal performance is achieved if the control input is calculated using the usual LQR control law, but the process state is replaced by the minimum mean squared error (MMSE) estimate of the state. Thus, the two-block design problem needs to be solved now for an optimal *estimation* problem.

The next step is to realize that for any allowed two-block design, an upper bound on estimation performance is provided by the strategy of every node transmitting every measurement it has access to at each time step. Notice that this strategy is not in the set of allowed two-block designs since the dimension of the transmitted quantity is not bounded with time. However, the same estimate is calculated at the decoder if the sensor transmits an estimate of the state at every time step and every other node (including the decoder) transmits the latest estimate it has access to from either its neighbors or its memory. This algorithm is recursive and involves every node transmitting a quantity with bounded dimension, however, since it leads to calculation of the same estimate at the decoder, and is, thus, optimal. It is worth remarking that the intermediate nodes do not require access to the control inputs. This is because the estimate at the decoder is a linear function of the control inputs and the measurements: thus, the effect of control inputs in the estimate can be separated from the effect of the measurements and included at the controller. Moreover, as long as the closed loop system is stable, the quantities transmitted by various nodes are also bounded. Thus, the two-block design problem can be solved.

The stability and performance analysis with the optimal design can also be performed. As an example, a necessary and sufficient stabilizability condition is that the inequality

$$p_{\max\text{cut}}\rho(A)^2 < 1,$$

holds, where $\rho(A)$ is the spectral radius of A and $p_{\max\text{cut}}$ is the max-cut probability evaluated as follows. Generate cut-sets from the network by dividing the nodes into two sets – a source set containing the sensor and a sink set containing

the controller. For each cut-set, obtain the cut-set probability by multiplying the erasure probabilities of the channels from the source set to the sink set. The max-cut probability is the maximum such cut-set probability. The necessity of the condition follows by recognizing that the channels from the source set to the sink set need to transmit data at a high enough rate even if the channels within each set are assumed not to erase any data. The sufficiency of the condition follows by using the Ford-Fulkerson algorithm to reduce the network into a collection of parallel paths from the sensor to the controller such that each path has links with equal erasure probability and the product of these probabilities for all paths is the max-cut probability. More details can be found in Gupta et al. (2009a).

Multiple Sensors

Let us now consider the case when the process is observed using multiple sensors that transmit data to a controller across an individual analog erasure channel. A separation principle to reduce the control design problem into the combination of an LQR control law and an estimation problem can once again be proven. Thus, the two-block design for the estimation problem asks the following question: what quantity should the sensors transmit such that the decoder is able to generate the optimal MMSE estimate of the state at every time step, given all the information the decoder has received till that time step. This problem is similar to the track-to-track fusion problem that has been studied since the 1980s and is still open for general cases (Chang et al. 1997). Suppose that at time k , the last successful transmission from sensor i happened at time $k_i \leq k$. The optimal estimate that the decoder can ever hope to achieve is the estimate of the state $x(k)$ based on all measurements from the sensor 1 till time k_1 , from sensor 2 till time k_2 , and so on. However, it is not known whether this estimate is achievable if the sensors are constrained to transmit real vectors with a bounded dimension. A fairly intuitive encoding scheme is if the sensors transmit the *local* estimates of the state based on their own measurements. However, it is known that the *global* estimate cannot, in general, be

obtained from local estimates because of the correlation introduced by the process noise. If erasure probabilities are zero, or if the process noise is not present, then the optimal encoding schemes are known. Another case for which the optimal encoding schemes are known is when the estimator sends back acknowledgments to the encoders.

Transmitting local estimates does, however, achieve optimal stability conditions as compared to the conditions obtained from the optimal (unknown) two-block design (Gupta et al. 2009b). As an example, the necessary and sufficient stability conditions for the two sensor cases are given by

$$\begin{aligned} p_1 \rho(A_1)^2 &< 1 \\ p_2 \rho(A_2)^2 &< 1 \\ p_1 p_2 \rho(A_3)^2 &< 1, \end{aligned}$$

where p_1 and p_2 are erasure probabilities from sensors 1 and 2, respectively, $\rho(A_1)$ is the spectral radius of the unobservable part of the matrix A from the second sensor, $\rho(A_2)$ is the spectral radius of the unobservable part of the matrix A from the first sensor, and $\rho(A_3)$ is the spectral radius of the observable part of the matrix A from both the sensors. The conditions are fairly intuitive. For instance, the first condition provides a bound on the rate of increase of modes for which only sensor 1 can provide information to the controller, in terms of how reliable the communication channel from the sensor 1 is.

Digital Noiseless Channels

Similar results as above can be derived for the digital noiseless channel model. For the digital noiseless channel model, it is easier to consider the system without either measurement or process noises (although results with such noises are available). Moreover, since quantization is inherently highly nonlinear, results such as separation between estimation and control are not available. Thus, encoders and controllers that optimize a cost function such as a quadratic performance metric are not available even for the single sensor or channel case. Most available results thus

discuss stabilizability conditions for a given data rate that the channels can support.

While early works used the one-block design framework to model the digital noiseless channel as introducing an additive white quantization noise, that framework obscures several crucial features of the channel. For instance, such an additive noise model suggests that at any bit rate, the process can be stabilized by a suitable controller. However, a simple argument can show that is not true. Consider a scalar process in which at time k , the controller knows that the state is within a set of length $l(k)$. Then, stabilization is possible only if $l(k)$ remains bounded as $k \rightarrow \infty$. Now, the evolution of $l(k)$ is governed by two processes: at every time step, this uncertainty can be (i) decreased by a factor of at most 2^m due to the data transmission across the channel and (ii) increased by a factor of a (where a is the process matrix governing the evolution of the state) due to the process evolution. This implies that for stabilization to be possible, the inequality $m \geq \log_2(a)$ must hold. Thus, the additive noise model is inherently wrong. Most results in the literature formalize this basic intuition above (Nair et al. 2007).

A Network of Communication Channels

For the case when there is only one sensor that transmits information to the controller across a network of communication channels connected in arbitrary topology, an analysis similar to that done for analog erasure channels can be performed (Tatikonda 2003). A max-flow min-cut like theorem again holds. The stability condition now becomes that for any cut-set

$$\sum R_j > \sum_{\text{all unstable eigenvalues}} \log_2(\lambda_i),$$

where $\sum R_j$ is the sum of data rates supported by the channels joining the source set to sink set for any cut-set and λ_i are the eigenvalues of the process matrix A . Note that the summation on the right hand side is only over the unstable eigenvalues, since no information needs to be transmitted about the modes that are stable in open loop.

Multiple Sensors

The case when multiple sensors transmit information across an individual digital noiseless channel to a controller can also be considered. For every sensor i , define a rate vector $\{R_{i_1}, R_{i_2}, \dots, R_{i_d}\}$ corresponding to the d modes of the system. If a mode j cannot be observed from the sensor i , set $R_{ij} = 0$. For stability, the condition

$$\sum_i R_{ij} \geq \max(0, \lambda_j),$$

for every mode j must be satisfied. All such rate vectors stabilize the system.

Summary and Future Directions

This entry provided a brief overview of some results available in the field of networked control systems. Although the area is seeing intense research activity, many problems remain open. For control across analog erasure channels, most existing results break down if a separation principle cannot be proved. Thus, for example, if control packets are also transmitted to the actuator across an analog erasure channel, the LQG optimal two-block design is unknown. There is some recent work on analyzing the stabilizability under such conditions (Gupta and Martins 2010), but the problem remains open in general. For digital noiseless channels, controllers that optimize some performance metric are largely unknown. Considering more general channel models is also an important research direction (Martins and Dahleh 2008; Sahai and Mitter 2006).

Cross-References

- [Averaging Algorithms and Consensus](#)
- [Distributed Estimation in Networks](#)
- [Estimation, Survey on](#)
- [Networked Control Systems: Estimation and Control Over Lossy Networks](#)
- [Oscillator Synchronization](#)

Bibliography

- Chang K-C, Saha RK, Bar-Shalom Y (1997) On optimal track-to-track fusion. *IEEE Trans Aerosp Electron Syst* AES-33:1271–1276
- Gupta V, Martins NC (2010) On stability in the presence of analog erasure channels between controller and actuator. *IEEE Trans Autom Control* 55(1): 175–179
- Gupta V, Dana AF, Hespanha J, Murray RM, Hassibi B (2009a) Data transmission over networks for estimation and control. *IEEE Trans Autom Control* 54(8):1807–1819
- Gupta V, Martins NC, Baras JS (2009b) Stabilization over erasure channels using multiple sensors. *IEEE Trans Autom Control* 54(7):1463–1476
- Martins NC, Dahleh M (2008) Feedback control in the presence of noisy channels: ‘bode-like’ fundamental limitations of performance. *IEEE Trans Autom Control* 53(7):1604–1615
- Nair GN, Fagnani F, Zampieri S, Evans RJ (2007) Feedback control under data rate constraints: an overview. *Proc IEEE* 95(1):108–137
- Sahai A, Mitter S (2006) The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link—Part I: scalar systems. *IEEE Trans Inf Theory* 52(8):3369–3395
- Tatikonda S (2003) Some scaling properties of large distributed control systems. In: 42th IEEE conference on decision and control, Maui, Dec 2003

Estimation for Random Sets

Ronald Mahler

Random Sets LLC, Eagan, MN, USA

Abstract

The *random set* (RS) concept generalizes that of a random vector. It permits the mathematical modeling of random systems that can be interpreted as *random patterns*. Algorithms based on RSs have been extensively employed in image processing. More recently, they have found application in multitarget detection and tracking and in the modeling and processing of human-mediated information sources. The purpose of this article is to briefly summarize the concepts, theory, and practical application of RSs.

Keywords

Random sets · Stochastic geometry · Point processes · Image processing · Multitarget tracking

Introduction

In ordinary signal processing, one models physical phenomena as “sources,” which generate “signals” obscured by random “noise.” The sources are to be extracted from the noise using optimal estimation algorithms. Random set (RS) theory was devised about 40 years ago by mathematicians who also wanted to construct optimal estimation algorithms. The “signals” and “noise” that they had in mind, however, were *geometric patterns in images*. The resulting theory, *stochastic geometry*, is the basis of the “morphological operators” commonly employed today in image-processing applications. It is also the basis for the theory of RSs. An important special case of RS theory, the theory of *random finite sets* (RFSs), addresses problems in which the patterns of interest consists of a finite number of points. It is the theoretical basis of many modern medical and other image-processing algorithms. In recent years, RFS theory has found application to the problem of detecting, localizing, and tracking unknown numbers of unknown, evasive point targets. Most recently and perhaps most surprisingly, RS theory provides a theoretically rigorous way of addressing “signals” that are *human-mediated*, such as natural-language statements and inference rules. The breadth of RS theory is suggested in the various chapters of Goutsias et al. (1997).

The purpose of this article is to summarize the RS and RFS theories and their applications. It is organized as follows: A simple example, Mathematics of Random Sets, Random Sets and Image Processing, Random Sets and Multitarget Processing, Random Sets and Human-Mediated Data, Summary and Future Directions, Cross-References, and Recommended Reading.

A Simple Example

To illustrate the concept of a RS, let us begin by examining a simple example: *locating stars in the nighttime sky*. We will proceed in successively more illustrative steps.

Locating a single non-dim star (estimating a random point). When we try to locate a star, we are trying to estimate its actual position—its “state” $\mathbf{x} = (\alpha_0, \theta_0)$ —in terms of its azimuth angle α_0 and elevation angle θ_0 . When the star is dim but not too dim, its apparent position will vary slightly. We can estimate its position by averaging many measurements—i.e., by applying a *point estimator*.

Locating a very dim star (estimating an RS with at most one element). Assume that the star is so dim that, when we see it, it might be just a momentary visual illusion. Before we can estimate its position, we must *first estimate whether or not it exists*. We must record not only its apparent position $\mathbf{z} = (\alpha, \theta)$ (if we see it) but its *apparent existence* ε , with $\varepsilon = 1$ (we saw it) or $\varepsilon = 0$ (we didn’t). Averaging ε over many observations, we get a number q between 0 and 1. If $q > \frac{1}{4}$ (say), we could declare that the star probably actually is a star; and then we could average the non-null observations to estimate its position.

Locating multiple stars (estimating an RFS). Suppose that we are trying to locate *all* of the stars in some patch of sky. In some cases, two dim stars may be so close that they are difficult to distinguish. We will then collect three kinds of measurements from them: $Z = \emptyset$ (didn’t see either star), $Z = \{(\alpha, \theta)\}$ (we saw one or the other), or $Z = \{(\alpha_1, \theta_1), (\alpha_2, \theta_2)\}$ (saw both). The total collected measurement in the patch of sky is a finite set $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_m\}$ of point measurements with $\mathbf{z}_j = (\theta_j, \alpha_j)$, where each $\mathbf{z}_i$ is random, where m is random, and where $m = 0$ corresponds to the null measurement $Z = \emptyset$.

Locating multiple stars in a quantized sky (estimation using imprecise measurements).

Suppose that, for computational reasons, the patch of sky must be quantized into a finite number of hexagonal-shaped cells, $c_1, \dots, c_M$. Then the measurement from any star is not a specific point $\mathbf{z}$ but instead the cell c that contains $\mathbf{z}$. The measurement c is *imprecise*—a randomly varying hexagonal cell c . There are two ways of thinking about the total measurement collection. First, it is a finite set $Z = \{c'_1, \dots, c'_m\} \subseteq \{c_1, \dots, c_M\}$ of cells. Second, it is the union $Z = c'_1 \cup \dots \cup c'_m$ of all of the observed cells—i.e., it is a *geometrical pattern*.

Locating multiple stars over an extended period of time (estimating multiple moving targets). As the night progresses, we must continually redetermine the existence and positions of each star—a process called *multitarget tracking*. We must also account for appearances and disappearances of the stars in the patch—i.e., for *target death and birth*.

Mathematics of Random Sets

The purpose of this section is to sketch the elements of the theory of random sets. It is organized as follows: General Theory of Random Sets, Random Finite Sets (Random Point Processes), and Stochastic Geometry. Of necessity, the material is less elementary than in later sections.

General Theory of Random Sets

Let $\mathfrak{Y}$ be a topological space—for example, an N -dimensional Euclidean space $\mathbb{R}^N$. The *power set* $2^{\mathfrak{Y}}$ of $\mathfrak{Y}$ is the class of all possible subsets $S \subseteq \mathfrak{Y}$. Any subclass of $2^{\mathfrak{Y}}$ is called a “hyperspace.” The “elements” or “points” of a hyperspace are thus actually subsets of some other space. For a hyperspace to be of interest, one must extend the topology on $\mathfrak{Y}$ to it. There are many possible topologies for hyperspaces (Michael 1950). The most well-studied is the *Fell-Matheron topology*, also called the “hit-and-miss” topology (Matheron 1975). It is applicable when $\mathfrak{Y}$ is Hausdorff, locally compact, and completely separable. It topologizes only

the hyperspace $\mathfrak{c}(2^{\mathfrak{Y}})$ of all *closed* subsets C of $\mathfrak{Y}$. In this case, a *random* (closed) set Θ is a measurable mapping from some probability space into $\mathfrak{c}(2^{\mathfrak{Y}})$.

The Fell-Matheron topology’s major strength is its relative simplicity. Let “ $\Pr(\mathcal{E})$ ” denote the probability of a probabilistic event $\mathcal{E}$. Then normally, the probability law of Θ would be described by a very abstract probability measure $p_{\Theta}(O) = \Pr(\Theta \in O)$. This measure must be defined on the Borel-measurable subsets $O \subseteq \mathfrak{c}(2^{\mathfrak{Y}})$, with respect to the Fell-Matheron topology, where O is itself a class of subsets of $\mathfrak{Y}$. However, define the *Choquet capacity functional* by $c_{\Theta}(G) = \Pr(\Theta \cap G \neq \emptyset)$ for all open subsets $G \subseteq \mathfrak{Y}$. Then the *Choquet-Matheron theorem* states that the probability law of Θ is completely described by the simpler, albeit nonadditive, measure $c_{\Theta}(G)$.

The theory of random sets has evolved into a substantial subgenre of statistical theory (Molchanov 2005). For estimation theory, the concept of the *expected value* $\mathbb{E}[\Theta]$ of a random set Θ is of particular interest. Most definitions of $\mathbb{E}[\Theta]$ are very abstract (Molchanov 2005, Chap. 2). In certain circumstances, however, more conventional-looking definitions are possible. Suppose that $\mathfrak{Y}$ is a Euclidean space and that $\mathfrak{c}(2^{\mathfrak{Y}})$ is restricted to $\mathfrak{K}(2^{\mathfrak{Y}})$, the *bounded, convex, closed subsets* of $\mathfrak{Y}$. If C, C' are two such subsets, their *Minkowski sum* is $C + C' = \{c + c' \mid c \in C, c' \in C'\}$. Endowed with this definition of addition, $\mathfrak{K}(2^{\mathfrak{Y}})$ can be homeomorphically and homomorphically embedded into a certain space of functions (Molchanov 2005, pp. 199–200). Denote this embedding by $C \mapsto \phi_C$. Then the expected value $\mathbb{E}[\Theta]$ of Θ , defined in terms of Minkowski addition, corresponds to the *conventional* expected value $\mathbb{E}[\phi_{\Theta}]$ of the random function ϕ_{Θ} .

Random Finite Sets (Random Point Processes)

Suppose that the $\mathfrak{c}(2^{\mathfrak{Y}})$ is restricted to $\mathfrak{f}(2^{\mathfrak{Y}})$, the class of *finite* subsets of $\mathfrak{Y}$. (In many formulations, $\mathfrak{f}(2^{\mathfrak{Y}})$ is taken to be the class of *locally finite* subsets of $\mathfrak{Y}$ —i.e., those whose intersection

with compact subsets is finite.) A *random finite set* (RFS) is a measurable mapping from a probability space into $\mathfrak{f}(2^{\mathfrak{Y}})$. An example: the field of twinkling stars in some patch of a night sky. RFS theory is a particular mathematical formulation of *point process theory* (Daley and Vere-Jones 1998; Snyder and Miller 1991; Stoyan et al. 1995).

A *Poisson RFS* Ψ is perhaps the simplest nontrivial example of a random point pattern. It is specified by a *spatial distribution* $s(\mathbf{x})$ and an *intensity* μ . At any given instant, the probability that there will be n points in the pattern is $p(n) = e^{-\mu} \mu^n / n!$ (the value of the Poisson distribution). The probability that one of these n points will be $\mathbf{y}$ is $s(\mathbf{y})$. The function $D_{\Psi}(\mathbf{y}) = \mu \cdot s(\mathbf{y})$ is called the *intensity function* of Ψ .

At any moment, the point pattern produced by Ψ is a finite set $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ of points $\mathbf{y}_1, \dots, \mathbf{y}_n$ in $\mathfrak{Y}$, where $n = 0, 1, \dots$ and where $X = \emptyset$ if $n = 0$. If $n = 0$ then X represents the hypothesis that no objects at all are present. If $n = 1$ then $Y = \{\mathbf{y}_1\}$ represents the hypothesis that a single object $\mathbf{y}_1$ is present. If $n = 2$ then $Y = \{\mathbf{y}_1, \mathbf{y}_2\}$ represents the hypothesis that there are two distinct objects $\mathbf{y}_1 \neq \mathbf{y}_2$. And so on.

The *probability distribution* of Ψ —i.e., the probability that Ψ will have $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ as an instantiation—is entirely determined by its intensity function $D_{\Psi}(\mathbf{y})$:

$$\begin{aligned} f_{\Psi}(Y) &= f_{\Psi}(\{\mathbf{y}_1, \dots, \mathbf{y}_n\}) \\ &= \frac{e^{-\mu}}{n!} \cdot D_{\Psi}(\mathbf{y}_1) \cdots D_{\Psi}(\mathbf{y}_n) \end{aligned}$$

Every suitably well-behaved RFS Ψ has a probability distribution $f_{\Psi}(Y)$ and an intensity function $D_{\Psi}(\mathbf{y})$ (a.k.a. *first-moment density*). A Poisson RFS is unique in that $f_{\Psi}(Y)$ is completely determined by $D_{\Psi}(\mathbf{y})$.

Conventional signal processing is often concerned with single-object random systems that have the form

$$\mathbf{Z} = \eta(\mathbf{x}) + \mathbf{V}$$

where $\mathbf{x}$ is the state of the system; $\eta(\mathbf{x})$ is the “signal” generated by the system; the zero-mean random vector $\mathbf{V}$ is the random “noise” associated

the sensor; and $\mathbf{Z}$ is the random measurement that is observed. The purpose of signal processing is to construct an estimate $\hat{\mathbf{x}}(\mathbf{z}_1, \dots, \mathbf{z}_k)$ of $\mathbf{x}$, using the information contained in one or more draws $\mathbf{z}_1, \dots, \mathbf{z}_k$ from the random variable $\mathbf{Z}$.

RFS theory is analogously concerned with random systems that have the form

$$\Sigma = \Upsilon(X) \cup \Omega$$

where a random finite point pattern $\Upsilon(X)$ is the “signal” generated by the point pattern X (which is an instantiation of a random point pattern Ξ); Ω is a random finite point “noise” pattern; Σ is the total random finite point pattern that has been observed; and “ $\cup$ ” denotes set-theoretic union. One goal of RFS theory is to devise algorithms that can construct an estimate $\hat{X}(Z_1, \dots, Z_k)$ of X , using multiple point patterns $Z_1, \dots, Z_k \subseteq \mathfrak{Y}$ drawn from Σ . One approximate approach is that of estimating only the first-moment density $D_{\Xi}(\mathbf{x})$ of Ξ .

Stochastic Geometry

Stochastic geometry addresses more complicated random patterns. An example: the field of twinkling stars in a *quantized* patch of the night sky, in which case the measurement is the union $c_1 \cup \dots \cup c_n$ of a finite number of hexagonally shaped cells.

This is one instance of a *germ-grain process* (Stoyan et al. 1995, pp. 59–64). Such a process is specified by two items: an RFS Ψ and a function $c_{\mathbf{y}}$ that associates with each $\mathbf{y}$ in $\mathfrak{Y}$ a closed subset $c_{\mathbf{y}} \subseteq \mathfrak{Z}$. For example, if $\mathfrak{Y} = \mathbb{R}^2$ is the real-valued plane, then $c_{\mathbf{y}}$ could be the disk of radius r centered at $\mathbf{y} = (x, y)$. Let $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ be a particular random draw from Ψ . The points $\mathbf{y}_1, \dots, \mathbf{y}_n$ are the “germs,” and $c_{\mathbf{y}_1}, \dots, c_{\mathbf{y}_n}$ are the “grains” of this random draw from the germ-grain process Θ . The total pattern in $\mathfrak{Y}$ is the union $c_{\mathbf{y}_1} \cup \dots \cup c_{\mathbf{y}_n}$ of the grains—a random draw from Θ . Germ-grain processes can be used to model many kinds of natural processes. One example is the distribution of graphite particles in a two-dimensional section of a piece of iron, in which case the $c_{\mathbf{y}}$ could be chosen to be line segments rather than disks.

Stochastic geometry is concerned with random binary images that have observation structures such as

$$\Theta = (S \cap \Delta) \cup \Omega$$

where S is a “signal” pattern; Δ is a random pattern that models obscurations; Ω is a random pattern that models clutter; and Θ is the total pattern that has been observed. A common simplifying assumption is that Ω and Δ^c are germ-grain processes. One goal of stochastic geometry is to devise algorithms that can construct an optimal estimate $\hat{S}(T_1, \dots, T_k)$ of S , using multiple patterns $T_1, \dots, T_k \subseteq \mathfrak{Y}$ drawn from Θ .

Random Sets and Image Processing

Both point process theory and stochastic geometry have found extensive application to image-processing applications. These are considered briefly in turn.

Stochastic Geometry and Image Processing

Stochastic geometry methods are based on the use of a “structuring element” B (a geometrical shape, such as a disk, sphere, or more complex structure) to modify an image.

The *dilation* of a set S by B is $S \oplus B$ where “ $\oplus$ ” is Minkowski addition (Stoyan et al. 1995). Dilation tends to fill in cavities and fissures in images. The *erosion* of S is $S \ominus B = (S^c \oplus B^c)^c$ where “ c ” indicates set-theoretic complement. Erosion tends to create and increase the size of cavities and fissures. *Morphological filters* are constructed from various combinations of dilation and erosion operators.

Suppose that a binary image $\Sigma = S$ has been degraded by some measurement process—for example, the process $\Theta = (S \cap \Delta) \cup \Omega$. Then *image restoration* refers to the construction of an estimate $\hat{S}(T)$ of the original image S from a single degraded image $\Theta = T$. The restoration operator $\hat{S}(T)$ is *optimal* if it can be shown to be optimally close to S , given some concept of closeness. The *symmetric difference*

$$T_1 \sqcup T_2 = (T_1 \cup T_2) - (T_1 \cap T_2)$$

is a commonly used method for measuring the dissimilarity of binary images. It can be used to construct measures of distance between random images. One such distance is

$$d(\Theta_1, \Theta_2) = \mathbb{E}[|\Theta_1 \sqcup \Theta_2|]$$

where $|S|$ denotes the size of the set S and $\mathbb{E}[A]$ is the expected value of the random number A . Other distances require some definition of the expected value $\mathbb{E}[\Theta]$ of a random set Θ . It has been shown that, under certain circumstances, certain morphological operators can be viewed as consistent maximum a posteriori (MAP) estimators of S (Goutsias et al. 1997, p. 97).

RFS Theory and Image Processing Positron-emission tomography (PET) is one example of the application of RFS theory. In PET, tissues of interest are suffused with a positron-emitting radioactive isotope. When a positron annihilates an electron in a suitable fashion, two photons are emitted in opposite directions. These photons are detected by sensors in a ring surrounding the radiating tissue. The location of the annihilation on the line can be estimated by calculating time difference-of-arrival.

Because of the physics of radioactive decay, the annihilations can be accurately modeled as a Poisson RFS Ψ . Since a Poisson RFS is completely determined by its intensity function $D_\Psi(\mathbf{x})$, it is natural to try to estimate $D_\Psi(\mathbf{x})$. This yields the spatial distribution $s_\Psi(\mathbf{y})$ of annihilations—which, in turn, is the basis of the PET image (Snyder and Miller 1991, pp. 115–119).

Random Sets and Multitarget Processing

The purpose of this section is to summarize the application of RFS theory to multitarget detection, tracking, and localization. An example: tracking the positions of stars in the night sky over an extended period of time.

Suppose that there are an unknown number n of targets with unknown states $\mathbf{x}_1, \dots, \mathbf{x}_n$. The state of the entire multitarget system is a finite set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ with $n \geq 0$. When interrogating a scene, many sensors (such as radars) produce a measurement of the form $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_m\}$ —i.e., a finite set of measurements. Some of these measurements are generated by background clutter Ω_k . Others are generated by the targets, with some targets possibly not having generated any. Mathematically speaking, Z is a random draw from an RFS Σ_k that can be decomposed as $\Sigma_k = \Upsilon(X_k) \cup \Omega_k$, where $\Upsilon(X_k)$ is the set of target-generated measurements.

Conventional Multitarget Detection and Tracking This is based on a “divide and conquer” strategy with three basic steps: *time-update*, *data association*, and *measurement-update*. At time t_k we have n “tracks” $\tau_1, \dots, \tau_n$ (hypothesized targets). In the time-update, an extended Kalman filter (EKF) is used to time-predict the tracks τ_i to *predicted tracks* τ_i^+ at the time t_{k+1} of the next measurement-set $Z_{k+1} = \{\mathbf{z}_1, \dots, \mathbf{z}_m\}$.

Given Z_{k+1} , we can construct the following *data-association hypothesis* H : for each $i = 1, \dots, n$, the predicted track τ_i^+ generated the detection $\mathbf{z}_{j_i}$ for some index j_i ; or, alternatively, this track was not detected at all. If we remove from Z_{k+1} all of the $\mathbf{z}_{j_1}, \dots, \mathbf{z}_{j_n}$, the remaining measurements are interpreted as being either clutter or as having been generated by new targets. Enumerating all possible association hypotheses (which is a combinatorially complex procedure), we end up with a “hypothesis table” $H_1, \dots, H_\nu$.

Given H_i , let $\mathbf{z}_{j_i}$ be the measurement that is hypothesized to have been generated by predicted track τ_i^+ . Then the measurement-update step of an EKF is used to construct a measurement-updated track τ_{i,j_i} from τ_i^+ and $\mathbf{z}_{j_i}$. Attached to each H_i is a *hypothesis probability* p_i —the probability that the particular hypothesis H_i is the correct one. The hypothesis with largest p_i yields the multitarget estimate $\hat{X} = \{\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_{\hat{n}}\}$.

RFS Multitarget Detection and Tracking In the place of tracks and hypothesis tables, this uses

multitarget state-sets and *multitarget probability distributions*. In place of the conventional time-update, data association, and measurement-update, it uses a *recursive Bayes filter*. A random multitarget state-set is an RFS $\Xi_{k|k}$ whose points are target states. A multitarget probability distribution is the probability distribution $f(X_k|Z_{1:k}) = f_{\Xi_{k|k}}(X)$ of the RFS $\Xi_{k|k}$, where $Z_{1:k} : Z_1, \dots, Z_k$ is the time-sequence of measurement-sets at time t_k .

RFS Time-Update The Bayes filter time-update step $f(X_k|Z_{1:k}) \rightarrow f(X_{k+1}|Z_{1:k})$ requires a *multitarget Markov transition function* $f(X_{k+1}|X_k)$. It is the probability that the multitarget system will have multitarget state-set X_{k+1} at time t_{k+1} , if it had multitarget state-set X_k at time t_k . It takes into account all pertinent characteristics of the targets: individual target motion, target appearance, target disappearance, environmental constraints, etc. It is explicitly constructed from an *RFS multitarget motion model* using a *multitarget integro-differential calculus*.

RFS Measurement-Update The Bayes filter measurement-update step $f(X_{k+1}|Z_{1:k}) \rightarrow f(X_{k+1}|Z_{1:k+1})$ is just Bayes’ rule. It requires a *multitarget likelihood function* $f(Z_{k+1}|X_{k+1})$ —the likelihood that a measurement-set Z will be generated, if a system of targets with state-set X is present. It takes into account all pertinent characteristics of the sensor(s): sensor noise, fields of view and obscurations, probabilities of detection, false alarms, and/or clutter. It is explicitly constructed from an *RFS measurement model* using multitarget calculus.

RFS State-Estimation Determination of the number n and states $\mathbf{x}_1, \dots, \mathbf{x}_n$ of the targets is accomplished using a *Bayes-optimal multitarget state estimator*. The idea is to determine the X_{k+1} that maximizes $f(X_{k+1}|Z_{1:k+1})$ in some sense.

Approximate Multitarget RFS Filters The multitarget Bayes filter is, in general, computationally intractable. Central to the RFS approach is a toolbox of techniques—including

the multitarget calculus—designed to produce *statistically principled* approximate multitarget filters. The two most well-studied are the *probability hypothesis density* (PHD) filter and its generalization, the *cardinalized PHD* (CPHD) filter. In such filters, $f(X_k|Z_{1:k})$ is replaced by the first-moment density $D(\mathbf{x}_k|Z_{1:k})$ of $\Xi_{k|k}$. These filters have been shown to be faster and perform as well as or better than conventional approaches in some applications.

Labeled RFSs and RFS Trackers In order to maintain connected target trajectories through time, the PHD and CPHD filters use heuristic rules to attach unique identifiers (“track labels”) to targets. The *labeled RFS theory* of Vo and Vo (2013) provides a theoretically rigorous approach to track labeling. In this theory, target states have the form $\mathbf{x} = (\mathbf{u}, \ell)$ where $\mathbf{u}$ is the kinematic state and ℓ is a uniquely identifying unknown target label. The *labeled* multitarget Bayes filter propagates the distribution $f(X_k|Z_{1:k}) = f_{\Xi_{k|k}}(X)$ of the LRFS $\Xi_{k|k}$. When Bayes-optimal multitarget state estimates $\hat{X}_k$ are extracted from the $f(X_k|Z_{1:k})$, the target trajectory with label ℓ consists of those $(\mathbf{u}_k, \ell)$ with $k \geq 1$ for which $(\mathbf{u}_k, \ell) \in \hat{X}_k$.

The Generalized Labeled Multi-Bernoulli (GLMB) Filter The Kalman filter is an exact closed-form solution of the single-target recursive Bayes filter. That is: suppose that the single-target Markov transition functions $f(\mathbf{x}_{k+1}|\mathbf{x}_k)$, single-target likelihood functions $f(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})$, and single-target initial distribution $f(\mathbf{x}_0)$ are linear Gaussian. Then $f(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = N_{P_{k|k-1}}(\mathbf{x}_k - \mathbf{x}_{k|k-1})$ and $f(\mathbf{x}_k|\mathbf{z}_{1:k}) = N_{P_{k|k}}(\mathbf{x}_k - \mathbf{x}_{k|k})$ for $k \geq 1$, where $(\mathbf{x}_{k|k}, P_{k|k})$, $(\mathbf{x}_{k|k-1}, P_{k|k-1})$ are the Kalman filter’s states and covariance matrices.

Vo and Vo (2013) demonstrated that this reasoning generalizes to multitarget tracking. Let $f(X_{k+1}|X_k)$ be the transition density for the “standard” multitarget motion model and $f(Z_{k+1}|X_{k+1})$ the likelihood function for the “standard” multitarget measurement model. Then the labeled multitarget Bayes filter is algebraically closed with respect to the

parametrized family of “GLMB distributions.” The GLMB filter results when GLMB parameters are propagated instead of GLMB distributions. It can be tractably implemented using Gibbs sampling techniques (Vo et al. 2017) and can simultaneously track over a million targets in two dimensions in significant clutter, using off-the-shelf computing equipment (Beard et al. 2018).

Random Sets and Human-Mediated Data

Natural-language statements and inference rules have already been mentioned as examples of human-mediated information. *Expert-systems theory* was introduced in part to address situations—such as this—that involve uncertainties other than randomness. Expert-system methodologies include *fuzzy set theory*, the *Dempster-Shafer* (D-S) *theory of uncertain evidence*, and *rule-based inference*. RS theory provides solid Bayesian foundations for them and allows human-mediated data to be processed using standard Bayesian estimation techniques. The purpose of this section is to briefly summarize this aspect of the RS approach.

The relationships between expert-systems theory and random set theory were first established by researchers such as Orlov (1978), Höhle (1981), Nguyen (1978), and Goodman and Nguyen (1985). At a relatively early stage, it was recognized that random set theory provided a potential means of unifying much of expert-systems theory (Goodman and Nguyen 1985; Kruse et al. 1991).

A conventional sensor measurement at time t_k is typically represented as $\mathbf{Z}_k = \eta(\mathbf{x}_k) + \mathbf{V}_k$ —equivalently formulated as a *likelihood function* $f(\mathbf{z}_k|\mathbf{x}_k)$. It is conventional to think of $\mathbf{z}_k$ as the actual “measurement” and of $f(\mathbf{z}_k|\mathbf{x}_k)$ as the full description of the uncertainty associated with it. In actuality, $\mathbf{z}$ is just a *mathematical model* $\mathbf{z}_{\zeta_k}$ of some *real-world measurement* ζ_k . Thus the likelihood actually has the form $f(\zeta_k|\mathbf{x}_k) = f(\mathbf{z}_{\zeta_k}|\mathbf{x}_k)$.

This observation assumes crucial importance when one considers human-mediated data. Consider the simple natural-language statement

$\zeta = \text{“The target is near the tower”}$

where the *tower* is a landmark, located at a known position (x_0, y_0) , and where the term “near” is assumed to have the following specific meaning: (x, y) is near (x_0, y_0) means that $(x, y) \in T_5$ where T_5 is a disk of radius 5 m, centered at (x_0, y_0) . If $\mathbf{z} = (x, y)$ is the actual measurement of the target’s position, then ζ is *equivalent to the formula* $\mathbf{z} \in T_5$. Since $\mathbf{z}$ is just one possible draw from $\mathbf{Z}_k$, we can say that ζ —or equivalently, T_5 —is actually a constraint on the underlying measurement process: $\mathbf{Z}_k \in T_5$.

Because the word “near” is rather vague, we could just as well say that $\mathbf{z} \in T_5$ is the best choice, with confidence $w_5 = 0.7$; that $\mathbf{z} \in T_4$ is the next-best choice, with confidence $w_4 = 0.2$; and that $\mathbf{z} \in T_6$ is the least best, with confidence $w_6 = 0.1$. Let Θ be the random subset of $\mathcal{Z}$ defined by $\Pr(\Theta = T_i) = w_i$ for $i = 4, 5, 6$. In this case, ζ is equivalent to the *random constraint*

$$\mathbf{Z}_k \in \Theta$$

The probability

$$\begin{aligned}\rho_k(\Theta|\mathbf{x}) &= \Pr(\eta(\mathbf{x}_k) + \mathbf{V}_k \in \Theta) \\ &= \Pr(\mathbf{Z} \in \Theta | \mathbf{X} = \mathbf{x})\end{aligned}$$

is called a *generalized likelihood function* (GLF). GLFs can be constructed for more complex natural-language statements, for inference rules, and more. Using their GLF representations, such “nontraditional measurements” can be processed using single- and multi-object recursive Bayes filters and their approximations. As a consequence, it can be shown that fuzzy-logic, the D-S theory, and rule-based inference can be subsumed within a single Bayesian-probabilistic paradigm.

Summary and Future Directions

In the engineering world, the theory of random sets has been associated primarily with certain specialized image processing applications, such as morphological filters and tomographic

imaging. It has more recently found application in fields such as multitarget tracking and in expert-systems theory. All of these fields of application remain areas of active research.

Cross-References

- Estimation, Survey on
- Extended Kalman Filters

Recommended Reading

Molchanov (2005) provides a definitive exposition of the general theory of random sets. Two excellent references for stochastic geometry are Stoyan et al. (1995) and Barndorff-Nielsen and Lieshout (1999). The books by Kingman (1993) and by Daley and Vere-Jones (1998) are good introductions to point process theory. The application of point process theory and stochastic geometry to image-processing is addressed in, respectively, Snyder and Miller (1991) and Stoyan et al. (1995). The application of RFSs to multitarget estimation is addressed in the tutorials Mahler (2013, 2014) and the books Mahler (2007, 2014). Introductions to the application of random sets to expert systems can be found in Kruse et al. (1991), Mahler (2007), Chaps. 3–6, and Mahler (2014) (Chap. 22).

Bibliography

- Barndorff-Nielsen O, van Lieshout M (1999) Stochastic geometry: likelihood and computation. Chapman&Hall/CRC, Boca Raton
- Beard M, Vo B-T, Vo B-N (2018) A solution for large-scale multi-object tracking. <http://arxiv.org/abs/1804.06622v1>
- Daley D, Vere-Jones D (1998) An introduction to the theory of point processes, 1st edn. Springer, New York
- Goodman I, Nguyen H (1985) Uncertainty Models for knowledge based systems. North-Holland, Amsterdam
- Goutsias J, Mahler R, Nguyen H (eds) (1997) Random sets: theory and applications. Springer, New York

- Höhle U (1981) A mathematical theory of uncertainty: fuzzy experiments and their realizations. In: Yager R (ed) Recent developments in fuzzy sets and possibility theory. Permagon Press, pp 344–355. Oxford, UK
- Kingman J (1993) Poisson processes. Oxford University Press, London
- Kruse R, Schwencke E, Heinsohn J (1991) Uncertainty and vagueness in knowledge-based systems. Springer, New York
- Mahler R (2004) ‘Statistics 101’ for multisensor, multitarget data fusion. *IEEE Trans Aerospace Electron Syst Mag Part 2 Tutorials* 19(1):53–64
- Mahler R (2007) Statistical multisource-multitarget information fusion. Artech House, Norwood
- Mahler R (2013) ‘Statistics 102’ for multisensor-multitarget tracking. *IEEE J Spec Top Sign Proc* 7(3):376–389
- Mahler R (2014) Advances in statistical multisource-multitarget information fusion. Artech House, Norwood
- Matheron G (1975) Random sets and integral geometry. Wiley, New York
- Michael E (1950) Topologies on spaces of subsets. *Trans Am Math Soc* 71:152–182
- Molchanov I (2005) Theory of random sets. Springer, London
- Nguyen H (1978) On random sets and belief functions. *J Math Anal Appl* 65:531–542
- Orlov A (1978) Fuzzy and random sets. *Prikladnoi Mnogomerni Statisticheskii Analys*, Moscow
- Snyder D, Miller M (1991) Random point processes in time and space, 2nd edn. Springer, New York
- Stoyan D, Kendall W, Mecke J (1995) Stochastic geometry and its applications, 2nd edn. Wiley, New York
- Vo B-T, Vo B-N (2013) Labeled random finite sets and multi-object conjugate priors. *IEEE Trans Sign Proc* 61(13):3460–3475
- Vo B-N, Vo B-T, Hoang HG (2017) An efficient implementation of the generalized labeled multi-Bernoulli filter. *IEEE Trans Sign Proc* 65(8):1975–1987

Estimation, Survey on

Luigi Chisci¹ and Alfonso Farina²

¹Dipartimento di Ingegneria dell’Informazione, Università di Firenze, Firenze, Italy

²Selex ES, Roma, Italy

Abstract

This entry discusses the history and describes the multitude of methods and applications of

this important branch of stochastic process theory.

Keywords

Linear stochastic filtering · Markov step processes · Maximum likelihood estimation · Riccati equation · Stratonovich-Kushner equation

Estimation is the process of inferring the value of an unknown given quantity of interest from noisy, direct or indirect, observations of such a quantity. Due to its great practical relevance, estimation has a long history and an enormous variety of applications in all fields of engineering and science. A certainly incomplete list of possible application domains of estimation includes the following: statistics (Lehmann and Casella 1998; Bard 1974; Wertz 1978; Ghosh et al. 1997; Koch 1999; Tsybakov 2009), telecommunication systems (Snyder 1968; Van Trees 1971; Sage and Melsa 1971; Schonhoff and Giordano 2006), signal and image processing (Kay 1993; Itakura 1971; Wakita 1973; Poor 1994; Elliott et al. 2008; Barkat 2005; Levy 2008; Najim 2008; Tuncer and Friedlander 2009; Woods and Radewan 1977; Biemond et al. 1983; Kim and Woods 1998), aerospace engineering (McGee and Schmidt 1985), tracking (Farina and Studer 1985, 1986; Blackman and Popoli 1999; Bar-Shalom et al. 2001, 2013; Bar-Shalom and Fortmann 1988), navigation (Schmidt 1966; Farrell and Barth 1999; Grewal et al. 2001; Smith et al. 1986; Dissanayake et al. 2001; Durrant-Whyte and Bailey 2006a,b; Thrun et al. 2006; Mullane et al. 2011), control systems (Kalman 1960a; Joseph and Tou 1961; Athans 1971; Anderson and Moore 1979; Maybeck 1979, 1982; Stengel 1994; Söderström 1994; Goodwin et al. 2005), econometrics (Zellner 1971; Pindyck and Roberts 1974; Aoki 1987), geophysics (e.g., seismic deconvolution) (Flinn et al. 1967; Bayless and Brigham 1970; Mendel 1977, 1983, 1990), oceanography (Ghil and Malanotte-Rizzoli 1991; Evensen 1994a), weather forecasting (McGarty

1971; Evensen 1994b, 2007), environmental engineering (Nachazel 1993; Dochain and Vanrolleghem 2001; Heemink and Segers 2002), demographic systems (Leibungudt et al. 1983), automotive systems (Stephant et al. 2004; Barbarisi et al. 2006), failure detection (Willsky 1976; Mangoubi 1998; Chen and Patton 1999), power systems (Debs and Larson 1970; Toyoda et al. 1970; Miller and Lewis 1971; Monticelli 1999; Abur and Gómez Espósito 2004), nuclear engineering (Sage and Masters 1967; Roman et al. 1971; Venerus and Bullock 1970; Robinson 1963), biomedical engineering (Stark 1968; Snyder 1970; Bekey 1973), pattern recognition (Ho and Agrawala 1968; Lainiotis 1972; Andrews 1972), social networks (Snijders et al. 2012), etc.

Chapter Organization

The rest of the chapter is organized as follows. Section “[Historical Overview on Estimation](#)” will provide a historical overview on estimation. The next section will discuss applications of estimation. Connections between estimation and information theories will be explored in the subsequent section. Finally, the section “[Conclusions and Future Trends](#)” will conclude the chapter by discussing future trends in estimation. An extensive list of references is also provided.

Historical Overview on Estimation

A possibly incomplete, list of the major achievements on estimation theory and applications is reported in Table 1. The entries of the table, sorted in chronological order, provide for each contribution the name of the inventor (or inventors), the date, and a short description with main bibliographical references.

Probably the first important application of estimation dates back to the beginning of the nineteenth century whenever least-squares estimation (LSE), invented by Gauss in 1795 (Gauss 1995; Legendre 1810), was successfully exploited in astronomy for predicting planet

orbits (Gauss 1806). Least-squares estimation follows a deterministic approach by minimizing the sum of squares of residuals defined as differences between observed data and model-predicted estimates. A subsequently introduced statistical approach is maximum likelihood estimation (MLE), popularized by R. A. Fisher between 1912 and 1922 (Fisher 1912, 1922, 1925). MLE consists of finding the estimate of the unknown quantity of interest as the value that maximizes the so-called likelihood function, defined as the conditional probability density function of the observed data given the quantity to be estimated. In intuitive terms, MLE maximizes the agreement of the estimate with the observed data. Whenever the observation noise is assumed Gaussian (Kim and Shevlyakov 2008; Park et al. 2013), MLE coincides with LSE.

While estimation problems had been addressed for several centuries, it was not until the 1940s that a systematic theory of estimation started to be established, mainly relying on the foundations of the modern theory of probability (Kolmogorov 1933). Actually, the roots of probability theory can be traced back to the calculus of combinatorics (the Stomachion puzzle invented by Archimedes (Netz and Noel 2011)) in the third century B.C. and to the gambling theory (work of Cardano, Pascal, de Fermat, Huygens) in the sixteenth–seventeenth centuries.

Differently from the previous work devoted to the estimation of constant parameters, in the period 1940–1960 the attention was mainly shifted toward the estimation of signals. In particular, Wiener in 1940 (Wiener 1949) and Kolmogorov in 1941 (Kolmogorov 1941) formulated and solved the problem of linear minimum mean-square error (MMSE) estimation of continuous-time and, respectively, discrete-time stationary random signals. In the late 1940s and in the 1950s, Wiener-Kolmogorov’s theory was extended and generalized in many directions exploiting both time-domain and frequency-domain approaches. At the beginning of the 1960s Rudolf E. Kálmán made pioneering contributions to estimation by providing the mathematical foundations of the modern theory

Estimation, Survey on, Table 1 Major developments on estimation

Archimedes	Third century B.C.	Combinatorics (Netz and Noel 2011) as the basis of probability
G. Cardano, B. Pascal, P. de Fermat, C. Huygens	Sixteenth–seventeenth centuries	Roots of the theory of probability (Devlin 2008)
J. F. Riccati	1722–1723	Differential Riccati equation (Riccati 1722, 1723), subsequently exploited in the theory of linear stochastic filtering
T. Bayes	1763	Bayes' formula on conditional probability (Bayes 1763; McGrayne 2011)
C. F. Gauss, A. M. Legendre	1795–1810	Least-squares estimation and its applications to the prediction of planet orbits (Gauss 1806, 1995; Legendre 1810)
P. S. Laplace	1814	Theory of probability (Laplace 1814)
R. A. Fisher	1912–1922	Maximum likelihood estimation (Fisher 1912, 1922, 1925)
A. N. Kolmogorov	1933	Modern theory of probability (Kolmogorov 1933)
N. Wiener	1940	Minimum mean-square error estimation of continuous-time stationary random signals (Wiener 1949)
A. N. Kolmogorov	1941	Minimum mean-square error estimation of discrete-time stationary random signals (Kolmogorov 1941)
H. Cramér, C. R. Rao	1945	Theoretical lower bound on the covariance of estimators (Rao 1945; Cramér 1946)
S. Ulam, J. von Neumann, N. Metropolis, E. Fermi	1946–1949	Monte Carlo method (Ulam et al. 1947; Metropolis and Ulam 1949; Ulam 1952; Los Alamos Scientific Laboratory 1966)
J. Sklansky, T. R. Benedict, G. W. Bordner, H. R. Simpson, S. R. Neal	1957–1967	$\alpha - \beta$ and $\alpha - \beta - \gamma$ filters (Sklansky 1957; Benedict and Bordner 1962; Simpson 1963; Neal 1967; Painter et al. 1990)
R. L. Stratonovich, H. J. Kushner	1959–1964	Bayesian approach to stochastic nonlinear filtering of continuous-time systems, i.e., Stratonovich-Kushner equation for the evolution of the state conditional probability density (Stratonovich 1959, 1960; Kushner 1962, 1967; Jazwinski 1970)
R. E. Kalman	1960	Linear filtering and prediction for discrete-time systems (Kalman 1960b)
R. E. Kalman	1961	Observability of linear dynamical systems (Kalman 1960a)
R. E. Kalman, R. S. Bucy	1961	Linear filtering and prediction for continuous-time systems (Kalman and Bucy 1961)
A. E. Bryson, M. Frazier, H. E. Rauch, F. Tung, C. T. Striebel, D. Q. Mayne, J. S. Meditch, D. C. Fraser, L. E. Zachrisson, B. D. O. Anderson, etc.	Since 1963	Smoothing of linear and nonlinear systems (Bryson and Frazier 1963; Rauch 1963; Rauch et al. 1965; Mayne 1966; Meditch 1967; Zachrisson 1969; Anderson and Chirarattananon 1972)
D. G. Luenberger	1964	State observer for a linear system (Luenberger 1964)
Y. C. Ho, R. C. K. Lee	1964	Bayesian approach to recursive nonlinear estimation for discrete-time systems (Ho and Lee 1964)
W. M. Wonham	1965	Optimal filtering for Markov step processes (Wonham 1965)
A. H. Jazwinski	1966	Bayesian approach to stochastic nonlinear filtering for continuous-time stochastic systems with discrete-time observations (Jazwinski 1966)

(continued)

Estimation, Survey on, Table 1 (continued)

Archimedes	Third century B.C.	Combinatorics (Netz and Noel 2011) as the basis of probability
S. F. Schmidt	1966	Extended Kalman filter and its application for the manned lunar missions (Schmidt 1966)
P. L. Falb, A. V. Balakrishnan, J. L. Lions, S. G. Tzafestas, J. M. Nightingale, H. J. Kushner, J. S. Meditch, etc.	Since 1967	State estimation for infinite-dimensional (e.g., distributed parameter, partial differential equation (PDE), delay) systems (Falb 1967; Balakrishnan and Lions 1967; Kwakernaak 1967; Tzafestas and Nightingale 1968; Kushner 1970; Meditch 1971)
T. Kailath	1968	Principle of orthogonality and innovation approach to estimation (Kailath 1968, 1970; Kailath and Frost 1968; Frost and Kailath 1971; Kailath et al. 2000)
A. H. Jazwinski, B. Rawlings, etc.	Since 1968	Limited memory (receding-horizon, moving-horizon) state estimation with constraints (Jazwinski 1968; Rao et al. 2001, 2003; Alessandri et al. 2005, 2008)
F. C. Schweppe, D. P. Bertsekas, I. B. Rhodes, M. Milanese, etc.	Since 1968	Set-membership recursive state estimation with systems with unknown but bounded noises (Schweppe 1968; Bertsekas and Rhodes 1971; Milanese and Belforte 1982; Milanese and Vicino 1993; Combettes 1993; Vicino and Zappa 1996; Chisci et al. 1996; Alamo et al. 2005)
J. E. Potter, G. Golub, F. Schmidt, P. G. Kaminski, A. E. Bryson, A. Andrews, G. J. Bierman, M. Morf, T. Kailath, etc.	1968–1975	Square-root filtering (Potter and Stern 1963; Golub 1965; Andrews 1968; Schmidt 1970; Kaminski and Bryson 1972; Bierman 1974, 1977; Morf and Kailath 1975)
C. W. Helstrom	1969	Quantum estimation (Helstrom 1969, 1976)
D. L. Alspach, H. W. Sorenson	1970–1972	Gaussian-sum filters for nonlinear and/or non-Gaussian systems (Sorenson and Alspach 1970, 1971; Alspach and Sorenson 1972)
T. Kailath, M. Morf, G. Sidhu	1973–1974	Fast Chandrasekhar-type algorithms for recursive state estimation of stationary linear systems (Kailath 1973; Morf et al. 1974)
A. Segall	1976	Recursive estimation from point processes (Segall 1976)
J. W. Woods and C. Radewan	1977	Kalman filter in two dimensions (Woods and Radewan 1977) for image processing
J. H. Taylor	1979	Cramér-Rao lower bound (CRLB) for recursive state estimation with no process noise (Taylor 1979)
D. Reid	1979	Multiple Hypothesis Tracking (MHT) filter for multi-target tracking (Reid 1979)
L. Servi, Y. Ho	1981	Optimal filtering for linear systems with uniformly distributed measurement noise (Servi and Ho 1981)
V. E. Benes	1981	Exact finite-dimensional optimal MMSE filter for a class of nonlinear systems (Benes 1981)
H. V. Poor, D. Looze, J. Daragh, S. Verdú, M. J. Grimbale, etc.	1981–1988	Robust (e.g., H_∞) filtering (Poor and Looze 1981; Daragh and Looze 1984; Verdú and Poor 1984; Grimbale 1988; Hassibi et al. 1999; Simon 2006)
V. J. Aidala, S. E. Hammel	1983	Bearings-only tracking (Aidala and Hammel 1983; Farina 1999)
F. E. Daum	1986	Extension of the Benes filter to a more general class of nonlinear systems (Daum 1986)

(continued)

Estimation, Survey on, Table 1 (continued)

Archimedes	Third century B.C.	Combinatorics (Netz and Noel 2011) as the basis of probability
L. Dai and others	Since 1987	State estimation for linear descriptor (singular, implicit) stochastic systems (Dai 1987, 1989; Nikoukhah et al. 1992; Chisci and Zappa 1992)
N. J. Gordon, D. J. Salmond, 1993 A. M. F. Smith		Particle (sequential Monte Carlo) filter (Gordon et al. 1993; Doucet et al. 2001; Ristic et al. 2004)
K. C. Chou, A. S. Willsky, 1994 A. Benveniste		Multiscale Kalman filter (Chou et al. 1994)
G. Evensen	1994	Ensemble Kalman filter for data assimilation in meteorology and oceanography (Evensen 1994b, 2007)
R. P. S. Mahler	1994	Random set filtering (Mahler 1994, 2007a; Ristic et al. 2013)
S. J. Julier, J. K. Uhlmann, 1995 H. Durrant-Whyte		Unscented Kalman filter (Julier et al. 1995; Julier and Uhlmann 2004)
A. Germani et al.	Since 1996	Polynomial extended Kalman filter for nonlinear and/or non-Gaussian systems (Carravetta et al. 1996; Germani et al. 2005)
P. Tichavsky, C. H. 1998 Muravchik, A. Nehorai		Posterior Cramér-Rao lower bound (PCRLB) for recursive state estimation (Tichavsky et al. 1998; van Trees and Bell 2007)
R. Mahler	2003, 2007	Probability hypothesis density (PHD) and cardinalized PHD (CPHD) filters (Mahler 2003, 2007b; Vo and Ma 1996; Vo et al. 2007; Ristic 2013)
A.G. Ramm	2005	Estimation of random fields (Ramm 2005)
M. Hernandez, A. Farina, B. 2006 Ristic		PCRLB for tracking in the case of detection probability less than one and false alarm probability greater than zero (Hernandez et al. 2006)
Olfati-Saber and others	Since 2007	Consensus filters (Olfati-Saber et al. 2007; Calafiore and Abrate 2009; Xiao et al. 2005; Alriksson and Rantzer 2006; Olfati-Saber 2007; Kamgarpour and Tomlin 2007; Stankovic et al. 2009; Battistelli et al. 2011, 2012, 2013; Battistelli and Chisci 2014) for networked estimation

based on state-variable representations. In particular, Kálmán solved the linear MMSE filtering and prediction problems both in discrete-time (Kalman 1960b) and in continuous-time (Kalman and Bucy 1961); the resulting optimal estimator was named after him, Kalman filter (KF). As a further contribution, Kalman also singled out the key technical conditions, i.e., observability and controllability, for which the resulting optimal estimator turns out to be stable. Kalman's work went well beyond earlier contributions of A. Kolmogorov, N. Wiener, and their followers ("frequency-domain" approach) by means of a general state-space approach. From the theoretical viewpoint, the KF is an optimal

estimator, in a wide sense, of the state of a linear dynamical system from noisy measurements; specifically it is the optimal MMSE estimator in the Gaussian case (e.g., for normally distributed noises and initial state) and the best linear unbiased estimator irrespective of the noise and initial state distributions. From the practical viewpoint, the KF enjoys the desirable properties of being linear and acting recursively, step-by-step, on a noise-contaminated data stream. This allows for cheap real-time implementation on digital computers. Further, the universality of "state-variable representations" allows almost any estimation problem to be included in the KF framework. For these reasons, the KF is,

and continues to be, an extremely effective and easy-to-implement tool for a great variety of practical tasks, e.g., to detect signals in noise or to estimate unmeasurable quantities from accessible observables. Due to the generality of the state estimation problem, which actually encompasses parameter and signal estimation as special cases, the literature on estimation since 1960 till today has been mostly concentrated on extensions and generalizations of Kalman's work in several directions. Considerable efforts, motivated by the ubiquitous presence of nonlinearities in practical estimation problems, have been devoted to nonlinear and/or non-Gaussian filtering, starting from the seminal papers of Stratonovich (1959, 1960) and Kushner (1962, 1967) for continuous-time systems, Ho and Lee (1964) for discrete-time systems, and Jazwinski (1966) for continuous-time systems with discrete-time observations. In these papers, state estimation is cast in a probabilistic (Bayesian) framework as the problem of evolving in time the state conditional probability density given observations (Jazwinski 1970). Work on nonlinear filtering has produced over the years several nonlinear state estimation algorithms, e.g., the extended Kalman filter (EKF) (Schmidt 1966), the unscented Kalman filter (UKF) (Julier et al. 1995; Julier and Uhlmann 2004), the Gaussian-sum filter (Sorenson and Alspach 1970, 1971; Alspach and Sorenson 1972), the sequential Monte Carlo (also called particle) filter (SMCF) (Gordon et al. 1993; Doucet et al. 2001; Ristic et al. 2004), and the ensemble Kalman filter (EnKF) (Evensen 1994a, b, 2007) which have been, and are still now, successfully employed in various application domains. In particular, the SMCF and EnKF are stochastic simulation algorithms taking inspiration from the work in the 1940s on the Monte Carlo method (Metropolis and Ulam 1949) which has recently got renewed interest thanks to the tremendous advances in computing technology. A thorough review on nonlinear filtering can be found, e.g., in Daum (2005) and Crisan and Rozovskii (2011).

Other interesting areas of investigation have concerned smoothing (Bryson and Frazier 1963), robust filtering for systems subject to

modeling uncertainties (Poor and Looze 1981), and state estimation for infinite-dimensional (i.e., distributed parameter and/or delay) systems (Balakrishnan and Lions 1967). Further, a lot of attention has been devoted to the implementation of the KF, specifically square-root filtering (Potter and Stern 1963) for improved numerical robustness and fast KF algorithms (Kailath 1973; Morf et al. 1974) for enhancing computational efficiency. Worth of mention is the work over the years on theoretical bounds on the estimation performance originated from the seminal papers of Rao (1945) and Cramér (1946) on the lower bound of the MSE for parameter estimation and subsequently extended in Tichavsky et al. (1998) to nonlinear filtering and in Hernandez et al. (2006) to more realistic estimation problems with possible missed and/or false measurements. An extensive review of this work on Bayesian bounds for estimation, nonlinear filtering, and tracking can be found in van Trees and Bell (2007). A brief review of the earlier (until 1974) state of art in estimation can be found in Lainiotis (1974).

Applications

Astronomy

The problem of making estimates and predictions on the basis of noisy observations originally attracted the attention many centuries ago in the field of astronomy. In particular, the first attempt to provide an optimal estimate, i.e., such that a certain measure of the estimation error be minimized, was due to Galileo Galilei that, in his *Dialogue on the Two World Chief Systems* (1632) (Galilei 1632), suggested, as a possible criterion for estimating the position of Tycho Brahe's supernova, the estimate that required the "minimum amendments and smallest corrections" to the data. Later, C. F. Gauss mathematically specified this criterion by introducing in 1795 the least-squares method (Gauss 1806, 1995; Legendre 1810) which was successfully applied in 1801 to predict the location of the asteroid Ceres. This asteroid, originally discovered by the Italian astronomer Giuseppe Piazzi on January 1, 1801, and then

lost in the glare of the sun, was in fact recovered 1 year later by the Hungarian astronomer F. X. von Zach exploiting the least-squares predictions of Ceres' position provided by Gauss.

Statistics

Starting from the work of Fisher in the 1920s (Fisher 1912, 1922, 1925), maximum likelihood estimation has been extensively employed in statistics for estimating the parameters of statistical models (Lehmann and Casella 1998; Bard 1974; Wertz 1978; Ghosh et al. 1997; Koch 1999; Tsybakov 2009).

Telecommunications and Signal/Image Processing

Wiener-Kolmogorov's theory on signal estimation, developed in the period 1940–1960 and originally conceived by Wiener during the Second World War for predicting aircraft trajectories in order to direct the antiaircraft fire, subsequently originated many applications in telecommunications and signal/image processing (Van Trees 1971; Kay 1993; Itakura 1971; Wakita 1973; Poor 1994; Elliott et al. 2008; Barkat 2005; Levy 2008; Najim 2008; Tuncer and Friedlander 2009; Woods and Radewan 1977; Biemond et al. 1983; Kim and Woods 1998). For instance, Wiener filters have been successfully applied to linear prediction, acoustic echo cancellation, signal restoration, and image/video de-noising. But it was the discovery of the Kalman filter in 1960 that revolutionized estimation by providing an effective and powerful tool for the solution of any, static or dynamic, stationary or adaptive, linear estimation problem. A recently conducted, and probably non-exhaustive, search has detected the presence of over 16,000 patents related to the “Kalman filter,” spreading over all areas of engineering and over a period of more than 50 years. What is astonishing is that even nowadays, more than 50 years after its discovery, one can see the continuous appearance of lots of new patents and scientific papers presenting novel applications and/or novel extensions in many directions (e.g., to nonlinear filtering) of the KF. Since 1992 the number of patents registered every year and related to the KF follows an exponential law.

Space Navigation and Aerospace Applications

The first important application of the Kalman filter was in the NASA (*National Aeronautic and Space Administration*) space program. As reported in a NASA technical report (McGee and Schmidt 1985), Kalman presented his new ideas while visiting Stanley F. Schmidt at the NASA Ames Research Center in 1960, and this meeting stimulated the use of the KF during the Apollo program (in particular, in the guidance system of Saturn V during Apollo 11 flight to the Moon), and, furthermore, in the NASA Space Shuttle and in Navy submarines and unmanned aerospace vehicles and weapons, such as cruise missiles. Further, to cope with the nonlinearity of the space navigation problem and the small word length of the onboard computer, the extended Kalman filter for nonlinear systems and square-root filter implementations for enhanced numerical robustness have been developed as part of the NASA's Apollo program. The aerospace field was only the first of a long and continuously expanding list of application domains where the Kalman filter and its nonlinear generalizations have found widespread and beneficial use.

Control Systems and System Identification

The work on Kalman filtering (Kalman 1960b; Kalman and Bucy 1961) had also a significant impact on control system design and implementation. In Kalman (1960a) duality between estimation and control was pointed out, in that for a certain class of control and estimation problems one can solve the control (estimation) problem for a given dynamical system by resorting to a corresponding estimation (control) problem for a suitably defined dual system. In particular, the Kalman filter has been shown to be dual of the linear-quadratic (LQ) regulator, and the two dual techniques constitute the linear-quadratic-Gaussian (LQG) (Joseph and Tou 1961) regulator. The latter consists of an LQ regulator feeding back in a linear way the state estimate provided by a Kalman filter, which can be independently designed in view of the separation principle. The KF as well as LSE and MLE techniques are also widely used in system identification

(Ljung 1999; Söderström and Stoica 1989) for both parameter estimation and output prediction purposes.

Tracking

One of the major application areas for estimation is tracking (Farina and Studer 1985, 1986; Blackman and Popoli 1999; Bar-Shalom et al. 2001, 2013; Bar-Shalom and Fortmann 1988), i.e., the task of following the motion of moving objects (e.g., aircrafts, ships, ground vehicles, persons, animals) given noisy measurements of kinematic variables from remote sensors (e.g., radar, sonar, video cameras, wireless sensors, etc.). The development of the Wiener filter in the 1940s was actually motivated by radar tracking of aircraft for automatic control of antiaircraft guns. Such filters began to be used in the 1950s whenever computers were integrated with radar systems, and then in the 1960s more advanced and better performing Kalman filters came into use. Still today it can be said that the Kalman filter and its nonlinear generalizations (e.g., EKF (Schmidt 1966), UKF (Julier and Uhlmann 2004), and particle filter (Gordon et al. 1993)) represent the workhorses of tracking and sensor fusion. Tracking, however, is usually much more complicated than a simple state estimation problem due to the presence of false measurements (clutter) and multiple objects in the surveillance region of interest, as well as for the uncertainty about the origin of measurements. This requires to use, besides filtering algorithms, smart techniques for object detection as well as for association between detected objects and measurements. The problem of joint target tracking and classification has also been formulated as a hybrid state estimation problem and addressed in a number of papers (see, e.g., Smeth and Ristic (2004) and the references therein).

Econometrics

State and parameter estimation have been widely used in econometrics (Aoki 1987) for analyzing and/or predicting financial time series (e.g., stock prices, interest rates, unemployment rates, volatility etc.).

Geophysics

Wiener and Kalman filtering techniques are employed in reflection seismology for estimating the unknown earth reflectivity function given noisy measurements of the seismic wavelet's echoes recorded by a geophone. This estimation problem, known as seismic deconvolution (Mendel 1977, 1983, 1990), has been successfully exploited, e.g., for oil exploration.

Data Assimilation for Weather Forecasting and Oceanography

Another interesting application of estimation theory is data assimilation (Ghil and Malanotte-Rizzoli 1991) which consists of incorporating noisy observations into a computer simulation model of a real system. Data assimilation has widespread use especially in weather forecasting and oceanography. A large-scale state-space model is typically obtained from the physical system model, expressed in terms of partial differential equations (PDEs), by means of a suitable spatial discretization technique so that data assimilation is cast into a state estimation problem. To deal with the huge dimensionality of the resulting state vector, appropriate filtering techniques with reduced computational load have been suitably developed (Evensen 2007).

Global Navigation Satellite Systems

Global Navigation Satellite Systems (GNSSs), such as GPS put into service in 1993 by the US Department of Defense, provide nowadays a commercially diffused technology exploited by millions of users all over the world for navigation purposes, wherein the Kalman filter plays a key role (Bar-Shalom et al. 2001). In fact, the Kalman filter not only is employed in the core of the GNSS to estimate the trajectories of all the satellites, the drifts and rates of all system clocks, and hundreds of parameters related to atmospheric propagation delay, but also any GNSS receiver uses a nonlinear Kalman filter, e.g., EKF, in order to estimate its own position and velocity along with the bias and drift of its own clock with respect to the GNSS time.

Robotic Navigation (SLAM)

Recursive state estimation is commonly employed in mobile robotics (Thrun et al. 2006) in order to on-line estimate the robot pose, location and velocity, and, sometimes, also the location and features of the surrounding objects in the environment exploiting measurements provided by onboard sensors; the overall joint estimation problem is referred to as SLAM (simultaneous localization and mapping) (Smith et al. 1986; Dissanayake et al. 2001; Durrant-Whyte and Bailey 2006a, b; Thrun et al. 2006; Mullane et al. 2011).

Automotive Systems

Several automotive applications of the Kalman filter, or of its nonlinear variants, are reported in the literature for the estimation of various quantities of interest that cannot be directly measured, e.g., roll angle, sideslip angle, road-tire forces, heading direction, vehicle mass, state of charge of the battery (Barbarisi et al. 2006), etc. In general, one of the major applications of state estimation is the development of virtual sensors, i.e., estimation algorithms for physical variables of interest, that cannot be directly measured for technical and/or economic reasons (Stephant et al. 2004).

Miscellaneous Applications

Other areas where estimation has found numerous applications include electric power systems (Debs and Larson 1970; Toyoda et al. 1970; Miller and Lewis 1971; Monticelli 1999; Abur and Gómez Espósito 2004), nuclear reactors (Sage and Masters 1967; Roman et al. 1971; Venerus and Bullock 1970; Robinson 1963), biomedical engineering (Stark 1968; Snyder 1970; Bekey 1973), pattern recognition (Ho and Agrawala 1968; Lainiotis 1972; Andrews 1972), and many others.

Connection Between Information and Estimation Theories

In this section, the link between two fundamental quantities in information theory and estimation

theory, i.e., the mutual information (MI) and respectively the minimum mean-square error (MMSE), is investigated. In particular, a strikingly simple but very general relationship can be established between the MI of the input and the output of an additive Gaussian channel and the MMSE in estimating the input given the output, regardless of the input distribution (Guo et al. 2005). Although this functional relation holds for general settings of the Gaussian channel (e.g., both discrete-time and continuous-time, possibly vector, channels), in order to avoid the heavy mathematical preliminaries needed to treat rigorously the general problem, two simple scalar cases, a static and a (continuous-time) dynamic one, will be discussed just to highlight the main concept.

Static Scalar Case

Consider two scalar real-valued random variables, x and y , related by

$$y = \sqrt{\sigma}x + v \quad (1)$$

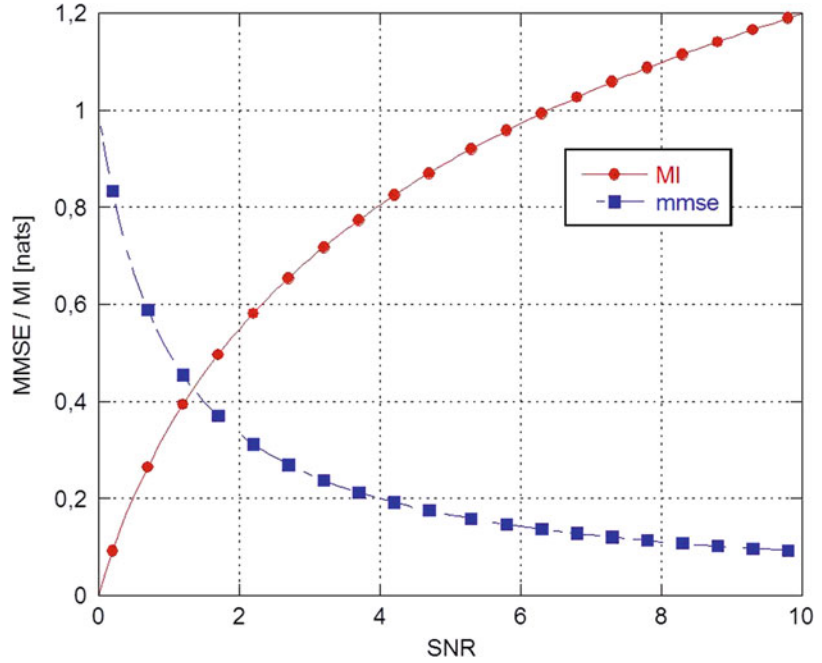
where v , the measurement noise, is a standard Gaussian random variable independent of x and σ can be regarded as the gain in the output signal-to-noise ratio (SNR) due to the channel. By considering the MI between x and y as a function of σ , i.e., $I(\sigma) = I(x, \sqrt{\sigma}x + v)$, it can be shown that the following relation holds (Guo et al. 2005):

$$\frac{d}{d\sigma} I(\sigma) = \frac{1}{2} E \left[(x - \hat{x}(\sigma))^2 \right] \quad (2)$$

where $\hat{x}(\sigma) = E[x|\sqrt{\sigma}x + v]$ is the minimum mean-square error estimate of x given y . Figure 1 displays the behavior of both MI, in natural logarithmic units of information (nats), and MMSE versus SNR.

As mentioned in Guo et al. (2005), the above information-estimation relationship (2) has found a number of applications, e.g., in nonlinear filtering, in multiuser detection, in power allocation over parallel Gaussian channels, in the proof of Shannon's entropy power inequality and its

Estimation, Survey on,
Fig. 1 MI and MMSE
versus SNR



generalizations, as well as in the treatment of the capacity region of several multiuser channels.

1970; Mayer-Wolf and Zakai 1983)

Linear Dynamic Continuous-Time Case

While in the static case the MI is assumed to be a function of the SNR, in the dynamic case it is of great interest to investigate the relationship between the MI and the MMSE as a function of time.

Consider the following first-order (scalar) linear Gaussian continuous-time stochastic dynamical system:

$$\begin{aligned} dx_t &= ax_t dt + dw_t \\ dy_t &= \sqrt{\sigma} x_t dt + dv_t \end{aligned} \quad (3)$$

where a is a real-valued constant while w_t and v_t are independent standard Brownian motion processes that represent the process and, respectively, measurement noises. Defining by $x_0^t \triangleq \{x_s, 0 \leq s \leq t\}$ the collection of all states up to time t and analogously $y_0^t \triangleq \{y_s, 0 \leq s \leq t\}$ for the channel outputs (i.e., measurements) and considering the MI between x_0^t and y_0^t as a function of time t , i.e., $I(t) = I(x_0^t, y_0^t)$, it can be shown that (Duncan

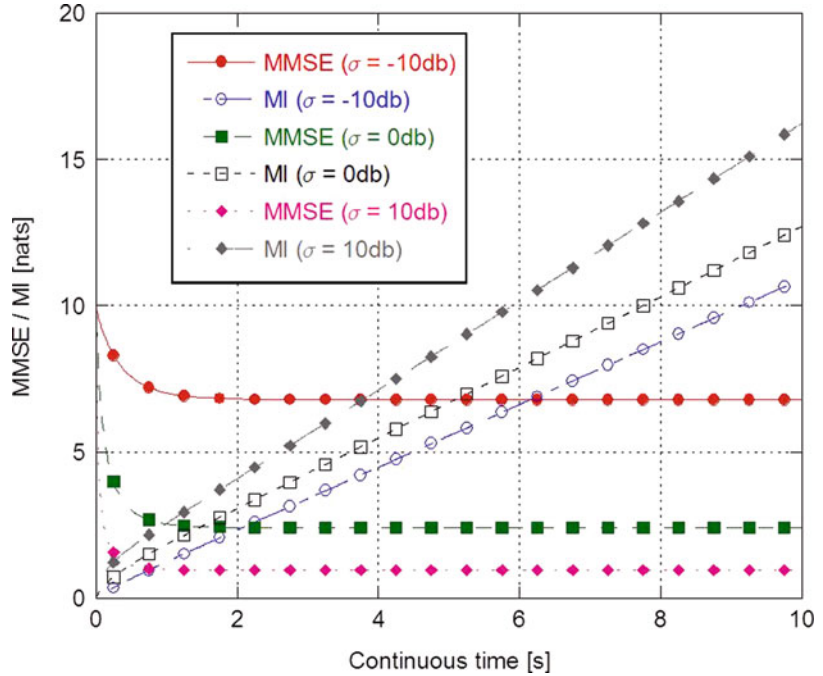
$$\frac{d}{dt} I(t) = \frac{\sigma}{2} E \left[(x_t - \hat{x}_t)^2 \right] \quad (4)$$

where $\hat{x}_t = E[x_t | y_0^t]$ is the minimum mean-square error estimate of the state x_t given all the channel outputs up to time t , i.e., y_0^t . Figure 2 depicts the time behavior of both MI and MMSE for several values of σ and $a = 1$.

Conclusions and Future Trends

Despite the long history of estimation and the huge amount of work on several theoretical and practical aspects of estimation, there is still a lot of research investigation to be done in several directions. Among the many new future trends, networked estimation and quantum estimation (briefly overviewed in the subsequent parts of this section) certainly deserve special attention due to the growing interest on wireless sensor networks and, respectively, quantum computing.

Estimation, Survey on,
Fig. 2 MI and MMSE for
different values of σ
(−10 dB, 0 dB, +10 dB)
and $a = 1$



Networked Information Fusion and Estimation

Information or data fusion is about combining, or fusing, information or data from multiple sources to provide knowledge that is not evident from a single source (Farina and Studer 1986; Bar-Shalom et al. 2013). In 1986, an effort to standardize the terminology related to data fusion began and the JDL (Joint Directors of Laboratories) data fusion working group was established. The result of that effort was the conception of a process model for data fusion and a data fusion lexicon (Hall and Llinas 1997; Blasch et al. 2012). Information and data fusion are mainly supported by sensor networks which present the following advantages over a single sensor:

- Can be deployed over wide regions
- Provide diverse characteristics/viewing angles of the observed phenomenon
- Are more robust to failures
- Gather more data that, once fused, provide a more complete picture of the observed phenomenon

- Allow better geographical coverage, i.e., wider area and less terrain obstructions.

Sensor network architectures can be centralized, hierarchical (with or without feedback), and distributed (peer-to-peer). Today's trend for many monitoring and decision-making tasks is to exploit large-scale networks of low-cost and low-energy consumption devices with sensing, communication, and processing capabilities. For scalability issues, such networks should operate in a fully distributed (peer-to-peer) fashion, i.e., with no centralized coordination, so as to achieve in each node a global estimation/decision objective through localized processing only.

The attainment of this goal actually requires several issues to be addressed like:

- Spatial and temporal sensor alignment
- Scalable fusion
- Robustness with respect to data incest (or double counting), i.e., repeated use of the same information
- Handling data latency (e.g., out-of-sequence measurements/estimates)
- Communication bandwidth limitations

In particular, to counteract data incest the so-called *covariance intersection* (Julier and Uhlmann 1997) robust fusion approach has been proposed to guarantee, at the price of some conservatism, consistency of the fused estimate when combining estimates from different nodes with unknown correlations. For scalable fusion, a consensus approach (Olfati-Saber et al. 2007) can be undertaken. This allows to carry out a global (i.e., over the whole network) processing task by iterating local processing steps among neighboring nodes.

Several consensus algorithms have been proposed for distributed parameter (Calafiore and Abrate 2009) or state (Xiao et al. 2005; Alriksson and Rantzer 2006; Olfati-Saber 2007; Kamgarpour and Tomlin 2007; Stankovic et al. 2009) estimation. Recently, Battistelli and Chisci (2014) introduced a generalized consensus on probability densities which opens up the possibility to perform in a fully distributed and scalable way any Bayesian estimation task over a sensor network. As by-products, this approach allowed to derive consensus Kalman filters with guaranteed stability under minimal requirements of system observability and network connectivity (Battistelli et al. 2011, 2012; Battistelli and Chisci 2014), consensus nonlinear filters (Battistelli et al. 2012), and a consensus CPHD filter for distributed multitarget tracking (Battistelli et al. 2013). Despite these interesting preliminary results, networked estimation is still a very active research area with many open problems related to energy efficiency, estimation performance optimality, robustness with respect to delays and/or data losses, etc.

Quantum Estimation

Quantum estimation theory consists of a generalization of the classical estimation theory in terms of quantum mechanics. As a matter of fact, the statistical theory can be seen as a particular case of the more general quantum theory (Helstrom 1969, 1976). Quantum mechanics presents practical applications in several fields of technology (Personick 1971) such as, the use of quantum number generators in place of the classical random number generators. Moreover,

manipulating the energy states of the cesium atoms, it is possible to suppress the quantum noise levels and consequently improve the accuracy of atomic clocks. Quantum mechanics can also be exploited to solve optimization problems, giving sometimes optimization algorithms that are faster than conventional ones. For instance, McGeoch and Wang (2013) provided an experimental study of algorithms based on quantum annealing. Interestingly, the results of McGeoch and Wang (2013) have shown that this approach allows to obtain better solutions with respect to those found with conventional software solvers. In quantum mechanics, also the Kalman filter has found its proper form, as the quantum Kalman filter. In Iida et al. (2010) the quantum Kalman filter is applied to an optical cavity composed of mirrors and crystals inside, which interacts with a probe laser. In particular, a form of a quantum stochastic differential equation can be written for such a system so as to design the algorithm that updates the estimates of the system variables on the basis of the measurement outcome of the system.

Cross-References

- [Averaging Algorithms and Consensus](#)
- [Bounds on Estimation](#)
- [Consensus of Complex Multi-agent Systems](#)
- [Data Association](#)
- [Estimation and Control over Networks](#)
- [Estimation for Random Sets](#)
- [Extended Kalman Filters](#)
- [Kalman Filters](#)
- [Moving Horizon Estimation](#)
- [Networked Control Systems: Estimation and Control over Lossy Networks](#)
- [Nonlinear Filters](#)
- [Observers for Nonlinear Systems](#)
- [Observers in Linear Systems Theory](#)
- [Particle Filters](#)

Acknowledgments The authors warmly recognize the contribution of S. Fortunati (University of Pisa, Italy), L. Pallotta (University of Napoli, Italy), and G. Battistelli (University of Firenze, Italy).

Bibliography

- Abur A, Gómez Espósito A (2004) Power system state estimation – theory and implementation. Marcel Dekker, New York
- Aidala VJ, Hammel SE (1983) Utilization of modified polar coordinates for bearings-only tracking. *IEEE Trans Autom Control* 28(3):283–294
- Alamo T, Bravo JM, Camacho EF (2005) Guaranteed state estimation by zonotopes. *Automatica* 41(6):1035–1043
- Alessandri A, Baglietto M, Battistelli G (2005) Receding-horizon estimation for switching discrete-time linear systems. *IEEE Trans Autom Control* 50(11):1736–1748
- Alessandri A, Baglietto M, Battistelli G (2008) Moving-horizon state estimation for nonlinear discrete-time systems: new stability results and approximation schemes. *Automatica* 44(7):1753–1765
- Alriksson P, Rantzer A (2006) Distributed Kalman filtering using weighted averaging. In: *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto
- Alsopch DL, Sorenson HW (1972) Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Trans Autom Control* 17(4):439–448
- Anderson BDO, Chirarattananon S (1972) New linear smoothing formulas. *IEEE Trans Autom Control* 17(1):160–161
- Anderson BDO, Moore JB (1979) *Optimal filtering*. Prentice Hall, Englewood Cliffs
- Andrews A (1968) A square root formulation of the Kalman covariance equations. *AIAA J* 6(6):1165–1166
- Andrews HC (1972) *Introduction to mathematical techniques in pattern recognition*. Wiley, New York
- Aoki M (1987) *State space modeling of time-series*. Springer, Berlin
- Athans M (1971) An optimal allocation and guidance laws for linear interception and rendezvous problems. *IEEE Trans Aerosp Electron Syst* 7(5):843–853
- Balakrishnan AV, Lions JL (1967) State estimation for infinite-dimensional systems. *J Comput Syst Sci* 1(4):391–403
- Barbarisi O, Vasca F, Glielmo L (2006) State of charge Kalman filter estimator for automotive batteries. *Control Eng Pract* 14(3):267–275
- Bard Y (1974) *Nonlinear parameter estimation*. Academic, New York
- Barkat M (2005) *Signal detection and estimation*. Artech House, Boston
- Bar-Shalom Y, Fortmann TE (1988) *Tracking and data association*. Academic, Boston
- Bar-Shalom Y, Li X, Kirubarajan T (2001) *Estimation with applications to tracking and navigation*. Wiley, New York
- Bar-Shalom Y, Willett P, Tian X (2013) *Tracking and data fusion: a handbook of algorithms*. YBS Publishing Storrs, CT
- Battistelli G, Chisci L, Morrocchi S, Papi F (2011) An information-theoretic approach to distributed state estimation. In: *Proceedings of the 18th IFAC World Congress*, Milan, pp 12477–12482
- Battistelli G, Chisci L, Mugnai G, Farina A, Graziano A (2012) Consensus-based algorithms for distributed filtering. In: *Proceedings of the 51st IEEE Control and Decision Conference*, Maui, pp 794–799
- Battistelli G, Chisci L (2014) Kullback-Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. *Automatica* 50:707–718
- Battistelli G, Chisci L, Fantacci C, Farina A, Graziano A (2013) Consensus CPHD filter for distributed multitarget tracking. *IEEE J Sel Topics Signal Process* 7(3):508–520
- Bayes T (1763) Essay towards solving a problem in the doctrine of chances. *Philos Trans R Soc Lond* 53:370–418. doi:10.1098/rstl.1763.0053
- Bayless JW, Brigham EO (1970) Application of the Kalman filter. *Geophysics* 35(1):2–23
- Bekey GA (1973) Parameter estimation in biological systems: a survey. In: *Proceedings of the 3rd IFAC Symposium Identification and System Parameter Estimation*, Delft, pp 1123–1130
- Benedict TR, Bordner GW (1962) Synthesis of an optimal set of radar track-while-scan smoothing equations. *IRE Trans Autom Control* 7(4):27–32
- Benes VE (1981) Exact finite-dimensional filters for certain diffusions with non linear drift. *Stochastics* 5(1):65–92
- Bertsekas DP, Rhodes IB (1971) Recursive state estimation for a set-membership description of uncertainty. *IEEE Trans Autom Control* 16(2):117–128
- Biernand J, Rieske J, Gerbrands JJ (1983) A fast Kalman filter for images degraded by both blur and noise. *IEEE Trans Acoust Speech Signal Process* 31(5):1248–1256
- Bierman GJ (1974) Sequential square root filtering and smoothing of discrete linear systems. *Automatica* 10(2):147–158
- Bierman GJ (1977) *Factorization methods for discrete sequential estimation*. Academic, New York
- Blackman S, Popoli R (1999) *Design and analysis of modern tracking systems*. Artech House, Norwood
- Blasch EP, Lambert DA, Valin P, Kokar MM, Llinas J, Das S, Chong C, Shahbazian E (2012) High level information fusion (HLIF): survey of models, issues, and grand challenges. *IEEE Aerosp Electron Syst Mag* 27(9):4–20
- Bryson AE, Frazier M (1963) Smoothing for linear and nonlinear systems, TDR 63-119, Aero System Division. Wright-Patterson Air Force Base, Ohio
- Calafiore GC, Abrate F (2009) Distributed linear estimation over sensor networks. *Int J Control* 82(5):868–882
- Carravetta F, Germani A, Raimondi M (1996) Polynomial filtering for linear discrete-time non-Gaussian systems. *SIAM J Control Optim* 34(5):1666–1690
- Chen J, Patton RJ (1999) *Robust model-based fault diagnosis for dynamic systems*. Kluwer, Boston

- Chisci L, Zappa G (1992) Square-root Kalman filtering of descriptor systems. *Syst Control Lett* 19(4):325–334
- Chisci L, Garulli A, Zappa G (1996) Recursive state bounding by parallelotopes. *Automatica* 32(7):1049–1055
- Chou KC, Willsky AS, Benveniste A (1994) Multiscale recursive estimation, data fusion, and regularization. *IEEE Trans Autom Control* 39(3):464–478
- Combettes P (1993) The foundations of set-theoretic estimation. *Proc IEEE* 81(2):182–208
- Cramér H (1946) *Mathematical methods of statistics*. University Press, Princeton
- Crisan D, Rozovski B (eds) (2011) *The Oxford handbook of nonlinear filtering*. Oxford University Press, Oxford
- Dai L (1987) State estimation schemes in singular systems. In: *Preprints 10th IFAC World Congress*, Munich, vol 9, pp 211–215
- Dai L (1989) Filtering and LQG problems for discrete-time stochastic singular systems. *IEEE Trans Autom Control* 34(10):1105–1108
- Darragh J, Looze D (1984) Noncausal minimax linear state estimation for systems with uncertain second-order statistics. *IEEE Trans Autom Control* 29(6):555–557
- Daum FE (1986) Exact finite dimensional nonlinear filters. *IEEE Trans Autom Control* 31(7):616–622
- Daum F (2005) Nonlinear filters: beyond the Kalman filter. *IEEE Aerosp Electron Syst Mag* 20(8):57–69
- Debs AS, Larson RE (1970) A dynamic estimator for tracking the state of a power system. *IEEE Trans Power Appar Syst* 89(7):1670–1678
- Devlin K (2008) *The unfinished game: Pascal, Fermat and the seventeenth-century letter that made the world modern*. Basic Books, New York. Copyright (C) Keith Devlin
- Dissanayake G, Newman P, Clark S, Durrant-Whyte H, Csorba M (2001) A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans Robot Autom* 17(3):229–241
- Dochain D, Vanrolleghem P (2001) *Dynamical modelling and estimation of wastewater treatment processes*. IWA Publishing, London
- Doucet A, de Freitas N, Gordon N (eds) (2001) *Sequential Monte Carlo methods in practice*. Springer, New York
- Duncan TE (1970) On the calculation of mutual information. *SIAM J Appl Math* 19(1):215–220
- Durrant-Whyte H, Bailey T (2006a) Simultaneous localization and mapping: Part I. *IEEE Robot Autom Mag* 13(2):99–108
- Durrant-Whyte H, Bailey T (2006b) Simultaneous localization and mapping: Part II. *IEEE Robot Autom Mag* 13(3):110–117
- Elliott RJ, Aggoun L, Moore JB (2008) *Hidden Markov models: estimation and control*. Springer, New York. (first edition in 1995)
- Evensen G (1994a) Inverse methods and data assimilation in nonlinear ocean models. *Physica D* 77:108–129
- Evensen G (1994b) Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J Geophys Res* 99(10):143–162
- Evensen G (2007) *Data assimilation – the ensemble Kalman filter*. Springer, Berlin
- Falb PL (1967) Infinite-dimensional filtering; the Kalman-Bucy filter in Hilbert space. *Inf Control* 11(1):102–137
- Farina A (1999) Target tracking with bearings-only measurements. *Signal Process* 78(1):61–78
- Farina A, Studer FA (1985) *Radar data processing. Volume 1 – introduction and tracking*, Editor P. Bowron. Research Studies Press, Letchworth/Wiley, New York (Translated in Russian (Radio I Sviaz, Moscow 1993) and in Chinese. China Defense Publishing House, 1988)
- Farina A, Studer FA (1986) *Radar data processing. Volume 2 – advanced topics and applications*, Editor P. Bowron. Research Studies Press, Letchworth/Wiley, New York (In Chinese, China Defense Publishing House, 1992)
- Farrell JA, Barth M (1999) *The global positioning system and inertial navigation*. McGraw-Hill, New York
- Fisher RA (1912) On an absolute criterion for fitting frequency curves. *Messenger Math* 41(5):155–160
- Fisher RA (1922) On the mathematical foundations of theoretical statistics. *Philos Trans R Soc Lond Ser A* 222:309–368
- Fisher RA (1925) Theory of statistical estimation. *Math Proc Camb Philos Soc* 22(5):700–725
- Flinn EA, Robinson EA, Treitel S (1967) Special issue on the MIT geophysical analysis group reports. *Geophysics* 32:411–525
- Frost PA, Kailath T (1971) An innovations approach to least-squares estimation, Part III: nonlinear estimation in white Gaussian noise. *IEEE Trans Autom Control* 16(3):217–226
- Galilei G (1632) *Dialogue concerning the two chief world systems* (Translated in English by S. Drake (Foreword of A. Einstein), University of California Press, Berkeley, 1953)
- Gauss CF (1806) *Theoria motus corporum coelestium in sectionibus conicis solem ambientum* (Translated in English by C.H. Davis, Reprinted as *Theory of motion of the heavenly bodies*, Dover, New York, 1963)
- Gauss CF (1995) *Theoria combinationis observationum erroribus minimis obnoxiae* (Translated by G.W. Stewart, *Classics in Applied Mathematics*), vol 11. SIAM, Philadelphia
- Germani A, Manes C, Palumbo P (2005) Polynomial extended Kalman filter. *IEEE Trans Autom Control* 50(12):2059–2064
- Ghil M, Malanotte-Rizzoli P (1991) Data assimilation in meteorology and oceanography. *Adv Geophys* 33:141–265
- Ghosh M, Mukhopadhyay N, Sen PK (1997) *Sequential estimation*. Wiley, New York
- Golub GH (1965) Numerical methods for solving linear least squares problems. *Numerische Mathematik* 7(3):206–216
- Goodwin GC, Seron MM, De Doná JA (2005) *Constrained control and estimation*. Springer, London

- Gordon NJ, Salmond DJ, Smith AFM (1993) Novel approach to nonlinear/non Gaussian Bayesian state estimation. *IEE Proc F* 140(2):107–113
- Grewal MS, Weill LR, Andrews AP (2001) Global positioning systems, inertial navigation and integration. Wiley, New York
- Grimble MJ (1988) H_∞ design of optimal linear filters. In: Byrnes C, Martin C, Sacks R (eds) Linear circuits, systems and signal processing: theory and applications. North Holland, Amsterdam, pp 535–540
- Guo D, Shamai S, Verdú S (2005) Mutual information and minimum mean-square error in Gaussian channels. *IEEE Trans Inf Theory* 51(4):1261–1282
- Hall DL, Llinas J (1997) An introduction to multisensor data fusion. *Proc IEEE* 85(1):6–23
- Hassibi B, Sayed AH, Kailath T (1999) Indefinite-quadratic estimation and control. SIAM, Philadelphia
- Heemink AW, Segers AJ (2002) Modeling and prediction of environmental data in space and time using Kalman filtering. *Stoch Environ Res Risk Assess* 16: 225–240
- Helstrom CW (1969) Quantum detection and estimation theory. *J Stat Phys* 1(2):231–252
- Helstrom CW (1976) Quantum detection and estimation theory. Academic, New York
- Hernandez M, Farina A, Ristic B (2006) PCRLB for tracking in cluttered environments: measurement sequence conditioning approach. *IEEE Trans Aerosp Electron Syst* 42(2):680–704
- Ho YC, Agrawala AK (1968) On pattern classification algorithms – introduction and survey. *Proc IEEE* 56(12):2101–2113
- Ho YC, Lee RCK (1964) A Bayesian approach to problems in stochastic estimation and control. *IEEE Trans Autom Control* 9(4):333–339
- Iida S, Ohki F, Yamamoto N (2010) Robust quantum Kalman filtering under the phase uncertainty of the probe-laser. In: IEEE international symposium on computer-aided control system design, Yokohama, pp 749–754
- Itakura F (1971) Extraction of feature parameters of speech by statistical methods. In: Proceedings of the 8th Symposium Speech Information Processing, Sendai, pp II-5.1–II-5.12
- Jazwinski AH (1966) Filtering for nonlinear dynamical systems. *IEEE Trans Autom Control* 11(4):765–766
- Jazwinski AH (1968) Limited memory optimal filtering. *IEEE Trans Autom Control* 13(5): 558–563
- Jazwinski AH (1970) Stochastic processes and filtering theory. Academic, New York
- Joseph DP, Tou TJ (1961) On linear control theory. *Trans Am Inst Electr Eng* 80(18):193–196
- Julier SJ, Uhlmann JK (1997) A non-divergent estimation algorithm in the presence of unknown correlations. In: Proceedings of the 1997 American Control Conference, Albuquerque, vol 4, pp 2369–2373
- Julier SJ, Uhlmann JK (2004) Unscented Filtering and Nonlinear Estimation. *Proc IEEE* 92(3): 401–422
- Julier SJ, Uhlmann JK, Durrant-Whyte H (1995) A new approach for filtering nonlinear systems. In: Proceedings of the 1995 American control conference, Seattle, vol 3, pp 1628–1632
- Kailath T (1968) An innovations approach to least-squares estimation, Part I: linear filtering in additive white noise. *IEEE Trans Autom Control* 13(6): 646–655
- Kailath T (1970) The innovations approach to detection and estimation theory. *Proc IEEE* 58(5): 680–695
- Kailath T (1973) Some new algorithms for recursive estimation in constant linear systems. *IEEE Trans Inf Theory* 19(6):750–760
- Kailath T, Frost PA (1968) An innovations approach to least-squares estimation, Part II: linear smoothing in additive white noise. *IEEE Trans Autom Control* 13(6):655–660
- Kailath T, Sayed AH, Hassibi B (2000) Linear estimation. Prentice Hall, Upper Saddle River
- Kalman RE (1960a) On the general theory of control systems. In: Proceedings of the 1st International Congress of IFAC, Moscow, USSR, vol 1, pp 481–492
- Kalman RE (1960b) A new approach to linear filtering and prediction problems. *Trans ASME J Basic Eng Ser D* 82(1):35–45
- Kalman RE, Bucy RS (1961) New results in linear filtering and prediction theory. *Trans ASME J Basic Eng Ser D* 83(3):95–108
- Kamgarpour M, Tomlin C (2007) Convergence properties of a decentralized Kalman filter. In: Proceedings of the 47th IEEE Conference on Decision and Control, New Orleans, pp 5492–5498
- Kaminski PG, Bryson AE (1972) Discrete square-root smoothing. In: Proceedings of the AIAA Guidance and Control Conference, paper no. 72–877, Stanford
- Kay SM (1993) Fundamentals of statistical signal processing, volume 1: estimation theory. Prentice Hall, Englewood Cliffs
- Kim K, Shevlyakov G (2008) Why Gaussianity? *IEEE Signal Process Mag* 25(2):102–113
- Kim J, Woods JW (1998) 3-D Kalman filter for image motion estimation. *IEEE Trans Image Process* 7(1):42–52
- Koch KR (1999) Parameter estimation and hypothesis testing in linear models. Springer, Berlin
- Kolmogorov AN (1933) Grundbegriffe der Wahrscheinlichkeitsrechnung (in German) (English translation edited by Nathan Morrison: Foundations of the theory of probability, Chelsea, New York, 1956). Springer, Berlin
- Kolmogorov AN (1941) Interpolation and extrapolation of stationary random sequences. *Izv Akad Nauk SSSR Ser Mat* 5:3–14 (in Russian) (English translation by G. Lindqvist, published in Selected works of A.N. Kolmogorov - Volume II: Probability theory and mathematical statistics, A.N. Shyryaev (Ed.), pp. 272–280, Kluwer, Dordrecht, Netherlands, 1992)
- Kushner HJ (1962) On the differential equations satisfied by conditional probability densities of Markov processes with applications. *SIAM J Control Ser A* 2(1):106–199

- Kushner HJ (1967) Dynamical equations for optimal non-linear filtering. *J Differ Equ* 3(2):179–190
- Kushner HJ (1970) Filtering for linear distributed parameter systems. *SIAM J Control* 8(3): 346–359
- Kwakernaak H (1967) Optimal filtering in linear systems with time delays. *IEEE Trans Autom Control* 12(2):169–173
- Lainiotis DG (1972) Adaptive pattern recognition: a state-variable approach. In: Watanabe S (ed) *Frontiers of pattern recognition*. Academic, New York
- Lainiotis DG (1974) Estimation: a brief survey. *Inf Sci* 7:191–202
- Laplace PS (1814) *Essai philosophique sur le probabilités* (Translated in English by F.W. Truscott, F.L. Emory, Wiley, New York, 1902)
- Legendre AM (1810) Méthode des moindres carrés, pour trouver le milieu le plus probable entre les résultats de différentes observations. In: *Mémoires de la Classe des Sciences Mathématiques et Physiques de l'Institut Impérial de France*, pp 149–154. Originally appeared as appendix to the book *Nouvelle méthodes pour la détermination des orbites des comètes*, 1805
- Lehmann EL, Casella G (1998) *Theory of point estimation*. Springer, New York
- Leibungudt BG, Rault A, Gendreau F (1983) Application of Kalman filtering to demographic models. *IEEE Trans Autom Control* 28(3):427–434
- Levy BC (2008) *Principles of signal detection and parameter estimation*. Springer, New York
- Ljung L (1999) *System identification: theory for the user*. Prentice Hall, Upper Saddle River
- Los Alamos Scientific Laboratory (1966) Fermi invention rediscovered at LASL. *The Atom*, pp 7–11
- Luenberger DG (1964) Observing the state of a linear system. *IEEE Trans Mil Electron* 8(2):74–80
- Mahler RPS (1996) A unified foundation for data fusion. In: Sadjadi FA (ed) *Selected papers on sensor and data fusion*. SPIE MS-124. SPIE Optical Engineering Press, Bellingham, pp 325–345. Reprinted from 7th Joint Service Data Fusion Symposium, pp 154–174, Laurel (1994)
- Mahler RPS (2003) Multitarget Bayes filtering via first-order multitarget moments. *IEEE Trans Aerosp Electron Syst* 39(4):1152–1178
- Mahler RPS (2007a) *Statistical multisource multitarget information fusion*. Artech House, Boston
- Mahler RPS (2007b) PHD filters of higher order in target number. *IEEE Trans Aerosp Electron Syst* 43(4):1523–1543
- Mangoubi ES (1998) *Robust estimation and failure detection: a concise treatment*. Springer, New York
- Maybeck PS (1979 and 1982) *Stochastic models, estimation and control*, vols I, II and III. Academic, New York. 1979 (volume I) and 1982 (volumes II and III)
- Mayer-Wolf E, Zakai M (1983) On a formula relating the Shannon information to the Fisher information for the filtering problem. In: Korezlioglu H, Mazziotto G and Szpirglas J (eds) *Lecture notes in control and information sciences*, vol 61. Springer, New York, pp 164–171
- Mayne DQ (1966) A solution of the smoothing problem for linear dynamic systems. *Automatica* 4(2):73–92
- McGarty TP (1971) The estimation of the constituent densities of the upper atmosphere. *IEEE Trans Autom Control* 16(6):817–823
- McGee LA, Schmidt SF (1985) Discovery of the Kalman filter as a practical tool for aerospace and industry, NASA-TM-86847
- McGeoch CC, Wang C (2013) Experimental evaluation of an adiabatic quantum system for combinatorial optimization. In: *Computing Frontiers 2013 Conference*, Ischia, article no. 23. doi:10.1145/2482767.2482797
- McGrayne SB (2011) The theory that would not die. How Bayes' rule cracked the enigma code, hunted down Russian submarines, and emerged triumphant from two centuries of controversies. Yale University Press, New Haven/London
- Meditch JS (1967) Orthogonal projection and discrete optimal linear smoothing. *SIAM J Control* 5(1):74–89
- Meditch JS (1971) Least-squares filtering and smoothing for linear distributed parameter systems. *Automatica* 7(3):315–322
- Mendel JM (1977) White noise estimators for seismic data processing in oil exploration. *IEEE Trans Autom Control* 22(5):694–706
- Mendel JM (1983) *Optimal seismic deconvolution: an estimation-based approach*. Academic, New York
- Mendel JM (1990) *Maximum likelihood deconvolution – a journey into model-based signal processing*. Springer, New York
- Metropolis N, Ulam S (1949) The Monte Carlo method. *J Am Stat Assoc* 44(247):335–341
- Milanese M, Belforte G (1982) Estimation theory and uncertainty intervals evaluation in presence of unknown but bounded errors: linear families of models and estimators. *IEEE Trans Autom Control* 27(2):408–414
- Milanese M, Vicino A (1993) Optimal estimation theory for dynamic systems with set-membership uncertainty. *Automatica* 27(6):427–446
- Miller WL, Lewis JB (1971) Dynamic state estimation in power systems. *IEEE Trans Autom Control* 16(6):841–846
- Mlodinow L (2009) *The drunkard's walk: how randomness rules our lives*. Pantheon Books, New York. Copyright (C) Leonard Mlodinow
- Monticelli A (1999) *State estimation in electric power systems: a generalized approach*. Kluwer, Dordrecht
- Morf M, Kailath T (1975) Square-root algorithms for least-squares estimation. *IEEE Trans Autom Control* 20(4):487–497
- Morf M, Sidhu GS, Kailath T (1974) Some new algorithms for recursive estimation in constant, linear, discrete-time systems. *IEEE Trans Autom Control* 19(4):315–323
- Mullane J, Vo B-N, Adams M, Vo B-T (2011) *Random finite sets for robotic mapping and SLAM*. Springer, Berlin
- Nachazel K (1993) *Estimation theory in hydrology and water systems*. Elsevier, Amsterdam

- Najim M (2008) Modeling, estimation and optimal filtering in signal processing. Wiley, Hoboken. First published in France by Hermes Science/Lavoisier as *Modélisation, estimation et filtrage optimal en traitement du signal* (2006)
- Neal SR (1967) Discussion of parametric relations for the filter predictor. *IEEE Trans Autom Control* 12(3): 315–317
- Netz R, Noel W (2011) *The Archimedes codex*. Phoenix, Philadelphia, PA
- Nikoukhah R, Willsky AS, Levy BC (1992) Kalman filtering and Riccati equations for descriptor systems. *IEEE Trans Autom Control* 37(9):1325–1342
- Olfati-Saber R (2007) Distributed Kalman filtering for sensor networks. In: *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, pp 5492–5498
- Olfati-Saber R, Fax JA, Murray R (2007) Consensus and cooperation in networked multi-agent systems. *Proc IEEE* 95(1):215–233
- Painter JH, Kerstetter D, Jowers S (1990) Reconciling steady-state Kalman and α – β filter design. *IEEE Trans Aerosp Electron Syst* 26(6):986–991
- Park S, Serpedin E, Qarage K (2013) Gaussian assumption: the least favorable but the most useful. *IEEE Signal Process Mag* 30(3):183–186
- Personick SD (1971) Application of quantum estimation theory to analog communication over quantum channels. *IEEE Trans Inf Theory* 17(3):240–246
- Pindyck RS, Roberts SM (1974) Optimal policies for monetary control. *Ann Econ Soc Meas* 3(1): 207–238
- Poor HV (1994) *An introduction to signal detection and estimation*, 2nd edn. Springer, New York. (first edition in 1988)
- Poor HV, Looze D (1981) Minimax state estimation for linear stochastic systems with noise uncertainty. *IEEE Trans Autom Control* 26(4):902–906
- Potter JE, Stern RG (1963) Statistical filtering of space navigation measurements. In: *Proceedings of the 1963 AIAA Guidance and Control Conference*, paper no. 63–333, Cambridge
- Ramm AG (2005) *Random fields estimation*. World Scientific, Hackensack
- Rao C (1945) Information and the accuracy attainable in the estimation of statistical parameters. *Bull Calcutta Math Soc* 37:81–89
- Rao CV, Rawlings JB, Lee JH (2001) Constrained linear state estimation – a moving horizon approach. *Automatica* 37(10):1619–1628
- Rao CV, Rawlings JB, Mayne DQ (2003) Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations. *IEEE Trans Autom Control* 48(2): 246–257
- Rauch HE (1963) Solutions to the linear smoothing problem. *IEEE Trans Autom Control* 8(4): 371–372
- Rauch HE, Tung F, Striebel CT (1965) Maximum likelihood estimates of linear dynamic systems. *AIAA J* 3(8):1445–1450
- Reid D (1979) An algorithm for tracking multiple targets. *IEEE Trans Autom Control* 24(6):843–854
- Riccati JF (1722) *Animadversiones in aequationes differentiales secundi gradii*. *Acta Eruditorum Lipsiae*, Supplementa 8:66–73. Observations regarding differential equations of second order, translation of the original Latin into English, by I. Bruce, 2007
- Riccati JF (1723) Appendix in *animadversiones in aequationes differentiales secundi gradii*. *Acta Eruditorum Lipsiae*
- Ristic B (2013) *Particle filters for random set models*. Springer, New York
- Ristic B, Arulampalam S, Gordon N (2004) *Beyond the Kalman filter: particle filters for tracking applications*. Artech House, Boston
- Ristic B, Vo B-T, Vo B-N, Farina A (2013) A tutorial on Bernoulli filters: theory, implementation and applications. *IEEE Trans Signal Process* 61(13):3406–3430
- Robinson EA (1963) Mathematical development of discrete filters for detection of nuclear explosions. *J Geophys Res* 68(19):5559–5567
- Roman WS, Hsu C, Habegger LT (1971) Parameter identification in a nonlinear reactor system. *IEEE Trans Nucl Sci* 18(1):426–429
- Sage AP, Masters GW (1967) Identification and modeling of states and parameters of nuclear reactor systems. *IEEE Trans Nucl Sci* 14(1):279–285
- Sage AP, Melsa JL (1971) *Estimation theory with applications to communications and control*. McGraw-Hill, New York
- Schmidt SF (1966) Application of state-space methods to navigation problems. *Adv Control Syst* 3:293–340
- Schmidt SF (1970) *Computational techniques in Kalman filtering*, NATO-AGARD-139
- Schönhoff TA, Giordano AA (2006) *Detection and estimation theory and its applications*. Pearson Prentice Hall, Upper Saddle River
- Schweppe FC (1968) Recursive state estimation with unknown but bounded errors and system inputs. *IEEE Trans Autom Control* 13(1):22–28
- Segall A (1976) Recursive estimation from discrete-time point processes. *IEEE Trans Inf Theory* 22(4): 422–431
- Servi L, Ho Y (1981) Recursive estimation in the presence of uniformly distributed measurement noise. *IEEE Trans Autom Control* 26(2):563–565
- Simon D (2006) *Optimal state estimation – Kalman, H_∞ and nonlinear approaches*. Wiley, New York
- Simpson HR (1963) Performance measures and optimization condition for a third-order tracker. *IRE Trans Autom Control* 8(2):182–183
- Sklansky J (1957) Optimizing the dynamic parameters of a track-while-scan system. *RCA Rev* 18:163–185
- Smeth P, Ristic B (2004) Kalman filter and joint tracking and classification in TBM framework. In: *Proceedings of the 7th international conference on information fusion*, Stockholm, pp 46–53
- Smith R, Self M, Cheeseman P (1986) Estimating uncertain spatial relationships in robotics. In: *Proceedings of*

- the 2nd conference on uncertainty in artificial intelligence, Philadelphia, pp 435–461
- Snijders TAB, Koskinen J, Schweinberger M (2012) Maximum likelihood estimation for social network dynamics. *Ann Appl Stat* 4(2):567–588
- Snyder DL (1968) The state-variable approach to continuous estimation with applications to analog communications. MIT, Cambridge
- Snyder DL (1970) Estimation of stochastic intensity functions of conditional Poisson processes. Monograph no. 128, Biomedical Computer Laboratory, Washington University, St. Louis
- Söderström T (1994) Discrete-time stochastic system – estimation & control. Prentice Hall, New York
- Söderström T, Stoica P (1989) System identification. Prentice Hall, New York
- Sorenson HW, Alspach DL (1970) Gaussian sum approximations for nonlinear filtering. In: Proceedings of the 1970 IEEE Symposium on Adaptive Processes, Austin, TX pp 19.3.1–19.3.9
- Sorenson HW, Alspach DL (1971) Recursive Bayesian estimation using Gaussian sums. *Automatica* 7(4): 465–479
- Stankovic SS, Stankovic MS, Stipanovic DM (2009) Consensus based overlapping decentralized estimation with missing observations and communication faults. *Automatica* 45(6):1397–1406
- Stark L (1968) Neurological control systems studies in bioengineering. Plenum Press, New York
- Stengel PF (1994) Optimal control and estimation. Dover, New York. Originally published as Stochastic optimal control. Wiley, New York, 1986
- Stephant J, Charara A, Meizel D (2004) Virtual sensor: application to vehicle sideslip angle and transversal forces. *IEEE Trans Ind Electron* 51(2): 278–289
- Stratonovich RL (1959) On the theory of optimal nonlinear filtering of random functions. *Theory Probab Appl* 4:223–225
- Stratonovich RL (1960) Conditional Markov processes. *Theory Probab Appl* 5(2):156–178
- Taylor JH (1979) The Cramér-Rao estimation error lower bound computation for deterministic nonlinear systems. *IEEE Trans Autom Control* 24(2): 343–344
- Thrun S, Burgard W, Fox D (2006) Probabilistic robotics. MIT, Cambridge
- Tichavsky P, Muravchik CH, Nehorai A (1998) Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Trans Signal Process* 6(5):1386–1396
- Toyoda J, Chen MS, Inoue Y (1970) An application of state estimation to short-term load forecasting, Part I: forecasting model and Part II: implementation. *IEEE Trans Power Appar Syst* 89(7):1678–1682 (Part I) and 1683–1688 (Part II)
- Tsybakov AB (2009) Introduction to nonparametric estimation. Springer, New York
- Tuncer TE, Friedlander B (2009) Classical and modern direction-of-arrival estimation. Academic, Burlington
- Tzafestas SG, Nightingale JM (1968) Optimal filtering, smoothing and prediction in linear distributed-parameter systems. *Proc Inst Electr Eng* 115(8): 1207–1212
- Ulam S (1952) Random processes and transformations. In: Proceedings of the International Congress of Mathematicians, Cambridge, vol 2, pp 264–275
- Ulam S, Richtmeyer RD, von Neumann J (1947) Statistical methods in neutron diffusion, Los Alamos Scientific Laboratory report LAMS-551
- Van Trees H (1971) Detection, estimation and modulation theory – Parts I, II and III. Wiley, New York. Republished in 2013
- van Trees HL, Bell KL (2007) Bayesian bounds for parameter estimation and nonlinear filtering/tracking. Wiley, Hoboken/IEEE, Piscataway
- Venerus JC, Bullock TE (1970) Estimation of the dynamic reactivity using digital Kalman filtering. *Nucl Sci Eng* 40:199–205
- Verdú S, Poor HV (1984) Minimax linear observers and regulators for stochastic systems with uncertain second-order statistics. *IEEE Trans Autom Control* 29(6):499–511
- Vicino A, Zappa G (1996) Sequential approximation of feasible parameter sets for identification with set membership uncertainty. *IEEE Trans Autom Control* 41(6):774–785
- Vo B-N, Ma WK (1996) The Gaussian mixture probability hypothesis density filter. *IEEE Trans Signal Process* 54(11):4091–4101
- Vo B-T, Vo B-N, Cantoni A (2007) Analytic implementations of the cardinalized probability hypothesis density filter. *IEEE Trans Signal Process* 55(7):3553–3567
- Wakita H (1973) Estimation of the vocal tract shape by inverse optimal filtering. *IEEE Trans Audio Electroacoust* 21(5):417–427
- Wertz W (1978) Statistical density estimation – a survey. Vandenhoeck and Ruprecht, Göttingen
- Wiener N (1949) Extrapolation, interpolation and smoothing of stationary time-series. MIT, Cambridge
- Willsky AS (1976) A survey of design methods for failure detection in dynamic systems. *Automatica* 12(6): 601–611
- Wonham WM (1965) Some applications of stochastic differential equations to optimal nonlinear filtering. *SIAM J Control* 2(3):347–369
- Woods JW, Radewan C (1977) Kalman filtering in two dimensions. *IEEE Trans Inf Theory* 23(4): 473–482
- Xiao L, Boyd S, Lall S (2005) A scheme for robust distributed sensor fusion based on average consensus. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, Los Angeles, pp 63–70
- Zachrisson LE (1969) On optimal smoothing of continuous-time Kalman processes. *Inf Sci* 1(2): 143–172
- Zellner A (1971) An introduction to Bayesian inference in econometrics. Wiley, New York

Event-Triggered and Self-Triggered Control

W.P.M.H. Heemels¹, Karl H. Johansson^{2,3}, and Paulo Tabuada^{4,5}

¹Department of Mechanical Engineering,
Eindhoven University of Technology,
Eindhoven, The Netherlands

²Electrical Engineering and Computer Science,
KTH – Royal Institute of Technology,
Stockholm, Sweden

³ACCESS Linnaeus Center, Royal Institute of
Technology, Stockholm, Sweden

⁴Electrical and Computer Engineering
Department, UCLA, Los Angeles, CA, USA

⁵Department of Electrical Engineering,
University of California, Los Angeles, CA, USA

Abstract

Recent developments in computer and communication technologies have led to a new type of large-scale resource-constrained wireless embedded control systems. It is desirable in these systems to limit the sensor and control computation and/or communication to instances when the system needs attention. However, classical sampled-data control is based on performing sensing and actuation periodically rather than when the system needs attention. This article discusses event- and self-triggered control systems where sensing and actuation is performed when needed. Event-triggered control is reactive and generates sensor sampling and control actuation when, for instance, the plant state deviates more than a certain threshold from a desired value. Self-triggered control, on the other hand, is proactive and computes the next sampling or actuation instance ahead of time. The basics of these control strategies are introduced together with references for further reading.

Keywords

Event-triggered control · Hybrid systems · Real-time control · Resource-constrained

embedded control · Sampled-data systems · Self-triggered control

Introduction

In standard control textbooks, e.g., Åström and Wittenmark (1997) and Franklin et al. (2010), periodic control is presented as the only choice for implementing feedback control laws on digital platforms. Although this time-triggered control paradigm has proven to be extremely successful in many digital control applications, recent developments in computer and communication technologies have led to a new type of large-scale resource-constrained (wireless) control systems that call for a reconsideration of this traditional paradigm. In particular, the increasing popularity of (shared) wired and wireless networked control systems raises the importance of explicitly addressing energy, computation, and communication constraints when designing feedback control loops. Aperiodic control strategies that allow the inter-execution times of control tasks to be varying in time offer potential advantages with respect to periodic control when handling these constraints, but they also introduce many new interesting theoretical and practical challenges.

Although the discussions regarding periodic vs. aperiodic implementation of feedback control loops date back to the beginning of computer-controlled systems, e.g., Gupta (1963), in the late 1990s two influential papers (Åström and Bernhardsson 1999; Årzén 1999) highlighted the advantages of *event-based* feedback control. These two papers spurred the development of the first *systematic designs* of event-based implementations of stabilizing feedback control laws, e.g., Yook et al. (2002), Tabuada (2007), Heemels et al. (2008), and Henningsson et al. (2008). Since then, several researchers have improved and generalized these results and alternative approaches have appeared. In the meantime, also so-called *self-triggered* control (Velasco et al. 2003) emerged. Event-triggered and self-triggered control systems consist of two elements, namely, a feedback controller that computes the control input and a triggering mechanism that determines when the control

input has to be updated again. The difference between event-triggered control and self-triggered control is that the former is reactive, while the latter is proactive. Indeed, in event-triggered control, a triggering condition based on current measurements is continuously monitored and when the condition holds, an event is triggered. In self-triggered control the next update time is precomputed at a control update time based on predictions using previously received data and knowledge of the plant dynamics. In some cases, it is advantageous to combine event-triggered and self-triggered control resulting in a control system reactive to unpredictable disturbances and proactive by predicting future use of resources.

Time-Triggered, Event-Triggered and Self-Triggered Control

To indicate the differences between various digital implementations of feedback control laws, consider the control of the nonlinear plant

$$\dot{x} = f(x, u) \quad (1)$$

with $x \in \mathbb{R}^{n_x}$ the state variable and $u \in \mathbb{R}^{n_u}$ the input variable. The system is controlled by a nonlinear state feedback law

$$u = h(x) \quad (2)$$

where $h : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$ is an appropriate mapping that has to be implemented on a digital platform. Recomputing the control value and updating the actuator signals will occur at times denoted by $t_0, t_1, t_2, \dots$ with $t_0 = 0$. If we assume the inputs to be held constant in between the successive recomputations of the control law (referred to as sample-and-hold or zero-order-hold), we have

$$u(t) = u(t_k) = h(x(t_k)) \quad \forall t \in [t_k, t_{k+1}), \quad k \in \mathbb{N}. \quad (3)$$

We refer to the instants $\{t_k\}_{k \in \mathbb{N}}$ as the *triggering* times or *execution* times. Based on these times we can easily explain the difference between

time-triggered control, event-triggered control, and self-triggered control.

In *time-triggered* control we have the equality $t_k = kT_s$ with $T_s > 0$ being the sampling period. Hence, the updates take place equidistantly in time irrespective of how the system behaves. There is no “feedback mechanism” in determining the execution times; they are determined a priori and in “open loop.” Another way of writing the triggering mechanism in time-triggered control is

$$t_{k+1} = t_k + T_s, \quad k \in \mathbb{N} \quad (4)$$

with $t_0 = 0$.

In *event-triggered* control the next execution time of the controller is determined by an event-triggering mechanism that continuously verifies if a certain condition based on the actual state variable becomes true. This condition includes often also information on the state variable $x(t_k)$ at the previous execution time t_k and can be written, for instance, as $C(x(t), x(t_k)) > 0$. Formally, the execution times are then determined by

$$t_{k+1} = \inf\{t > t_k \mid C(x(t), x(t_k)) > 0\} \quad (5)$$

with $t_0 = 0$. Hence, it is clear from (5) that there is a *feedback* mechanism present in the determination of the next execution time as it is based on the measured state variable. In this sense event-triggered control is *reactive*.

Finally, in *self-triggered* control the next execution time is determined *proactively* based on the measured state $x(t_k)$ at the previous execution time. In particular, there is a function $M : \mathbb{R}^{n_x} \rightarrow \mathbb{R}_{\geq 0}$ that specifies the next execution time as

$$t_{k+1} = t_k + M(x(t_k)) \quad (6)$$

with $t_0 = 0$. As a consequence, in self-triggered control both the control value $u(t_k)$ and the next execution time t_{k+1} are computed at execution time t_k . In between t_k and t_{k+1} , no further actions are required from the controller. Note that the time-triggered implementation can be seen as a special case of the self-triggered implementation by taking $M(x) = T_s$ for all $x \in \mathbb{R}^{n_x}$.

Clearly, in all the three implementation schemes T_s , C and M are chosen together with the feedback law given through h to provide stability and performance guarantees and to realize a certain utilization of computer and communication resources.

Lyapunov-Based Analysis

Much work on event-triggered control used one of the following two modeling and analysis frameworks: The perturbation approach and the hybrid system approach.

Perturbation Approach

In the perturbation approach one adopts perturbed models that describe how the event-triggered implementation of the control law perturbs the ideal continuous-time implementation $u(t) = h(x(t))$, $t \in \mathbb{R}_{\geq 0}$. In order to do so, consider the error e given by

$$e(t) = x(t_k) - x(t) \text{ for } t \in [t_k, t_{k+1}), k \in \mathbb{N}. \quad (7)$$

Using this error variable we can write the closed-loop system based on (1) and (3) as

$$\dot{x} = f(x, h(x + e)). \quad (8)$$

Essentially, the three implementations discussed above have their own way of indicating when an execution takes place and the error e is reset to zero. The equation (8) clearly shows how the ideal closed-loop system is *perturbed* by using a time-triggered, event-triggered, or self-triggered implementation of the feedback law in (2). Indeed, when $e = 0$ we obtain the ideal closed loop

$$\dot{x} = f(x, h(x)). \quad (9)$$

The control law in (2) is typically chosen so as to guarantee that the system in (9) has certain global asymptotic stability (GAS) properties. In particular, it is often assumed that there exists a Lyapunov function $V : \mathbb{R}_{n_x} \rightarrow \mathbb{R}_{\geq 0}$ in the sense

that V is positive definite and for all $x \in \mathbb{R}_{n_x}$ we have

$$\frac{\partial V}{\partial x} f(x, h(x)) \leq -\|x\|^2. \quad (10)$$

Note that this inequality is stronger than strictly needed (at least for nonlinear systems), but for pedagogical reasons we choose this simpler formulation. For the perturbed model, the inequality in (10) can in certain cases (including linear systems) be modified to

$$\frac{\partial V}{\partial x} f(x, h(x)) \leq -\|x\|^2 + \beta \|e\|^2 \quad (11)$$

in which $\beta > 0$ is a constant used to indicate how the presence of the implementation error e affects the decrease of the Lyapunov function. Based on (10) one can now choose the function C in (5) to preserve GAS of the event-triggered implementation. For instance, $C(x(t), x(t_k)) = \|x(t_k) - x(t)\| - \sigma \|x(t)\|$, i.e.,

$$t_{k+1} = \inf\{t > t_k \mid \|e(t)\| > \sigma \|x(t)\|\}, \quad (12)$$

assures that

$$\|e\| \leq \sigma \|x\| \quad (13)$$

holds. When $\sigma < 1/\beta$, we obtain from (11) and (13) that GAS properties are preserved for the event-triggered implementation. Besides, under certain conditions provided in Tabuada (2007), a global positive lower bound exists on the inter-execution times, i.e., there exists a $\tau_{\min} > 0$ such that $t_{k+1} - t_k > \tau_{\min}$ for all $k \in \mathbb{N}$ and all initial states x_0 .

Also self-triggered controllers can be derived using the perturbation approach. In this case, stability properties can be guaranteed by choosing M in (6) ensuring that $C(x(t), x(t_k)) \leq 0$ holds for all times $t \in [t_k, t_{k+1})$ and all $k \in \mathbb{N}$.

Hybrid System Approach

By taking as a state variable $\xi = (x, e)$, one can write the closed-loop event-triggered control system given by (1), (3), and (5) as the hybrid impulsive system (Goebel et al. 2009)

$$\dot{\xi} = \begin{pmatrix} f(x, h(x+e)) \\ -f(x, h(x+e)) \end{pmatrix} \text{ when } C(x, x+e) \geq 0 \quad (14a)$$

$$\xi^+ = \begin{pmatrix} x \\ 0 \end{pmatrix} \text{ when } C(x, x+e) \leq 0. \quad (14b)$$

This observation was made in Donkers and Heemels (2010, 2012) and Postoyan et al. (2011). Tools from hybrid system theory can be used to analyze this model, which is more accurate as it includes the error dynamics of the event-triggered closed-loop system. In fact, the stability bounds obtained via the hybrid system approach can be proven to be never worse than ones obtained using the perturbation approach in many cases, see, e.g., Donkers and Heemels (2012), and typically the hybrid system approach provides (strictly) better results in practice. However, in general an analysis via the hybrid system approach is more complicated than using a perturbation approach.

Note that by including a time variable τ , one can also write the closed-loop system corresponding to self-triggered control (1), (3), and (6) as a hybrid system using the state variable $\chi = (x, e, \tau)$. This leads to the model

$$\dot{\chi} = \begin{pmatrix} f(x, h(x+e)) \\ -f(x, h(x+e)) \\ 1 \end{pmatrix} \text{ when } 0 \leq \tau \leq M(x+e) \quad (15a)$$

$$\chi^+ = \begin{pmatrix} x \\ 0 \\ 0 \end{pmatrix} \text{ when } \tau = M(x+e), \quad (15b)$$

which can be used for analysis based on hybrid tools as well.

Alternative Event-Triggering Mechanisms

There are various alternative event-triggering mechanisms. A few of them are described in this section.

Relative, Absolute, and Mixed Triggering Conditions

Above we discussed a very basic event-triggering condition in the form given in (12), which is sometimes called *relative* triggering as the next control task is executed at the instant when the ratio of the norms of the error $\|e\|$ and the measured state $\|x\|$ is larger than or equal to σ . Also *absolute* triggering of the form

$$t_{k+1} = \inf\{t > t_k \mid \|e(t)\| \geq \delta\} \quad (16)$$

can be considered. Here $\delta > 0$ is an absolute threshold, which has given this scheme the name send-on-delta (Miskowicz 2006). Recently, a *mixed* triggering mechanism of the form

$$t_{k+1} = \inf\{t > t_k \mid \|e(t)\| \geq \sigma \|x(t)\| + \delta\}, \quad (17)$$

combining an absolute and a relative threshold, was proposed (Donkers and Heemels 2012). It is particularly effective in the context of output-based control.

Model-Based Triggering

In the triggering conditions discussed so far, essentially the current control value $u(t)$ is based on a *held* value $x(t_k)$ of the state variable, as specified in (3). However, if good model-based information regarding the plant is available, one can use better model-based predictions of the actuator signal. For instance, in the linear context, (Lunze and Lehmann 2010) proposed to use a *control input generator* instead of a plain zero-order hold function. In fact, the plant model was described by

$$\dot{x} = Ax + Bu + Ew \quad (18)$$

with $x \in \mathbb{R}^{n_x}$ the state variable, $u \in \mathbb{R}^{n_u}$ the input variable, and $w \in \mathbb{R}^{n_w}$ a bounded disturbance input. It was assumed that a well functioning state feedback controller $u = Kx$ was available. The control input generator was then based on the model-based predictions given for $[t_k, t_{k+1})$ by

$$\begin{aligned}\dot{x}_s(t) &= (A + BK)x_s(t) + E\hat{w}(t_k) \\ \text{with } x_s(t_k) &= x(t_k)\end{aligned}\quad (19)$$

and $\hat{w}(t_k)$ is an estimate for the (average) disturbance value, which is determined at execution time t_k , $k \in \mathbb{N}$. The applied input to the actuator is then given by $u(t) = Kx_s(t)$ for $t \in [t_k, t_{k+1})$, $k \in \mathbb{N}$. Note that (19) provides a prediction of the closed-loop state evolution using the latest received value of the state $x(t_k)$ and the estimate $\hat{w}(t_k)$ of the disturbances. Also the event-triggering condition is based on this model-based prediction of the state as it is given by

$$t_{k+1} = \inf\{t > t_k \mid \|x_s(t) - x(t)\| \geq \delta\}. \quad (20)$$

Hence, when the prediction $x_s(t)$ diverts to far from the measured state $x(t)$, the next event is triggered so that updates of the state are sent to the actuator. These model-based triggering schemes can enhance the communication savings as they reduce the number of events by using model-based knowledge.

Other model-based event-triggered control schemes are proposed, for instance, in Yook et al. (2002), Garcia and Antsaklis (2013), and Heemels and Donkers (2013).

Triggering with Time-Regularization

Time-regularization was proposed for *output-based* triggering to avoid the occurrence of accumulations in the execution times (Zeno behavior) that would obstruct the existence of a positive lower bound on the inter-execution times $t_{k+1} - t_k$, $k \in \mathbb{N}$. In Tallapragada and Chopra (2012a, b), the triggering update

$$t_{k+1} = \inf\{t > t_k + T \mid \|e(t)\| \geq \sigma \|x(t)\|\} \quad (21)$$

was proposed, where $T > 0$ is a built-in lower bound on the minimal inter-execution times. The authors discussed how T and σ can be designed to guarantee closed-loop stability. In Heemels et al. (2008) a similar triggering was proposed using an absolute-type of triggering.

An alternative to exploiting a built-in lower bound T is combining ideas from time-triggered

control and event-triggering control. Essentially, the idea is to only verify a specific event-triggering condition at certain equidistant time instants kT_s , $k \in \mathbb{N}$, where $T_s > 0$ is the sampling period. Such proposals were mentioned in, for instance, Årzén (1999), Yook et al. (2002), Henningsson et al. (2008), and Heemels et al. (2008, 2013). In this case the execution times are given by

$$\begin{aligned}t_{k+1} &= \inf\{t > t_k \mid t = kT_s, k \in \mathbb{N}, \\ &\text{and } \|e(t)\| \geq \sigma \|x(t)\|\}\end{aligned}\quad (22)$$

in case a relative triggering is used. In Heemels et al. (2013) the term *periodic event-triggered control* was coined for this type of control.

Decentralized Triggering Conditions

Another important extension of the mentioned event-triggered controllers, especially in large-scale networked systems, is the decentralization of the event-triggered control. Indeed, if one focuses on any of the abovementioned event-triggering conditions (take, e.g., (5)), it is obvious that the full state variable $x(t)$ has to be continuously available in a central coordinator to determine if an event is triggered or not. If the sensors that measure the state are physically distributed over a wide area, this assumption is prohibitive for its implementation. In such cases, it is of high practical importance that the event-triggering mechanism can be decentralized and the execution of control tasks can be executed based on local information. One first idea could be to use *local* event-triggering mechanisms for the i -th sensor that measures x_i . One could “decentralize” the condition (5), into

$$t_{k^i+1}^i = \inf\{t > t_{k^i}^i \mid \|e_i(t)\| \geq \sigma \|x_i(t)\|\}, \quad (23)$$

in which $e_i(t) = x_i(t_{k^i}^i) - x_i(t)$ for $t \in [t_{k^i}^i, t_{k^i+1}^i)$, $k^i \in \mathbb{N}$. Note that each sensor now has its own execution times $t_{k^i}^i$, $k^i \in \mathbb{N}$ at which the information $x_i(t)$ is transmitted. More importantly, the triggering condition (23) is based on local data only and does not need a central coordinator having access to the complete state information.

Besides since (23) still guarantees that (13) holds, stability properties can still be guaranteed; see Mazo and Tabuada (2011).

Several other proposals for decentralized event-triggered control schemes were made, e.g., Persis et al. (2013), Wang and Lemmon (2011), Garcia and Antsaklis (2013), Yook et al. (2002), and Donkers and Heemels (2012).

Triggering for Multi-agent Systems

Event-triggered control strategies are suitable for cooperative control of multi-agent systems. In multi-agent systems, local control actions of individual agents should lead to a desirable global behavior of the overall system. A prototype problem for control of multi-agent systems is the agreement problem (also called the consensus or rendezvous problem), where the states of all agents should converge to a common value (sometimes the average of the agents' initial conditions). The agreement problem has been shown to be solvable for certain low-order dynamical agents in both continuous and discrete time, e.g., Olfati-Saber et al. (2007). It was recently shown in Dimarogonas et al. (2012), Shi and Johansson (2011), and Seyboth et al. (2013) that the agreement problem can be solved using event-triggered control. In Seyboth et al. (2013) the triggering times for agent i are determined by

$$t_{ki+1}^i = \inf\{t > t_{ki}^i \mid C_i(x_i(t), x_i(t_{ki}^i)) > 0\}, \quad (24)$$

which should be compared to the triggering times as specified through (5). The triggering condition compares the current state value with the one previously communicated, similarly to the previously discussed decentralized event-triggered control (see (23)), but now the communication is *only* to the agent's neighbors. Using such event-triggered communication, the convergence rate to agreement (i.e., $\|x_i(t) - x_j(t)\| \rightarrow 0$ as $t \rightarrow \infty$ for all i, j) can be maintained with a much lower communication rate than for time-triggered communication.

Outlook

Many simulation and experimental results show that event-triggered and self-triggered control strategies are capable of reducing the number of control task executions, while retaining a satisfactory closed-loop performance. In spite of these results, the actual deployment of these novel control paradigms in relevant applications is still rather marginal. Some exceptions include recent event-triggered control applications in underwater vehicles (Teixeira et al. 2010), process control (Lehmann et al. 2012), and control over wireless networks (Araujo et al. 2014). To foster the further development of event-triggered and self-triggered controllers in the future, it is therefore important to validate these strategies in practice, next to building up a complete system theory for them. Regarding the latter, it is fair to say that, even though many interesting results are currently available, the system theory for event-triggered and self-triggered control is far from being mature, certainly compared to the vast literature on time-triggered (periodic) sampled-data control. As such, many theoretical and practical challenges are ahead of us in this appealing research field.

Cross-References

- [Discrete Event Systems and Hybrid Systems, Connections Between](#)
- [Hybrid Dynamical Systems, Feedback Control of](#)
- [Models for Discrete Event Systems: An Overview](#)
- [Supervisory Control of Discrete-Event Systems](#)

Acknowledgments The work of Maurice Heemels was partially supported by the Dutch Technology Foundation (STW) and the Dutch Organization for Scientific Research (NWO) under the VICI grant “Wireless controls systems: A new frontier in automation”. The work of Karl Johansson was partially supported by the Knut and Alice Wallenberg Foundation and the Swedish Research Council. Maurice Heemels and Karl Johansson were also supported by the European 7th Framework Programme

Network of Excellence under grant HYCON2-257462. The work of Paulo Tabuada was partially supported by NSF awards 0834771 and 0953994.

Bibliography

- Araujo J, Mazo M Jr, Anta A, Tabuada P, Johansson KH (2014) System architectures, protocols, and algorithms for aperiodic wireless control systems. *Industrial Informatics, IEEE Trans Ind Inform.* 10(1): 175–184
- Årzén K-E (1999) A simple event-based PID controller. In: *Proceedings of the IFAC World Congress, Beijing, China*, vol 18, pp 423–428. Preprints
- Åström KJ, Bernhardsson BM (1999) Comparison of periodic and event based sampling for first order stochastic systems. In: *Proceedings of the IFAC World Congress, Beijing, China*, pp 301–306
- Aström, KJ, Wittenmark B (1997) *Computer controlled systems*. Prentice Hall, Upper Saddle River
- De Persis C, Sailer R, Wirth F (2013) Parsimonious event-triggered distributed control: a Zeno free approach. *Automatica* 49(7):2116–2124
- Dimarogonas DV, Frazzoli E, Johansson KH (2012) Distributed event-triggered control for multi-agent systems. *IEEE Trans Autom Control* 57(5):1291–1297
- Donkers MCF, Heemels WPMH (2010) Output-based event-triggered control with guaranteed $\mathcal{L}_\infty$ -gain and improved event-triggering. In: *Proceedings of the IEEE conference on decision and control, Atlanta, Georgia, USA*, pp 3246–3251
- Donkers MCF, Heemels WPMH (2012) Output-based event-triggered control with guaranteed $\mathcal{L}_\infty$ -gain and improved and decentralised event-triggering. *IEEE Trans Autom Control* 57(6): 1362–1376
- Franklin GF, Powell JD, Emami-Naeini A (2010) *Feedback control of dynamical systems*. Prentice Hall, Upper Saddle River
- Garcia E, Antsaklis PJ (2013) Model-based event-triggered control for systems with quantization and time-varying network delays. *IEEE Trans Autom Control* 58(2):422–434
- Goebel R, Sanfelice R, Teel AR (2009) Hybrid dynamical systems. *IEEE Control Syst Mag* 29: 28–93
- Gupta S (1963) Increasing the sampling efficiency for a control system. *IEEE Trans Autom Control* 8(3): 263–264
- Heemels WPMH, Donkers MCF (2013) Model-based periodic event-triggered control for linear systems. *Automatica* 49(3):698–711
- Heemels WPMH, Sandee JH, van den Bosch PPJ (2008) Analysis of event-driven controllers for linear systems. *Int J Control* 81:571–590
- Heemels WPMH, Donkers MCF, Teel AR (2013) Periodic event-triggered control for linear systems. *IEEE Trans Autom Control* 58(4):847–861
- Henningssson T, Johansson E, Cervin A (2008) Sporadic event-based control of first-order linear stochastic systems. *Automatica* 44:2890–2895
- Lehmann D, Kiener GA, Johansson KH (2012) Event-triggered PI control: saturating actuators and anti-windup compensation. In: *Proceedings of the IEEE conference on decision and control, Maui*
- Lunze J, Lehmann D (2010) A state-feedback approach to event-based control. *Automatica* 46:211–215
- Mazo M Jr, Tabuada P (2011) Decentralized event-triggered control over wireless sensor/actuator networks. *IEEE Trans Autom Control* 56(10):2456–2461. Special issue on Wireless Sensor and Actuator Networks
- Miskowicz M (2006) Send-on-delta concept: an event-based data-reporting strategy. *Sensors* 6: 49–63
- Olfati-Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. *Proc IEEE* 95(1):215–233
- Postoyan R, Anta A, Nešić D, Tabuada P (2011) A unifying Lyapunov-based framework for the event-triggered control of nonlinear systems. In: *Proceedings of the joint IEEE conference on decision and control and European control conference, Orlando*, pp 2559–2564
- Seyboth GS, Dimarogonas DV, Johansson KH (2013) Event-based broadcasting for multi-agent average consensus. *Automatica* 49(1):245–252
- Shi G, Johansson KH (2011) Multi-agent robust consensus—part II: application to event-triggered coordination. In: *Proceedings of the IEEE conference on decision and control, Orlando*
- Tabuada P (2007) Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Trans Autom Control* 52(9):1680–1685
- Tallapragada P, Chopra N (2012a) Event-triggered decentralized dynamic output feedback control for LTI systems. In: *IFAC workshop on distributed estimation and control in networked systems*, pp 31–36
- Tallapragada P, Chopra N (2012b) Event-triggered dynamic output feedback control for LTI systems. In: *IEEE 51st annual conference on decision and control (CDC), Maui*, pp 6597–6602
- Teixeira PV, Dimarogonas DV, Johansson KH, Borges de Sousa J (2010) Event-based motion coordination of multiple underwater vehicles under disturbances. In: *IEEE OCEANS, Sydney*
- Velasco M, Martí P, Fuertes JM (2003) The self triggered task model for real-time control systems. In: *Proceedings of 24th IEEE real-time systems symposium, work-in-progress session, Cancun, Mexico*
- Wang X, Lemmon MD (2011) Event-triggering in distributed networked systems with data dropouts and delays. *IEEE Trans Autom Control* 56:586–601
- Yook JK, Tilbury DM, Soparkar NR (2002) Trading computation for bandwidth: reducing communication in distributed control systems using state estimators. *IEEE Trans Control Syst Technol* 10(4): 503–518

Evolutionary Games

Eitan Altman

INRIA, Sophia-Antipolis, France

Abstract

Evolutionary games constitute the most recent major mathematical tool for understanding, modelling and predicting evolution in biology and other fields. They complement other well established tools such as branching processes and the Lotka-Volterra (1910) equations (e.g. for the predator - prey dynamics or for epidemics evolution). Evolutionary Games also brings novel features to game theory. First, it focuses on the dynamics of competition rather than restricting attention to the equilibrium. In particular, it tries to explain how an equilibrium emerges. Second, it brings new definitions of stability, that are more adapted to the context of large populations. Finally, in contrast to standard game theory, players are not assumed to be “rational” or “knowledgeable” as to anticipate the other players’ choices. The objective of this article, is to present foundations as well as recent advances in evolutionary games, highlight the novel concepts that they introduce with respect to game theory as formulated by John Nash, and describe through several examples their huge potential as tools for modeling interactions in complex systems.

Keywords

Evolutionary stable strategies · Fitness · Replicator dynamics

Introduction

Evolutionary game theory is the youngest of several mathematical tools used in describing and modeling evolution. It was preceded by the theory of branching processes (Watson and Francis Galton 1875) and its extensions (Altman 2008)

which have been introduced in order to explain the evolution of family names in the English population of the second half of the nineteenth century. This theory makes use of the probabilistic distribution of the number of offspring of an individual in order to predict the probability at which the whole population would become eventually extinct. It describes the evolution of the number of offsprings of a given individual. The Lotka-Volterra equations (Lotka-Volterra 1910) and their extensions are differential equations that describe the population size of each of several species that have a predator-prey type relation. One of the foundations in evolutionary games (and its extension to population games) which is often used as the starting point in their definition is the replicator dynamics which, similarly to the Lotka-Volterra equations, describe the evolution of the size of various species that interact with each other (or of various behaviors within a given population). In both the Lotka-Volterra equations and in replicator dynamics, the evolution of the size of one type of population may depend on the sizes of all other populations. Yet, unlike the Lotka-Volterra equations, the object of the modeling is the normalized sizes of populations rather than the size itself. By normalized size of some type, we mean the fraction of that type within the whole population. A basic feature in evolutionary games is, thus, that the evolution of the fraction of a given type in the population depends on the sizes of other types only through the normalized size rather than through their actual one.

The relative rate of the decrease or increase of the normalized population size of some type in the replicator dynamics is what we call fitness and is to be understood in the Darwinian sense. If some type or some behavior increases more than another one, then it has a larger fitness. the evolution of the fitness as described by the replicator dynamics is a central object of study in evolutionary games.

So far we did not actually consider any game and just discussed ways of modeling evolution. The relation to game theory is due to the fact that under some conditions, the fitness converges to some fixed limit, which can be

identified as an equilibrium of a matrix game in which the utilities of the players are the fitnesses. This limit is then called an ESS – evolutionary stable strategy – as defined by Maynard Smith and Price in Maynard Smith and Price (1973). It can be computed using elementary tools in matrix games and then used for predicting the (long term) distribution of behaviors within a population. Note that an equilibrium in a matrix game can be obtained only when the players of the matrix game are rational (each one maximizing its expected utility, being aware of the utilities of other players and of the fact that these players maximize their utilities, etc.). A central contribution of evolutionary games is thus to show that evolution of possibly nonrational populations converges under some conditions to the equilibrium of a game played by rational players. This surprising relationship between the equilibrium of a noncooperative matrix game and the limit points of the fitness dynamics has been supported by a rich body of experimental results; see Friedman (1996).

On the importance of the ESS for understanding the evolution of species, Dawkins writes in his book “The Selfish Gene” (Dawkins 1976): “we may come to look back on the invention of the ESS concept as one of the most important advances in evolutionary theory since Darwin.” He further specifies: “Maynard Smith’s concept of the ESS will enable us, for the first time, to see clearly how a collection of independent selfish entities can come to resemble a single organized whole.”

Here we shall follow the nontraditional approach describing evolutionary games: we shall first introduce the replicator dynamics and then introduce the game theoretic interpretation related to it.

Replicator Dynamics

In the biological context, the replicator dynamics is a differential equation that describes the way in which the usage of strategies changes in time. They are based on the idea that the average

growth rate per individual that uses a given strategy is proportional to the excess of fitness of that strategy with respect to the average fitness.

In engineering, the replicator dynamics could be viewed as a rule for updating mixed strategies by individuals. It is a decentralized rule since it only requires knowing the average utility of the population rather than the strategy of each individual.

Replicator dynamics is one of the most studied dynamics in evolutionary game theory. It has been introduced by Taylor and Jonker (1978). The replicator dynamics has been used for describing the evolution of road traffic congestion in which the fitness is determined by the strategies chosen by all drivers (Sandholm 2009). It has also been studied in the context of the association problem in wireless communications (Shakkottai et al. 2007).

Consider a set of N strategies and let $p_j(t)$ be the fraction of the whole population that uses strategy j at time t . Let $p(t)$ be the corresponding N -dimensional vector. A function f_j is associated with the growth rate of strategy j , and it is assumed to depend on the fraction of each of the N strategies in the population. There are various forms of replicator dynamics (Sandholm 2009) and we describe here the one most commonly used. It is given by

$$\dot{p}_j(t) = \mu p_j(t) \left[f_j(p(t)) - \sum_{k=1}^N p_k(t) f_k(p(t)) \right], \quad (1)$$

where μ is some positive constant and the payoff function f_k is called the fitness of strategy k .

In evolutionary games, evolution is assumed to be due to pairwise interactions between players, as will be described in the next section. Therefore, f_k has the form $f_k(p) = \sum_{i=1}^N J(k, i) p(i)$ where $J(k, i)$ is the fitness of an individual playing k if it interacts with an individual that plays strategy i .

Within quite general settings (Weibull 1995), the above replicator dynamics is known to converge to an ESS (which we introduce in the next section).

Evolutionary Games: Fitnesses

Consider an infinite population of players. Each individual i plays at times t_n^i , $n = 1, 2, 3, \dots$ (assumed to constitute an independent Poisson process with some rate λ) a matrix game against some player $j(n)$ randomly selected within the population. The choice $j(n)$ of the other players at different times is independent. All players have the same finite space of pure strategies (also called actions) K . Each time it plays, a player may use a mixed strategy p , i.e., a probability measure over the set of pure strategies. We consider $J(k, i)$ (defined in the previous section) to be the payoff for a tagged individual if it uses a strategy k , and it interacts with an individual using strategy i . With some abuse of notation, one denotes by $J(p, q)$ the expected payoff for a player who uses a mixed strategy p when meeting another individual who adopts the mixed strategy q . If we define a payoff matrix A and consider p and q to be column vectors, then $J(p, q) = p'Aq$. The payoff function J is indeed linear in p and q . A strategy q is called a Nash equilibrium if

$$\forall p \in \Delta(K), \quad J(q, q) \geq J(p, q) \quad (2)$$

where $\Delta(K)$ is the set of probabilities over the set K .

Suppose that the whole population uses a strategy q and that a small fraction ϵ (called “mutations”) adopts another strategy p . Evolutionary forces are expected to select against p if

$$J(q, \epsilon p + (1 - \epsilon)q) > J(p, \epsilon p + (1 - \epsilon)q). \quad (3)$$

Evolutionary Stable Strategies: ESS

Definition 1 q is said to be an evolutionary stable strategy (ESS) if for every $p \neq q$ there exists some $\bar{\epsilon}_p > 0$ such that (3) holds for all $\epsilon \in (0, \bar{\epsilon}_p)$.

The definition of ESS is thus related to a robustness property against deviations by a whole (possibly small) fraction of the population. This is an important difference that distinguishes the equilibrium in populations as seen by biologists and the standard Nash equilibrium often used in

economics context, in which robustness is defined against the possible deviation of a single user. Why do we need the stronger type of robustness? Since we deal with large populations, it is likely to be expected that from time to time, some group of individuals may deviate. Thus robustness against deviations by a single user is not sufficient to ensure that deviations will not develop and end up being used by a growing portion of the population.

Often ESS is defined through the following equivalent definition.

Theorem 1 (Weibull 1995, Proposition 2.1 or Hofbauer and Sigmund 1998, Theorem 6.4.1, p 63) *A strategy q is said to be an evolutionary stable strategy if and only if $\forall p \neq q$ one of the following conditions holds:*

$$J(q, q) > J(p, q), \quad (4)$$

or

$$J(q, q) = J(p, q) \text{ and } J(q, p) > J(p, p). \quad (5)$$

In fact, if condition (4) is satisfied, then the fraction of mutations in the population will tend to decrease (as it has a lower fitness, meaning a lower growth rate). Thus, the strategy q is then immune to mutations. If it does not but if still the condition (5) holds, then a population using q is “weakly” immune against a mutation using p . Indeed, if the mutant’s population grows, then we shall frequently have individuals with strategy q competing with mutants. In such cases, the condition $J(q, p) > J(p, p)$ ensures that the growth rate of the original population exceeds that of the mutants.

A mixed strategy q that satisfies (4) for all $p \neq q$ is called strict Nash equilibrium. Recall that a mixed strategy q that satisfies (2) for all $p \neq q$ is a Nash equilibrium. We conclude from the above theorem that being a strict Nash equilibrium implies being an ESS, and being an ESS implies being a Nash equilibrium. Note that whereas a mixed Nash equilibrium is known to exist in a matrix game, an ESS may not exist. However, an ESS is known to exist in

evolutionary games where the number of strategies available to each player is 2 (Weibull 1995).

Proposition 1 *In a symmetric game with two strategies for each player and no pure Nash equilibrium, there exists a unique mixed Nash equilibrium which is an ESS.*

Example: The Hawk and Dove Game

We briefly describe the hawk and dove game (Maynard Smith and Price 1973). A bird that searches food finds itself competing with another bird over food and has to decide whether to adopt a peaceful behavior (dove) or an aggressive one (hawk). The advantage of behaving aggressively is that in an interaction with a peaceful bird, the aggressive one gets access to all the food. This advantage comes at a cost: a hawk which meets another hawk ends up fighting with it and thus takes a risk of getting wounded. In contrast, two doves that meet in a contest over food share it without fighting. The fitnesses for player 1 (who chooses a row) are summarized in Table 1, in which the cost for fighting is taken to be some parameter $\delta > 1/2$.

This game has a unique mixed Nash equilibrium (and thus a unique ESS) in which the fraction p of aggressive birds is given by

$$p = \frac{2}{1.5 + \delta}$$

Extension: Evolutionary Stable Sets

Assume that there are two mixed strategies p_i and p_j that have the same performance against each other, i.e., $J(p_i, p_j) = J(p_j, p_j)$. Then neither one of them can be an ESS, even if they are quite robust against other strategies. Now assume that when excluding one of them from the set

of mixed strategies, the other one is an ESS. This could imply that different combinations of these two ESS's could coexist and would together be robust to any other mutations. This motivates the following definition of an ESSet (Cressman 2003):

Definition 2 A set E of symmetric Nash equilibria is an evolutionarily stable set (ESSet) if, for all $q \in E$, we have $J(q, p) > J(p, p)$ for all $p \notin E$ and such that $J(p, q) = J(q, q)$.

Properties of ESSet:

- (i) For all p and p' in an ESSet E , we have $J(p', p) = J(p, p)$.
- (ii) If a mixed strategy is an ESS, then the singleton containing that mixed strategy is an ESSet.
- (iii) If the ESSet is not a singleton, then there is no ESS.
- (iv) If a mixed strategy is in an ESSet, then it is a Nash equilibrium (see Weibull 1995, p. 48, Example 2.7).
- (v) Every ESSet is a disjoint union of Nash equilibria.
- (vi) A perturbation of a mixed strategy which is in the ESSet can move the system to another mixed strategy in the ESSet. In particular, every ESSet is asymptotically stable for the replicator dynamics (Cressman 2003).

Summary and Future Directions

The entry has provided an overview of the foundations of evolutionary games which include the ESS (evolutionary stable strategy) equilibrium concept that is stronger than the standard Nash equilibrium and the modeling of the dynamics of the competition through the replicator dynamics. Evolutionary game framework is a first step in linking game theory to evolutionary processes. The payoff of a player is identified as its fitness, i.e., the rate of reproduction. Further development of this mathematical tool is needed for handling hierarchical fitness, i.e., the cases where the individual that interacts cannot be directly

Evolutionary Games, Table 1 The hawk-dove game

	H	D
H	$1/2 - \delta$	1
D	0	$1/2$

identified with the reproduction as it is part of a larger body. For example, the behavior of a blood cell in the human body when interacting with a virus cannot be modeled as directly related to the fitness of the blood cell but rather to that of the human body. A further development of the theory of evolutionary games is needed to define meaningful equilibrium notions and relate them to replication in such contexts.

Cross-References

- [Dynamic Noncooperative Games](#)
- [Game Theory: A General Introduction and a Historical Overview](#)

Recommended Reading

Several books cover evolutionary game theory well. These include Cressman (2003), Hofbauer and Sigmund (1998), Sandholm (2009), Vincent and Brown (2005), and Weibull (1995). In addition, the book *The Selfish Gene* by Dawkins presents an excellent background on evolution in biology.

Acknowledgments The author has been supported by CONGAS project FP7-ICT-2011-8-317672, see www.congas-project.eu.

Bibliography

- Altman E (2008) Semi-linear stochastic difference equations. *Discret Event Dyn Syst* 19:115–136
- Cressman R (2003) *Evolutionary dynamics and extensive form games*. MIT, Cambridge
- Dawkins R (1976) *The selfish gene*. Oxford University Press, Oxford
- Friedman D (1996) Equilibrium in evolutionary games: some experimental results. *Econ J* 106: 1–25
- Hofbauer J, Sigmund K (1998) *Evolutionary games and population dynamics*. Cambridge University Press, Cambridge/New York
- Lotka-Volterra AJ (1910) Contribution to the theory of periodic reaction. *J Phys Chem* 14(3): 271–274
- Maynard Smith J, Price GR (1973) The logic of animal conflict. *Nature* 246(5427):15–18
- Sandholm WH (2009) *Population games and evolutionary dynamics*. MIT
- Shakkottai S, Altman E, Kumar A (2007) Multihoming of users to access points in WLANs: a population game perspective. *IEEE J Sel Areas Commun Spec Issue Non-Coop Behav Netw* 25(6):1207–1215
- Taylor P, Jonker L (1978) Evolutionary stable strategies and game dynamics. *Math Biosci* 16: 76–83
- Vincent TL, Brown JS (2005) *Evolutionary game theory, natural selection & Darwinian dynamics*. Cambridge University Press, Cambridge
- Watson HW, Galton F (1875) On the probability of the extinction of families. *J Anthropol Inst Great Br* 4: 138–144
- Weibull JW (1995) *Evolutionary game theory*. MIT, Cambridge

Experiment Design and Identification for Control

Håkan Hjalmarsson

School of Electrical Engineering and Computer Science, Centre for Advanced Bioproduction, KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

The experimental conditions have a major impact on the estimation result. Therefore, the available degrees of freedom in this respect that are at the disposal to the user should be used wisely. This entry provides the fundamentals for the available techniques. We also briefly discuss the particulars of identification for model-based control, one of the main applications of system identification.

Keywords

Adaptive experiment design · Application-oriented experiment design · Cramér-Rao lower bound · Crest factor · Experiment design · Fisher information matrix · Identification for control · Least-costly identification · Multisine · Pseudorandom binary signal (PRBS) · Robust experiment design

Introduction

The accuracy of an identified model is governed by:

- (i) Information content in the data used for estimation
- (ii) The complexity of the model structure

The former is related to the noise properties and the “energy” of the external excitation of the system and how it is distributed. In regard to (ii), a model structure which is not flexible enough to capture the true system dynamics will give rise to a systematic error, while an overly flexible model will be overly sensitive to noise (so-called overfitting). The model complexity is closely associated with the number of parameters used. For a linear model structure with n parameters modeling the dynamics, it follows from the invariance result in Rojas et al. (2009) that to obtain a model for which the variance of the frequency function estimate is less than $1/\gamma$ over all frequencies, the signal-to-noise ratio, as measured by input energy over noise variance, must be at least $n\gamma$. With energy being power $\times$ time and as input power is limited in physical systems, this indicates that the experiment time grows at least linearly with the number of model parameters. When the input energy budget is limited, the only way around this problem is to sacrifice accuracy over certain frequency intervals. The methodology to achieve this in a systematic way is known as experiment design.

Model Quality Measures

The Cramér-Rao bound provides a lower bound on the covariance matrix of the estimation error for an unbiased estimator. With $\hat{\theta}_N \in \mathbb{R}^n$ denoting the parameter estimate (based on N input-output samples) and θ_o the true parameters,

$$NE \left[\left(\hat{\theta}_N - \theta_o \right) \left(\hat{\theta}_N - \theta_o \right)^T \right] \geq N I_F^{-1}(\theta_o, N) \quad (1)$$

where $I_F(\theta_o, N) \in \mathbb{R}^{n \times n}$ appearing in the lower bound is the so-called Fisher information matrix (Ljung 1999). For consistent estimators, i.e., when $\hat{\theta}_N \rightarrow \theta_o$ as $N \rightarrow \infty$, the inequality (1) typically holds asymptotically as the sample size N grows to infinity. The right-hand side in (1) is then replaced by the inverse of the per sample Fisher information $I_F(\theta_o) := \lim_{N \rightarrow \infty} I_F(\theta_o, N)/N$. An estimator is said to be asymptotically efficient if equality is reached in (1) as $N \rightarrow \infty$.

Even though it is possible to reduce the mean-square error by constraining the model flexibility appropriately, it is customary to use consistent estimators since the theory for biased estimators is still not well understood. For such estimators, using some function of the Fisher information as performance measure is natural.

General-Purpose Quality Measures

Over the years a number of “general-purpose” quality measures have been proposed. Perhaps the most frequently used is the determinant of the inverse Fisher information. This represents the volume of confidence ellipsoids for the parameter estimates and minimizing this measure is known as D-optimal design. Two other criteria relating to confidence ellipsoids are E-optimal design, which uses the length of the longest principal axis (the minimum eigenvalue of I_F) as quality measure, and A-optimal design, which uses the sum of the squared lengths of the principal axes (the trace of I_F^{-1}).

Application-Oriented Quality Measures

When demands are high and/or experimentation resources are limited, it is necessary to tailor the experiment carefully according to the intended use of the model. Below we will discuss a couple of closely related application-oriented measures.

Average Performance Degradation

Let $V_{\text{app}}(\theta) \geq 0$ be a measure of how well the model corresponding to parameter θ performs when used in the application. In finance, V_{app} can, e.g., represent the ability to predict the stock market. In process industry, V_{app} can

represent the profit gained using a feedback controller based on the model corresponding to θ . Let us assume that V_{app} is normalized such that $\min_{\theta} V_{\text{app}}(\theta) = V_{\text{app}}(\theta_o) = 0$. That V_{app} has minimum corresponding to the parameters of the true system is quite natural. We will call V_{app} the application cost. Assuming that the estimator is asymptotically efficient, using a second-order Taylor approximation gives that the average application cost can be expressed as (the first-order term vanishes since θ_o is the minimizer of V_{app})

$$\begin{aligned} E[V_{\text{app}}(\hat{\theta}_N)] &\approx \frac{1}{2} E\left[(\hat{\theta}_N - \theta_o)^T V''_{\text{app}}(\theta_o) (\hat{\theta}_N - \theta_o) \right] \\ &= \frac{1}{2N} \text{Tr} \left\{ V''_{\text{app}}(\theta_o) I_F^{-1}(\theta_o) \right\} \end{aligned} \quad (2)$$

This is a generalization of the A-optimal design measure, and its minimization is known as L-optimal design.

Acceptable Performance

Alternatively, one may define a set of acceptable models, i.e., a set of models which will give acceptable performance when used in the application. With a performance degradation measure defined of the type V_{app} above, this would be a level set

$$\mathcal{E}_{\text{app}} = \left\{ \theta : V_{\text{app}}(\theta) \leq \frac{1}{\gamma} \right\} \quad (3)$$

for some constant $\gamma > 0$. The objective of the experiment design is then to ensure that the resulting estimate ends up in $\mathcal{E}_{\text{app}}$ with high probability.

Design Variables

In an identification experiment, there are a number of design variables at the user's disposal. Below we discuss three of the most important ones.

Sampling Interval

For the sampling interval, the general advice from an information theoretic point of view is to sample as fast as possible (Ljung 1999). However, sampling much faster than the time constants of the system may lead to numerical issues when estimating discrete time models as there will be poles close to the unit circle. Downsampling may thus be required.

Feedback

Generally speaking, feedback has three effects from an identification and experiment design point of view:

- (i) Not all the power in the input can be used to estimate the system dynamics when a noise model is estimated as a part of the input signal has to be used for the latter task; see Section 8.1 in Forssell and Ljung (1999). When a very flexible noise model is used, the estimate of the system dynamics then has to rely almost entirely on external excitation.
- (ii) Feedback can reduce the effect of disturbances and noise at the output. When there are constraints on the outputs, this allows for larger (input) excitation and therefore more informative experiments.
- (iii) The cross-correlation between input and noise/disturbances requires good noise models to avoid biased estimates (Ljung 1999).

Strictly speaking, (i) is only valid when the system and noise models are parametrized separately. Items (i) and (ii) imply that when there are constraints on the input only, then the optimal design is always in open loop, whereas for output constrained only problems, the experiment should be conducted in closed loop (Agüero and Goodwin 2007).

External Excitation Signals

The most important design variable is the external excitation, including the length of the experiment. Even for moderate experiment lengths, solving optimal experiment design problems with

respect to the entire excitation sequence can be a formidable task. Fortunately, for experiments of reasonable length, the design can be split up in two steps:

- (i) First, optimization of the probability density function of the excitation
- (ii) Generation of the actual sequence from the obtained density function through a stochastic simulation procedure

More details are provided in section “[Computational Issues](#).”

Experimental Constraints

An experiment is always subject to constraints, physical as well as economical. Such constraints are typically translated into constraints on the following signal properties:

- (i) *Variability*. For example, too high level of excitation may cause the end product to go off-spec, resulting in product waste and associated high costs.
- (ii) *Frequency content*. Often, too harsh movements of the inputs may damage equipment.
- (iii) *Amplitudes*. For example, actuators have limited range, restricting input amplitudes.
- (iv) *Waveforms*. In process industry, it is not uncommon that control equipment limits the type of signals that can be applied. In other applications, it may be physically possible to realize only certain types of excitation. See section “[Waveform Generation](#)” for further discussion.

It is also often desired to limit the experiment time so that the process may go back to normal operation, reducing, e.g., cost of personnel. The latter is especially important in the process industry where dynamics are slow. The above type of constraints can be formulated as constraints on the design variables in section “[Design Variables](#)” and associated variables.

Experiment Design Criteria

There are two principal ways to define an optimal experiment design problem:

- (i) *Best effort*. Here the best quality as, e.g., given by one of the quality measures in section “[Model Quality Measures](#)” is sought under constraints on the experimental effort and cost. This is the classical problem formulation.
- (ii) *Least-costly*. The cheapest experiment is sought that results in a predefined model quality. Thus, as compared to best effort design, the optimization criterion and constraint are interchanged. This type of design was introduced by Bombois and coworkers; see Bombois et al. (2006).

As shown in Rojas et al. (2008), the two approaches typically lead to designs only differing by a scaling factor.

Computational Issues

The optimal experiment design problem based on the Fisher information is typically non-convex. For example, consider a finite-impulse response model subject to an experiment of length N with the measured outputs collected in the vector

$$Y = \Phi\theta + E, \quad \Phi = \begin{bmatrix} u(0) & \dots & u(-(n-1)) \\ \vdots & \vdots & \vdots \\ u(N-1) & \dots & u(N-n) \end{bmatrix}$$

where $E \in \mathbb{R}^N$ is zero-mean Gaussian noise with covariance matrix $\sigma^2 I_{N \times N}$. Then it holds that

$$I_F(\theta_o, N) = \frac{1}{\sigma^2} \Phi^T \Phi \quad (4)$$

From an experiment design point of view, the input vector $u = [u(-(n-1)) \dots u(N)]^T$ is the design variable, but with the elements of $I_F(\theta_o, N)$ being a quadratic function of the input

sequence, all typical quality measures become non-convex.

While various methods for non-convex numerical optimization can be used to solve such problems, they often encounter problems with, e.g., local minima. To address this, a number of techniques have been developed either where the problem is reparametrized so that it becomes convex or where a convex approximation is used. The latter technique is called convex relaxation and is often based on a reparametrization as well. We use the example above to provide a flavor of the different techniques.

Reparametrization

If the input is constrained to be periodic so that $u(t) = u(t + N)$, $t = -n, \dots, -1$, it follows that the Fisher information is linear in the sample correlations of the input. Using these as design variables instead of u results in that all quality measures referred to above become convex functions.

This reparametrization thus results in the two-step procedure discussed in section “[External Excitation Signals](#)”: First, the sample correlations are obtained from an optimal experiment design problem, and then an input sequence is generated that has this sample correlation. In the second step, there is a considerable freedom. Notice, however, that since correlations do not directly relate to the actual amplitudes of the resulting signals, it is difficult to incorporate waveform constraints in this approach. On the contrary, variance constraints are easy to incorporate.

Convex Relaxations

There are several approaches to obtain convex relaxations.

Using the per Sample Fisher Information

If the input is a realization of a stationary random process and the sample size N is large enough, $I_F(\theta_o, N)/N$ is approximately equal to the per sample Fisher matrix which only depends on the correlation sequence of the input. Using this approximation, one can now follow the same procedure as in the reparametrization approach and first optimize the input correlation sequence.

The generation of a stationary signal with a certain correlation is a stochastic realization problem which can be solved using spectral factorization followed by filtering white noise sequence, i.e., a sequence of independent identically distributed random variables, through the (stable) spectral factor (Jansson and Hjalmarsson 2005).

More generally, it turns out that the per sample Fisher information for linear models/systems only depends on the joint input/noise spectrum (or the corresponding correlation sequence). A linear parametrization of this quantity thus typically leads to a convex problem (Jansson and Hjalmarsson 2005).

The set of all spectra is infinite dimensional, and this precludes a search over all possible spectra. However, since there is a finite-dimensional parametrization of the per sample Fisher information (it is a symmetric $n \times n$ matrix), it is also possible to find finite-dimensional sets of spectra that parametrize all possible per sample Fisher information matrices. Multisine with appropriately chosen frequencies is one possibility. However, even though all per sample Fisher information matrices can be generated, the solution may be suboptimal depending on which constraints the problem contains.

The situation for nonlinear problems is conceptually the same, but here the entire probability density function of the stationary process generating the input plays the same role as the spectrum in the linear case. This is a much more complicated object to parametrize.

Lifting

An approach that can deal with amplitude constraints is based on a so-called lifting technique: Introduce the matrix $U = uu^T$, representing all possible products of the elements of u . This constraint is equivalent to

$$\begin{bmatrix} U & u \\ u^T & 1 \end{bmatrix} \succeq 0, \quad \text{rank} \begin{bmatrix} U & u \\ u^T & 1 \end{bmatrix} = 1 \quad (5)$$

The idea of lifting is now to observe that the Fisher information matrix is linear in the elements of U and by dropping the rank constraint in (5) a convex relaxation is obtained, where both

U and u (subject to the matrix inequality in (5)) are decision variables.

Frequency-by-Frequency Design

An approximation for linear systems that allows frequency-by-frequency design of the input spectrum and feedback is obtained by assuming that the model is of high order. Then the variance of an n th-order estimate, $G(e^{i\omega}, \hat{\theta}_N)$, of the frequency function can approximately be expressed as

$$\text{Var } G(e^{i\omega}, \hat{\theta}_N) \approx \frac{n}{N} \frac{\Phi_v(\omega)}{\Phi_u(\omega)} \quad (6)$$

([System Identification: An Overview](#)) in the open-loop case (there is a closed-loop extension as well), where Φ_u and Φ_v are the input and noise spectra, respectively. Performance measures of the type (2) can then be written as

$$\int_{-\pi}^{\pi} W(e^{i\omega}) \frac{\Phi_v(\omega)}{\Phi_u(\omega)} d\omega$$

where the weighting $W(e^{i\omega}) \geq 0$ depends on the application. When only variance constraints are present, such problems can be solved frequency by frequency, providing both simple calculations and insight into the design.

Implementation

We have used the notation $I_F(\theta_o, N)$ to indicate that the Fisher information typically (but not always) depends on the parameter corresponding to the true system. That the optimal design depends on the to-be identified system is a fundamental problem in optimal experiment design. There are two basic approaches to address this problem which are covered below. Another important aspect is the choice of waveform for the external excitation signal. This is covered last in this section.

Robust Experiment Design

In robust experiment design, it is assumed that it is known beforehand that the true parameter belongs to some set, i.e., $\theta_o \in \Theta$. A minimax

approach is then typically taken, finding the experiment that minimizes the worst performance over the set Θ . Such optimization problems are computationally very difficult.

Adaptive Experiment Design

The alternative to robust experiment design is to perform the design adaptively or sequentially, meaning that first a design is performed based on some initial “guess” of the true parameter, and then as samples are collected, the design is revised taking advantage of the data information. Interestingly, the convergence rate of the parameter estimate is typically sufficiently fast that for this approach the asymptotic distribution is the same as for the design based on the true model parameter (Hjalmarsson 2009).

Waveform Generation

We have argued above that it is the spectrum of the excitation (together with the feedback) that determines the achieved model accuracy in the linear time-invariant case. In section “[Using the per Sample Fisher Information](#),” we argued that a signal with a particular spectrum can be obtained by filtering a white noise sequence through a stable spectral factor of the desired spectrum. However, we have also in section “[Experimental Constraints](#)” argued that particular applications may require particular waveforms. We will here elaborate further on how to generate a waveform with desired characteristics.

From an accuracy point of view, there are two general issues that should be taken into account when the waveform is selected:

- *Persistence of excitation.* A signal with a spectrum having n nonzero frequencies (on the interval $(-\pi, \pi]$) can be used to estimate at most n parameters. Thus, as is typically the case, if there is uncertainty regarding which model structure to use before the experiment, one has to ensure that a sufficient number of frequencies is excited.
- *The crest factor.* For all systems, the maximum input amplitude, say A , is constrained. To deal with this from an experiment design point of

view, it is convenient to introduce what is called the crest factor of a signal:

$$C_r^2 = \frac{\max_t u^2(t)}{\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N u^2(t)}$$

The crest factor is thus the ratio between the squared maximum amplitude and the power of the signal. Now, for a class of signal waveforms with a given crest factor, the input power that can be used is upper-bounded by

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N u^2(t) \leq \frac{A^2}{C_r^2} \quad (7)$$

However, the power is the integral of the signal spectrum, and since increasing the amplitude of the input signal spectrum will increase a model's accuracy, cf. (6), it is desirable to use as much signal power as possible. By (7) we see that this means that waveforms with low crest factor should be used.

A lower bound for the crest factor is readily seen to be 1. This bound is achieved for binary symmetric signals. Unfortunately, there exists no systematic way to design a binary sequence that has a prescribed spectrum. However, the so-called arcsin law may be used. It states that the sign of a zero-mean Gaussian process with correlation sequence r_τ gives a binary signal having correlation sequence $\tilde{r}_\tau = 2/\pi \arcsin(r_\tau)$. With $\tilde{r}_\tau$ given, one can try to solve this relation for the corresponding r_τ .

A crude, but often sufficient, method to generate binary sequences with desired spectral content is based on the use of *pseudorandom binary signals* (PRBS). Such signals (which are generated by a shift register) are periodic signals which have correlation sequences similar to random white noise, i.e., a flat spectrum. By resampling such sequences, the spectrum can be modified. It should be noted that binary sequences are less attractive when it comes to identifying nonlinearities. This is easy to understand by considering a static system. If only one amplitude of the input is

used, it will be impossible to determine whether the system is nonlinear or not.

A PRBS is a periodic signal and can therefore be split into its Fourier terms. With a period of M , each such term corresponds to one frequency on the grid $2\pi k/M$, $k = 0, \dots, M-1$. Such a signal can thus be used to estimate at most M parameters. Another way to generate a signal with period M is to add sinusoids corresponding to the above frequencies, with desired amplitudes. A periodic signal generated in this way is commonly referred to as a *Multisine*. The crest factor of a multisine depends heavily on the relation between the phases of the sinusoids, times the number of sinusoids. It is possible to optimize the crest factor with respect to the choice of phases (Rivera et al. 2009). There exist also simple deterministic methods for choosing phases that give a good crest factor, e.g., Schroeder phasing. Alternatively, phases can be drawn randomly and independently, giving what is known as random-phase multisines (Pintelon and Schoukens 2012), a family of random signals with properties similar to Gaussian signals. Periodic signals have some useful features:

- *Estimation of nonlinearities.* A linear time-invariant system responds to a periodic input signal with a signal consisting of the same frequencies but with different amplitudes and phases. Thus, it can be concluded that the system is nonlinear if the output contains other frequencies than the input. This can be explored in a systematic way to estimate also the nonlinear part of a system.
- *Estimation of noise variance.* For a linear time-invariant system, the difference in the output between different periods is due entirely to the noise if the system is in steady state. This can be used to devise simple methods to estimate the noise level.
- *Data compression.* By averaging measurements over different periods, the noise level can be reduced at the same time as the number of measurements is reduced.

Further details on waveform generation and general-purpose signals useful in system

identification can be found in Pintelon and Schoukens (2012) and Ljung (1999).

Implications for the Identification Problem Per Se

In order to get some understanding of how optimal experimental conditions influence the identification problem, let us return to the finite-impulse response model example in section “Computational Issues.” Consider a least-costly setting with an acceptable performance constraint. More specifically, we would like to use the minimum input energy that ensures that the parameter estimate ends up in a set of the type (3). An approximate solution to this is that a 99 % confidence ellipsoid for the resulting estimate is contained in $\mathcal{E}_{\text{app}}$. Now, it can be shown that a confidence ellipsoid is a level set for the average least-squares cost $E[V_N(\theta)] = E[\|Y - \Phi\theta\|^2] = \|\theta - \theta_o\|_{\Phi^T \Phi}^2 + \sigma^2$. Assuming the application cost V_{app} also is quadratic in θ , it follows after a little bit of algebra (see Hjalmarsson 2009) that it must hold that

$$E[V_N(\theta)] \geq \sigma^2 (1 + \gamma c V_{\text{app}}(\theta)), \quad \forall \theta \quad (8)$$

for a constant c that is not important for our discussion. The value of $E[V_N(\theta)] = \|\theta - \theta_o\|_{\Phi^T \Phi}^2 + \sigma^2$ is determined by how large the weighting $\Phi^T \Phi$ is, which in turn depends on how large the input u is. In a least-costly setting with the energy $\|u\|^2$ as criterion, the best solution would be that we have equality in (8). Thus we see that optimal experiment design tries to shape the identification criterion after the application cost. We have the following implications of this result:

- (i) *Perform identification under appropriate scaling of the desired operating conditions.* Suppose that $V_{\text{app}}(\theta)$ is a function of how the system outputs deviate from a desired trajectory (determined by θ_o). Performing an experiment which performs the desired trajectory then gives that the sum of the squared prediction errors are an approximation of $V_{\text{app}}(\theta)$, at least for

parameters close to θ_o . Obtaining equality in (8) typically requires an additional scaling of the input excitation or the length of the experiment. The result is intuitively appealing: The desired operating conditions should reveal the system properties that are important in the application.

- (ii) *Identification cost for application performance.* We see that the required energy grows (almost) linearly with γ , which is a measure of how close to the ideal performance (using the true parameter θ_o) we want to come. Furthermore, it is typical that as the performance requirements in the application increase, the sensitivity to model errors increases. This means that $V_{\text{app}}(\theta)$ increases, which thus in turn means that the identification cost increases. In summary, the identification cost will be higher, the higher performance that is required in the application. The inequality (8) can be used to quantify this relationship.
- (iii) *Model structure sensitivity.* As V_{app} will be sensitive to system properties important for the application, while insensitive to system properties of little significance, with the identification criterion V_N matched to V_{app} , it is only necessary that the model structure is able to model the important properties of the system.

In any case, whatever model structure that is used, the identified model will be the best possible in that structure for the intended application. This is very different from an arbitrary experiment where it is impossible to control the model fit when a model of restricted complexity is used.

We conclude that optimal experiment design simplifies the overall system identification problem.

Identification for Control

Model-based control is one of the most important applications of system identification. Robust control ensures performance and stability in the presence of model uncertainty. However, the

majority of such design methods do not employ the parametric ellipsoidal uncertainty sets resulting from standard system identification. In fact only in the last decade analysis and design tools for such type of model uncertainty have started to emerge, e.g., Raynaud et al. (2000) and Gevers et al. (2003).

The advantages of matching the identification criterion to the application have been recognized since long in this line of research. For control applications this typically implies that the identification experiment should be performed under the same closed-loop operation conditions as the controller to be designed. This was perhaps first recognized in the context of minimum variance control (see Gevers and Ljung 1986) where variance errors were the concern. Later on this was recognized to be the case also for the bias error, although here pre-filtering can be used to achieve the same objective.

To account for that the controller to be designed is not available, techniques where control and identification are iterated have been developed, cf. adaptive experiment design in section “[Adaptive Experiment Design](#).” Convergence of such schemes has been established when the true system is in the model set but has proved out of reach for models of restricted complexity.

In recent years, techniques integrating experiment design and model predictive control have started to appear. A general-purpose design criterion is used in Rathouský and Havlena (2013), while Larsson et al. (2013) uses an application-oriented criterion.

Summary and Future Directions

When there is the “luxury” to design the experiment, then this opportunity should be seized by the user. Without informative data there is little that can be done. In this exposé we have outlined the techniques that exist but also emphasized that a well-conceived experiment, reflecting the intended application, significantly can simplify the overall system identification problem.

Further developments of computational techniques are high on the agenda, e.g., how to handle

time-domain constraints and nonlinear models. To this end, developments in optimization methods are rapidly being incorporated. While, as reported in Hjalmarsson (2009), there are some results on how the identification cost depends on the performance requirements in the application, further understanding of this issue is highly desirable. Theory and further development of the emerging model predictive control schemes equipped with experiment design may very well be the direction that will have most impact in practice. Nonparametric kernel methods have proven to be highly competitive estimation methods and developments of experiment design methods for such estimators only started recently in the system identification community; see, e.g., Fujimoto et al. (2018).

Cross-References

► [System Identification: An Overview](#)

Recommended Reading

A classical text on optimal experiment design is Fedorov (1972). The textbooks Goodwin and Payne (1977) and Zarrop (1979) cover this theory adapted to a dynamical system framework. A general overview is provided in Pronzato (2008). A semi-definite programming framework based on the per sample Fisher information is provided in Jansson and Hjalmarsson (2005). The least-costly framework is covered in Bombois et al. (2006). The lifting technique was introduced for input design in Manchester (2010). Details of the frequency-by-frequency design approach can be found in Ljung (1999). References to robust and adaptive experiment design can be found in Pronzato (2008) and Hjalmarsson (2009). For an account of the implications of optimal experiment design for the system identification problem as a whole, see Hjalmarsson (2009). Thorough accounts of the developments in identification for control are provided in Hjalmarsson (2005) and Gevers (2005).

Acknowledgments This work was supported by the European Research Council under the advanced grant LEARN, contract 267381, and by the Swedish Research Council, contracts 2016-06079.

Bibliography

- Agüero JC, Goodwin GC (2007) Choosing between open and closed loop experiments in linear system identification. *IEEE Trans Autom Control* 52(8):1475–1480
- Bombois X, Scorletti G, Gevers M, Van den Hof PMJ, Hildebrand R (2006) Least costly identification experiment for control. *Automatica* 42(10):1651–1662
- Fedorov VV (1972) Theory of optimal experiments. Probability and mathematical statistics, vol 12. Academic, New York
- Forssell U, Ljung L (1999) Closed-loop identification revisited. *Automatica* 35:1215–1241
- Fujimoto Y, Maruta I, Sugie T (2018) Input design for kernel-based system identification from the viewpoint of frequency response. *IEEE Trans Autom Control* 63(9):3075–3082
- Gevers M (2005) Identification for control: from the early achievements to the revival of experiment design. *Eur J Control* 11(4–5):335–352. Semi-plenary lecture at IEEE conference on decision and control – European control conference
- Gevers M, Ljung L (1986) Optimal experiment designs with respect to the intended model application. *Automatica* 22(5):543–554
- Gevers M, Bombois X, Codrons B, Scorletti G, Anderson BDO (2003) Model validation for control and controller validation in a prediction error identification framework – part I: theory. *Automatica* 39(3):403–445
- Goodwin GC, Payne RL (1977) Dynamic system identification: experiment design and data analysis. Academic, New York
- Hjalmarsson H (2005) From experiment design to closed loop control. *Automatica* 41(3):393–438
- Hjalmarsson H (2009) System identification of complex and structured systems. *Eur J Control* 15(4):275–310. Plenary address. European control conference
- Jansson H, Hjalmarsson H (2005) Input design via LMIs admitting frequency-wise model specifications in confidence regions. *IEEE Trans Autom Control* 50(10):1534–1549
- Larsson CA, Hjalmarsson H, Rojas CR, Bombois X, Mesbah A, Modén P-E (2013) Model predictive control with integrated experiment design for output error systems. In: European control conference, Zurich
- Ljung L (1999) System identification: theory for the user, 2nd edn. Prentice-Hall, Englewood Cliffs
- Manchester IR (2010) Input design for system identification via convex relaxation. In: 49th IEEE conference on decision and control, Atlanta, pp 2041–2046
- Pintelon R, Schoukens J (2012) System identification: a frequency domain approach, 2nd edn. Wiley/IEEE, Hoboken/Piscataway
- Pronzato L (2008) Optimal experimental design and some related control problems. *Automatica* 44(2):303–325
- Rathouský J, Havlena V (2013) MPC-based approximate dual controller by information matrix maximization. *Int J Adapt Control Signal Process* 27(11):974–999
- Raynaud HF, Pronzato L, Walter E (2000) Robust identification and control based on ellipsoidal parametric uncertainty descriptions. *Eur J Control* 6(3):245–255
- Rivera DE, Lee H, Mittelman HD, Braun MW (2009) Constrained multisine input signals for plant-friendly identification of chemical process systems. *J Process Control* 19(4):623–635
- Rojas CR, Agüero JC, Welsh JS, Goodwin GC (2008) On the equivalence of least costly and traditional experiment design for control. *Automatica* 44(11):2706–2715
- Rojas CR, Welsh JS, Agüero JC (2009) Fundamental limitations on the variance of parametric models. *IEEE Trans Autom Control* 54(5):1077–1081
- Zarrop M (1979) Optimal experiment design for dynamic system identification. Lecture notes in control and information sciences, vol 21. Springer, Berlin

Explicit Model Predictive Control

Alberto Bemporad

IMT Institute for Advanced Studies Lucca,
Lucca, Italy

Abstract

Model predictive control (MPC) has been used in the process industries for more than 40 years because of its ability to control multivariable systems in an optimized way under constraints on input and output variables. Traditionally, MPC requires the solution of a quadratic program (QP) online to compute the control action, sometimes restricting its applicability to slow processes. Explicit MPC completely removes the need for online solvers by precomputing the control law off-line, so that online operations reduce to a simple function evaluation. Such a function is piecewise affine in most cases, so that the MPC controller is equivalently

expressed as a lookup table of linear gains, a form that is extremely easy to code, requires only basic arithmetic operations, and requires a maximum number of iterations that can be exactly computed a priori.

Keywords

Constrained control · Embedded optimization · Model predictive control · Multiparametric programming · Quadratic programming

Introduction

Model predictive control (MPC) is a well-known methodology for synthesizing feedback control laws that optimize closed-loop performance subject to prespecified operating constraints on inputs, states, and outputs (Mayne and Rawlings 2009; Borrelli et al. 2017). In MPC, the control action is obtained by solving a finite horizon open-loop optimal control problem at each sampling instant. Each optimization yields a sequence of optimal control moves, but only the first move is applied to the process: At the next time step, the computation is repeated over a shifted time horizon by taking the most recently available state information as the new initial condition of the new optimal control problem. For this reason, MPC is also called “receding horizon control.” In most practical applications, MPC is based on a linear discrete-time time-invariant model of the controlled system and quadratic penalties on tracking errors and actuation efforts; in such a formulation, the optimal control problem can be recast as a quadratic programming (QP) problem, whose linear term of the cost function and right-hand side of the constraints depend on a vector of parameters that may change from one step to another (such as the current state and reference signals). To enable the implementation of MPC in real industrial products, a QP solution method must be embedded in the control hardware. The method must be fast enough to provide a solution within short sampling intervals and require

simple hardware, limited memory to store the data defining the optimization problem and the code implementing the algorithm itself, a simple program code, and good worst-case estimates of the execution time to meet real-time system requirements.

Several *online* solution algorithms have been studied for embedding quadratic optimization in control hardware, such as active-set methods (Ricker 1985; Bemporad 2016), interior-point methods (Wang and Boyd 2010), fast gradient-projection methods (Patrinos and Bemporad 2014), and the alternating direction method of multipliers (Banjac et al. 2017). Explicit MPC takes a different approach to meet the above requirements, where multiparametric quadratic programming is proposed to pre-solve the QP *off-line*, therefore converting the MPC law into a continuous and piecewise-affine function of the parameter vector (Bemporad et al. 2002b; Bemporad 2015). We review the main ideas of explicit MPC in the next section, referring the reader to Alessio and Bemporad (2009) for a more complete survey paper on explicit MPC.

Model Predictive Control Problem

Consider the following finite-time optimal control problem formulation for MPC:

$$V^*(p) = \min_z \ell_N(x_N) + \sum_{k=0}^{N-1} \ell(x_k, u_k) \quad (1a)$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k \quad (1b)$$

$$C_x x_k + C_u u_k \leq c \quad (1c)$$

$$k = 0, \dots, N-1$$

$$C_N x_N \leq c_N \quad (1d)$$

$$x_0 = x \quad (1e)$$

where N is the prediction horizon; $x \in \mathbb{R}^m$ is the current state vector of the controlled system; $u_k \in \mathbb{R}^{n_u}$ is the vector of manipulated variables at prediction time k , $k = 0, \dots, N-1$; $z \triangleq$

$[u'_0 \dots u'_{N-1}]' \in \mathbb{R}^n$, $n \triangleq n_u N$, is the vector of decision variables to be optimized;

$$\ell(x, u) = \frac{1}{2}x'Qx + u'Ru \quad (2a)$$

$$\ell_N(x) = \frac{1}{2}x'Px \quad (2b)$$

are the stage cost and terminal cost, respectively; Q , P are symmetric and positive semidefinite matrices; and R is a symmetric and positive definite matrix.

Let $n_c \in \mathbb{N}$ be the number of constraints imposed at prediction time $k = 0, \dots, N-1$, namely, $C_x \in \mathbb{R}^{n_c \times m}$, $C_u \in \mathbb{R}^{n_c \times n_u}$, and $c \in \mathbb{R}^{n_c}$, and let n_N be the number of terminal constraints, namely, $C_N \in \mathbb{R}^{n_N \times m}$ and $c_N \in \mathbb{R}^{n_N}$. The total number q of linear inequality constraints imposed in the MPC problem formulation (1) is $q = Nn_c + n_N$.

By eliminating the states $x_k = A^k x + \sum_{j=0}^{k-1} A^j B u_{k-1-j}$ from problem (1), the optimal control problem (1) can be expressed as the convex quadratic program (QP):

$$V^*(x) \triangleq \min_z \quad \frac{1}{2}z'H z + x'F'z + \frac{1}{2}x'Yx \quad (3a)$$

$$\text{s.t.} \quad Gz \leq W + Sx \quad (3b)$$

where $H = H' \in \mathbb{R}^{n \times n}$ is the Hessian matrix; $F \in \mathbb{R}^{n \times m}$ defines the linear term of the cost function; $Y \in \mathbb{R}^{m \times m}$ has no influence on the optimizer, as it only affects the optimal value of (3a); and the matrices $G \in \mathbb{R}^{q \times n}$, $S \in \mathbb{R}^{q \times m}$, $W \in \mathbb{R}^q$ define in a compact form the constraints imposed in (1). Because of the assumptions made on the weight matrices Q , R , and P , matrix H is positive definite, and matrix $\begin{bmatrix} H & F' \\ F & Y \end{bmatrix}$ is positive semidefinite.

The MPC control law is defined by setting

$$u(x) = [I \ 0 \ \dots \ 0]z(x) \quad (4)$$

where $z(x)$ is the optimizer of the QP problem (3) for the current value of x and I is the identity matrix of dimension $n_u \times n_u$.

Multiparametric Solution

Rather than using a numerical QP solver online to compute the optimizer $z(x)$ of (3) for each given current state vector x , the basic idea of explicit MPC is to pre-solve the QP off-line for the entire set of states x (or for a convex polyhedral subset $X \subseteq \mathbb{R}^m$ of interest) to get the optimizer function z , and therefore the MPC control law u , explicitly as a function of x .

The main tool to get such an explicit solution is *multiparametric quadratic programming* (mpQP). For mpQP problems of the form (3), Bemporad et al. (2002b) proved that the optimizer function $z^* : X_f \mapsto \mathbb{R}^n$ is piecewise affine and continuous over the set X_f of parameters x for which the problem is feasible (X_f is a polyhedral set, possibly $X_f = X$) and that the value function $V^* : X_f \mapsto \mathbb{R}$ associating with every $x \in X_f$ the corresponding optimal value of (3) is continuous, convex, and piecewise quadratic.

An immediate corollary is that the explicit version of the MPC control law u in (4), being the first n_u components of vector $z(x)$, is also a continuous and piecewise-affine state-feedback law defined over a partition of the set X_f of states into M polyhedral cells:

$$u(x) = \begin{cases} F_1 x + g_1 & \text{if } H_1 x \leq K_1 \\ \vdots & \vdots \\ F_M x + g_M & \text{if } H_M x \leq K_M \end{cases} \quad (5)$$

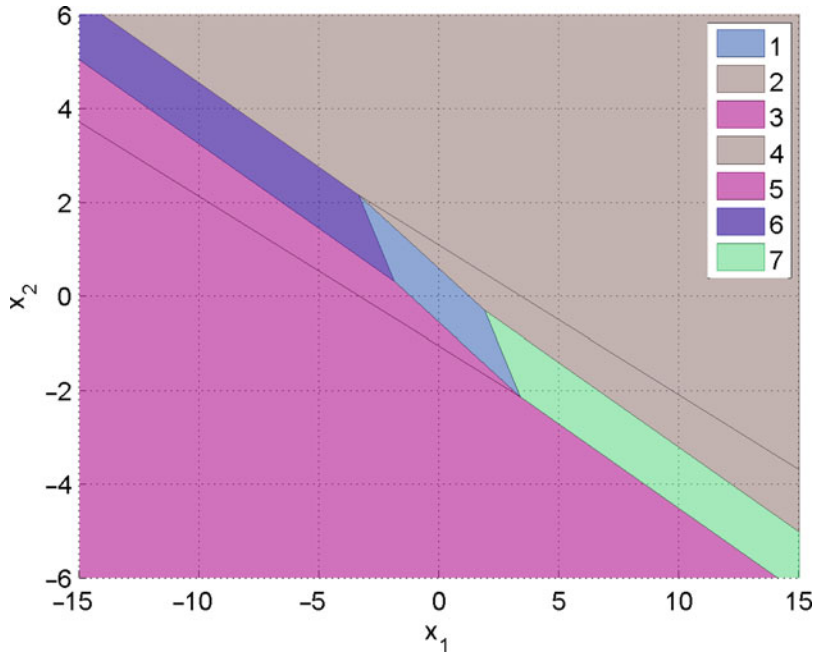
An example of such a partition is depicted in Fig. 1. The explicit representation (5) has mapped the MPC law (4) into a lookup table of linear gains, meaning that for each given x , the values computed by solving the QP (3) online and those obtained by evaluating (5) are exactly the same.

Multiparametric QP Algorithms

A few algorithms have been proposed in the literature to solve the mpQP problem (3). All of them construct the solution by exploiting the Karush-Kuhn-Tucker (KKT) conditions for optimality:

$$Hz + Fx + G'\lambda = 0 \quad (6a)$$

Explicit Model Predictive Control, Fig. 1 Explicit MPC solution for the double integrator example



E

$$\lambda_i(G^i z - W^i - S^i x) = 0, \forall i = 1, \dots, q \quad (6b)$$

$$Gz \leq W + Sx \quad (6c)$$

$$\lambda \geq 0 \quad (6d)$$

$$\hat{\lambda}(x) = 0 \quad (8b)$$

where $\lambda \in \mathbb{R}^q$ is the vector of Lagrange multipliers. For the strictly convex QP (3), conditions (6) are necessary and sufficient to characterize optimality.

An mpQP algorithm starts by fixing an arbitrary starting parameter vector $x_0 \in \mathbb{R}^m$ (e.g., the origin $x_0 = 0$), solving the QP (3) to get the optimal solution $z(x_0)$, and identifying the subset

$$\tilde{G}z(x) = \tilde{S}x + \tilde{W} \quad (7a)$$

of all constraints (6c) that are active at $z(x_0)$ and the remaining inactive constraints:

$$\hat{G}z(x) \leq \hat{S}x + \hat{W} \quad (7b)$$

Correspondingly, in view of the complementarity condition (6b), the vector of Lagrange multipliers is split into two subvectors:

$$\tilde{\lambda}(x) \geq 0 \quad (8a)$$

We assume for simplicity that the rows of $\tilde{G}$ are linearly independent. From (6a), we have the relation

$$z(x) = -H^{-1}(Fx + \tilde{G}'\tilde{\lambda}(x)) \quad (9)$$

that, when substituted into (7a), provides

$$\tilde{\lambda}(x) = -\tilde{M}(\tilde{W} + (\tilde{S} + \tilde{G}H^{-1}F)x) \quad (10)$$

where $\tilde{M} = \tilde{G}'(\tilde{G}H^{-1}\tilde{G}')^{-1}$ and, by substitution in (9),

$$z(x) = H^{-1}(\tilde{M}\tilde{W} + \tilde{M}(\tilde{S} + \tilde{G}H^{-1}F)x - Fx) \quad (11)$$

The solution $z(x)$ provided by (11) is the correct one for all vectors x such that the chosen combination of active constraints remains optimal. Such all vectors x are identified by imposing constraints (7b) and (8a) on $z(x)$ and $\tilde{\lambda}(x)$, respectively, that leads to constructing the polyhedral set (“critical region”):

$$CR_0 = \{x \in \mathbb{R}^n : \tilde{\lambda}(x) \geq 0, \hat{G}z(x) \leq \hat{W} + \hat{S}x\} \quad (12)$$

Different mpQP solvers were proposed to cover the rest $X \setminus CR_0$ of the parameter set with other critical regions corresponding to new combinations of active constraints. The most efficient methods exploit the so-called “facet-to-facet” property of the multiparametric solution (Spjøtvold et al. 2006) to identify neighboring regions as in Tøndel et al. (2003a), Baotić (2002), and Bemporad (2015). Alternative methods were proposed in Jones and Morari (2006), based on looking at (6) as a multiparametric linear complementarity problem, in Patrinos and Sarimveis (2010), which provides algorithms for determining all neighboring regions even in the case the facet-to-facet property does not hold, and in Gupta et al. (2011) based on the implicit enumeration of active constraint combinations.

All methods handle the case of *degeneracy*, which may happen for some combinations of active constraints that are linearly dependent, that is, the associated matrix $\tilde{G}$ has no full row rank (in this case, $\tilde{\lambda}(x)$ may not be uniquely defined).

Extensions

The explicit approach described earlier can be extended to the following MPC setting:

$$\min_{z, \epsilon} \sum_{k=0}^{N-1} \frac{1}{2} (y_k - \mathbf{r}_k)' Q_y (y_k - \mathbf{r}_k) + \frac{1}{2} \Delta u_k' R \Delta u_k + (u_k - \mathbf{u}_k^r)' R (u_k - \mathbf{u}_k^r)' + \rho \epsilon^2 \quad (13a)$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k + B_v \mathbf{v}_k \quad (13b)$$

$$y_k = Cx_k + Du_k + D_v \mathbf{v}_k \quad (13c)$$

$$u_k = u_{k-1} + \Delta u_k, \quad k = 0, \dots, N-1 \quad (13d)$$

$$\Delta u_k = 0, \quad k = N_u, \dots, N-1 \quad (13e)$$

$$\mathbf{u}_{\min}^k \leq u_k \leq \mathbf{u}_{\max}^k, \quad k = 0, \dots, N_u - 1 \quad (13f)$$

$$\Delta \mathbf{u}_{\min}^k \leq \Delta u_k \leq \Delta \mathbf{u}_{\max}^k, \quad k = 0, \dots, N_u - 1 \quad (13g)$$

$$\mathbf{y}_{\min}^k - \epsilon V_{\min} \leq y_k \leq \mathbf{y}_{\max}^k + \epsilon V_{\max} \quad (13h)$$

$$k = 0, \dots, N_c - 1$$

where R_Δ is a symmetric and positive definite matrix; matrices Q_y and R are symmetric and positive semidefinite; $\mathbf{v}_k$ is a vector of measured disturbances; y_k is the output vector; $\mathbf{r}_k$ its corresponding reference to be tracked; Δu_k is the vector of input increments; $\mathbf{u}_k^r$ is the input reference; $\mathbf{u}_{\min}^k, \mathbf{u}_{\max}^k, \Delta \mathbf{u}_{\min}^k, \Delta \mathbf{u}_{\max}^k, \mathbf{y}_{\min}^k, \mathbf{y}_{\max}^k$ are bounds; and N, N_u, N_c are, respectively, the prediction, control, and constraint horizons. The extra variable ϵ is introduced to soften output constraints, penalized by the (usually large) weight ρ_ϵ in the cost function (13a).

Everything marked in boldface in (13), together with the command input u_{-1} applied at the previous sampling step and the current state x , can be treated as a parameter with respect to which to solve the mpQP problem and obtain the explicit form of the MPC controller. For example, for a tracking problem with no anticipative action ($\mathbf{r}_k \equiv r_0, \forall k = 0, \dots, N-1$), no measured disturbance, and fixed upper and lower bounds, the explicit solution is a continuous piecewise-affine function of the parameter vector $\begin{bmatrix} x \\ r_0 \\ u_{-1} \end{bmatrix}$. Note that prediction models and/or weight matrices in (13) cannot be treated as parameters to maintain the mpQP formulation (3).

Linear MPC Based on Convex Piecewise-Affine Costs

A similar setting can be repeated for MPC problems based on linear prediction models and convex piecewise-affine costs, such as 1- and ∞ -norms. In this case, the MPC problem is mapped into a multiparametric linear programming (mpLP) problem, whose solution is again continuous and piecewise-affine with respect to the vector of parameters. For details, see Bemporad et al. (2002a).

Robust MPC

Explicit solutions to min-max MPC problems that provide robustness with respect to additive and/or multiplicative unknown-but-bounded uncertainty were proposed in Bemporad et al. (2003), based on a combination of mpLP and dynamic programming. Again the solution is piecewise affine with respect to the state vector.

Hybrid MPC

An MPC formulation based on convex piecewise-affine costs (like 1- or ∞ -norms) and linear hybrid dynamics expressed in mixed-logical dynamical (MLD) form can be solved explicitly by treating the optimization problem associated with MPC as a multiparametric mixed integer linear programming (mpMILP) problem. The solution is still piecewise affine but may be discontinuous, due to the presence of binary variables (Bemporad et al. 2000). A better approach based on dynamic programming combined with mpLP (or mpQP) was proposed in Borrelli et al. (2005) for hybrid systems in piecewise-affine (PWA) dynamical form and linear (or quadratic) costs. For explicit hybrid MPC based on quadratic costs, the issue of possible partition overlaps is solved in Fuchs et al. (2015).

Applicability of Explicit MPC

Complexity of the Solution

The complexity of the solution is given by the number M of regions that form the explicit solution (5), dictating the amount of memory to store the parametric solution (F_i , G_i , H_i , K_i , $i = 1, \dots, M$) and the worst-case execution time required to compute $F_i x + G_i$ once the problem of identifying the index i of the region $\{x : H_i x \leq K_i\}$ containing the current state x is solved (which usually takes most of the time). The latter is called the “point location problem,” and a few methods have been proposed to solve the problem more efficiently than searching linearly through the list of regions (see, e.g., the tree-based approach of Tøndel et al. 2003b).

An upper bound to M is 2^q , which is the number of all possible combinations of active constraints. In practice, M is much smaller than 2^q , as most combinations are never active at optimality for any of the vectors x (e.g., lower and upper limits on an actuation signal cannot be active at the same time, unless they coincide). Moreover, regions in which the first n_u component of the multiparametric solution $z(x)$ is the same can be joined together, provided that their union is a convex set (an optimal merging algorithm was proposed by Geyer et al. (2008) to get a minimal number M of partitions). Nonetheless, the complexity of the explicit MPC law typically grows exponentially with the number q of constraints. The number m of parameters is less critical and mainly affects the number of elements to be stored in memory (i.e., the number of columns of matrices F_i , H_i). The number n of free variables also affects the number M of regions, mainly because they are usually upper and lower bounded.

Computer-Aided Tools

The Model Predictive Control Toolbox (Bemporad et al. 2014) offers functions for designing explicit MPC controllers in MATLAB since 2014, implementing the algorithm described (Bemporad 2015). Other tools exist such as the Hybrid Toolbox (Bemporad 2003) and the Multi-Parametric Toolbox (Kvasnica et al. 2006).

Summary and Future Directions

Explicit MPC is a powerful tool to convert an MPC design into an equivalent control law that can be implemented as a lookup table of linear gains. Whether explicit MPC is preferable to online QP depends on available CPU time, data memory, program memory, and other practical considerations. In order to answer this question, Cimini and Bemporad (2017) analyzed the behavior of a state-of-the-art dual active set method for QP in a parametric way, finding out that the number of iterations to solve the QP online is a piecewise constant function of the

parameter vector. They precisely quantify how many flops the online QP solver takes in the worst case, so that the corresponding CPU time, as well as memory footprint, can be exactly compared with explicit MPC. Although suboptimal explicit MPC methods have been proposed for reducing the complexity of the control law, still the multiparametric approach remains convenient for relatively small problems (say one or two command inputs, short control and constraint horizons, up to ten states). For larger-size problems and/or time-varying prediction models, online QP solution methods tailored to embedded MPC are usually preferable.

Cross-References

- [Model Predictive Control in Practice](#)
- [Nominal Model-Predictive Control](#)
- [Optimization Algorithms for Model Predictive Control](#)

Recommended Reading

For getting started in explicit MPC, we recommend reading the paper by Bemporad et al. (2002b) and the survey paper by Alessio and Bemporad (2009). Hands-on experience using one of the MATLAB tools listed above is also useful for fully appreciating the potentials and limitations of explicit MPC. For understanding how to program a good multiparametric QP solver, the reader is recommended to take one of the approaches described in Tøndel et al. (2003a), Spjøtvold et al. (2006), Bemporad (2015), Patrinos and Sarimveis (2010), or Jones and Morari (2006).

Bibliography

- Alessio A, Bemporad A (2009) A survey on explicit model predictive control. In: Magni L, Raimondo DM, Allgower F (eds) *Nonlinear model predictive control: towards new challenging applications*. Lecture notes in control and information sciences, vol 384. Springer, Berlin/Heidelberg, pp 345–369
- Banjac G, Stellato B, Moehle N, Goulart P, Bemporad A, Boyd S (2017) Embedded code generation using the OSQP solver. *IEEE Conference on Decision and Control*, Melbourne, pp 1906–1911
- Baotić M (2002) An efficient algorithm for multiparametric quadratic programming. Technical Report AUT02-05, Automatic Control Institute, ETH, Zurich
- Bemporad A (2003) Hybrid toolbox – user’s guide. <http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox>
- Bemporad A (2015) A multiparametric quadratic programming algorithm with polyhedral computations based on nonnegative least squares. *IEEE Trans Autom Control* 60(11):2892–2903
- Bemporad A (2016) A quadratic programming algorithm based on nonnegative least squares with applications to embedded model predictive control. *IEEE Trans Autom Control* 61(4):1111–1116
- Bemporad A, Borrelli F, Morari M (2000) Piecewise linear optimal controllers for hybrid systems. In: *Proceedings of American Control Conference*, Chicago, pp 1190–1194
- Bemporad A, Borrelli F, Morari M (2002a) Model predictive control based on linear programming – the explicit solution. *IEEE Trans Autom Control* 47(12):1974–1985
- Bemporad A, Morari M, Dua V, Pistikopoulos E (2002b) The explicit linear quadratic regulator for constrained systems. *Automatica* 38(1):3–20
- Bemporad A, Borrelli F, Morari M (2003) Min-max control of constrained uncertain discrete-time linear systems. *IEEE Trans Autom Control* 48(9):1600–1606
- Bemporad A, Morari M, Ricker N (2014) Model predictive control toolbox for MATLAB – User’s guide. The Mathworks, Inc. <http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/>
- Borrelli F, Baotić M, Bemporad A, Morari M (2005) Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica* 41(10):1709–1721
- Borrelli F, Bemporad A, Morari M (2017) *Predictive control for linear and hybrid systems*. Cambridge University Press, Cambridge/New York
- Cimini G, Bemporad A (2017) Exact complexity certification of active-set methods for quadratic programming. *IEEE Trans Autom Control* 62(12):6094–6109
- Fuchs A, Axehill D, Morari M (2015) Lifted evaluation of mp-MIQP solutions. *IEEE Trans Autom Control* 60(12):3328–3331
- Geyer T, Torrisi F, Morari M (2008) Optimal complexity reduction of polyhedral piecewise affine systems. *Automatica* 44:1728–1740
- Gupta A, Bhartiya S, Nataraj PSV (2011) A novel approach to multiparametric quadratic programming. *Automatica* 47(9):2112–2117
- Jones C, Morari M (2006) Multiparametric linear complementarity problems. In: *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, pp 5687–5692
- Kvasnica M, Grieder P, Baotić M (2006) Multi parametric toolbox (MPT). <http://control.ee.ethz.ch/~mpt/>

- Mayne D, Rawlings J (2009) Model predictive control: theory and design. Nob Hill Publishing, LCC, Madison
- Patrinos P, Bemporad A (2014) An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Trans Autom Control* 59(1):18–33
- Patrinos P, Sarimveis H (2010) A new algorithm for solving convex parametric quadratic programs based on graphical derivatives of solution mappings. *Automatica* 46(9):1405–1418
- Ricker N (1985) Use of quadratic programming for constrained internal model control. *Ind Eng Chem Process Des Dev* 24(4):925–936
- Spjøtvold J, Kerrigan E, Jones C, Tøndel P, Johansen TA (2006) On the facet-to-facet property of solutions to convex parametric quadratic programs. *Automatica* 42(12):2209–2214
- Tøndel P, Johansen TA, Bemporad A (2003a) An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica* 39(3):489–497
- Tøndel P, Johansen TA, Bemporad A (2003b) Evaluation of piecewise affine control via binary search tree. *Automatica* 39(5):945–950
- Wang Y, Boyd S (2010) Fast model predictive control using online optimization. *IEEE Trans Control Syst Technol* 18(2):267–278

Extended Kalman Filters

Frederick E. Daum
Raytheon Company, Woburn, MA, USA

Abstract

The extended Kalman filter (EKF) is the most popular estimation algorithm in practical applications. It is based on a linear approximation to the Kalman filter theory. There are thousands of variations of the basic EKF design, which are intended to mitigate the effects of nonlinearities, non-Gaussian errors, ill-conditioning of the covariance matrix and uncertainty in the parameters of the problem.

Keywords

Estimation · Nonlinear filters

Synonyms

[EKF](#)

The extended Kalman filter (EKF) is by far the most popular nonlinear filter in practical engineering applications. It uses a linear approximation to the nonlinear dynamics and measurements and exploits the Kalman filter theory, which is optimal for linear and Gaussian problems; Gelb (1974) is the most accessible but thorough book on the EKF. The real-time computational complexity of the EKF is rather modest; for example, one can run an EKF with high-dimensional state vectors (d = several hundreds) in real time on a single microprocessor chip. The computational complexity of the EKF scales as the cube of the dimension of the state vector (d) being estimated. The EKF often gives good estimation accuracy for practical nonlinear problems, although the EKF accuracy can be very poor for difficult nonlinear non-Gaussian problems. There are many different variations of EKF algorithms, most of which are intended to improve estimation accuracy. In particular, the following types of EKFs are common in engineering practice: (1) second-order Taylor series expansion of the nonlinear functions, (2) iterated measurement updates that recompute the point at which the first order Taylor series is evaluated for a given measurement, (3) second-order iterated (i.e., combination of items 1 and 2), (4) special coordinate systems (e.g., Cartesian, polar or spherical, modified polar or spherical, principal axes of the covariance matrix ellipse, hybrid coordinates, quaternions rather than Euler angles, etc.), (5) preferred order of processing sequential scalar measurement updates, (6) decoupled or partially decoupled or quasi-decoupled covariance matrices, and many more variations. In fact, there is no such thing as “the” EKF, but rather there are thousands of different versions of the EKF. There are also many different versions of the Kalman filter itself, and all of these can be used to design EKFs as well. For example, there are many different equations to update the Kalman filter error covariance matrices with the intent of mitigating ill-conditioning and improving robustness, including (1) square-root factorization of the covariance matrix, (2) information matrix update, (3) square-root information update, (4) Joseph’s robust version

of the covariance matrix update, (5) at least three distinct algebraic versions of the covariance matrix update, as well as hybrids of the above.

Many of the good features of the Kalman filter are also enjoyed by the EKF, but unfortunately not all. For example, we have a very good theory of stability for the Kalman filter, but there is no theory that guarantees that an EKF will be stable in practical applications. The only method to check whether the EKF is stable is to run Monte Carlo simulations that cover the relevant regions in state space with the relevant measurement parameters (e.g., data rate and measurement accuracy). Secondly, the Kalman filter computes the theoretical error covariance matrix, but there is no guarantee that the error covariance matrix computed by the EKF approximates the actual filter errors, but rather the EKF covariance matrix could be optimistic by orders of magnitude in real applications. Third, the numerical values of the process noise covariance matrix can be computed theoretically for the Kalman filter, but there is no guarantee that these will work well for the EKF, but rather engineers typically tune the process noise covariance matrix using Monte Carlo simulations or else use a heuristic adaptive process (e.g., IMM). All of these shortcomings of the EKF compared with the Kalman filter theory are due to a myriad of practical issues, including (1) nonlinearities in the dynamics or measurements, (2) non-Gaussian measurement errors, (3) unmodeled measurement error sources (e.g., residual sensor bias), (4) unmodeled errors in the dynamics, (5) data association errors, (6) unresolved measurement data, (7) ill-conditioning of the covariance matrix, etc. The actual estimation accuracy of an EKF can only be gauged by Monte Carlo simulations over the relevant parameter space.

The actual performance of an EKF can depend crucially on the specific coordinate system that is used to represent the state vector. This is extremely well known in practical engineering applications (e.g., see Mehra 1971; Stallard 1991; Miller 1982; Markley 2007; Daum 1983; Schuster 1993). Intuitively, this is because the dynamics and measurement equations can be exactly linear in one coordinate system but not

another; this is very easy to see; start with dynamics and measurements that are exactly linear in Cartesian coordinates and transform to polar coordinates and we will get highly nonlinear equations. Likewise, we can have approximately linear dynamics and measurements in a specific coordinate system but highly nonlinear equations in another coordinate system. But in theory, the optimal estimation accuracy does not depend on the coordinate system. Moreover, in math and physics, coordinate-free methods are preferred, owing to their greater generality and simplicity and power. The physics does not depend on the specific coordinate system; this is essentially a definition of what “physics” means, and it has resulted in great progress in physics over the last few hundred years (e.g., general relativity, gauge invariance in quantum field theory, Lorentz invariance in special relativity, as well as a host of conservation laws in classical mechanics that are explained by Noether’s theorem which relates invariance to conserved quantities). Similarly in math, coordinate-free methods have been the royal road to progress over the last 100 years but not so for practical engineering of EKFs, because EKFs are approximations rather than being exact, and the accuracy of the EKF approximation depends crucially on the specific coordinate system used. Moreover, the effect of ill-conditioning of the covariance matrices in EKFs depends crucially on the specific coordinate system used in the computer; for example, if we could compute the EKF in principal coordinates, then the covariance matrices would be diagonal, and there would be no effect of ill-conditioning, despite enormous condition numbers of the covariance matrices. Surprisingly, these two simple points about coordinate systems are still not well understood by many researchers in nonlinear filtering.

Cross-References

- [Estimation, Survey on](#)
- [Kalman Filters](#)
- [Nonlinear Filters](#)
- [Particle Filters](#)

Bibliography

- Crisan D, Rozovskii B (eds) (2011) The Oxford handbook of nonlinear filtering. Oxford University Press, Oxford/New York
- Daum FE, Fitzgerald RJ (1983) Decoupled Kalman filters for phased array radar tracking. *IEEE Trans Autom Control* 28:269–283
- Gelb A et al (1974) Applied optimal estimation. MIT, Cambridge
- Markley FL, Crassidis JL, Cheng Y (2007) Nonlinear attitude filtering methods. *AIAA J* 30:12–28
- Mehra R (1971) A comparison of several nonlinear filters for reentry vehical tracking. *IEEE Trans Autom Control* 16:307–310
- Miller KS, Leskiw D (1982) Nonlinear observations with radar measurements. *IEEE Trans Aerosp Electron Syst* 2:192–200
- Ristic B, Arulampalam S, Gordon N (2004) Beyond the Kalman filter. Artech House, Boston
- Schuster MD (1993) A survey of attitude representations. *J Astronaut Sci* 41:439–517
- Sorenson H (ed) (1985) Kalman filtering: theory and application. IEEE, New York
- Stallard T (1991) Angle-only tracking filter in modified spherical coordinates. *AIAA J Guid* 14:694–696
- Tanizaki H (1996) Nonlinear filters, 2nd edn. Springer, Berlin/New York

Extremum Seeking Control

Miroslav Krstic
Department of Mechanical and Aerospace
Engineering, University of California,
San Diego, La Jolla, CA, USA

Abstract

Extremum seeking (ES) is a method for real-time non-model-based optimization. Though ES was invented in 1922, the “turn of the twenty-first century” has been its golden age, both in terms of the development of theory and in terms of its adoption in industry and in fields outside of control engineering. This entry overviews basic gradient- and Newton-based versions of extremum seeking with periodic and stochastic perturbation signals.

Keywords

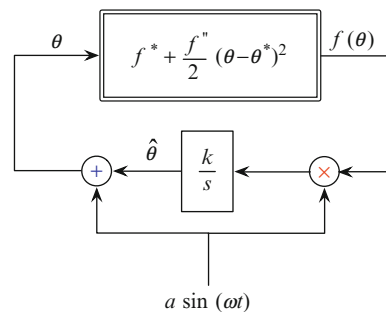
Extremum seeking · Averaging · Optimization · Stability

Introduction

Many versions of extremum seeking exist, with various approaches to their stability study (Krstic and Wang 2000; Tan et al. 2006; Liu and Krstic 2012). The most common version employs perturbation signals for the purpose of estimating the gradient of the unknown map that is being optimized. To understand the basic idea of extremum seeking, it is best to first consider the case of a static single-input map of the quadratic form, as shown in Fig. 1.

The Basic Idea of Extremum Seeking

Three different thetas appear in Fig. 1: θ^* is the unknown optimizer of the map, $\hat{\theta}(t)$ is the real-time estimate of θ^* , and $\theta(t)$ is the actual input into the map. The actual input $\theta(t)$ is based on the estimate $\hat{\theta}(t)$ but is perturbed by the signal $a \sin(\omega t)$ for the purpose of estimating the



Extremum Seeking Control, Fig. 1 The simplest perturbation-based extremum seeking scheme for a quadratic single-input map $f(\theta) = f^* + \frac{f''}{2} (\theta - \theta^*)^2$, where f^* , f'' , θ^* are all unknown. The user has to only know the sign of f'' , namely, whether the quadratic map has a maximum or a minimum, and has to choose the adaptation gain k such that $\text{sgn} k = -\text{sgn} f''$. The user has to also choose the frequency ω as relatively large compared to a , k , and f''

unknown gradient $f'' \cdot (\theta - \theta^*)$ of the map $f(\theta)$. The sinusoid is only one choice for a perturbation signal – many other perturbations, from square waves to stochastic noise, can be used in lieu of sinusoids, provided they are of zero mean. The estimate $\hat{\theta}(t)$ is generated with the integrator k/s with the adaptation gain k controlling the speed of estimation.

The ES algorithm is successful if the error between the estimate $\hat{\theta}(t)$ and the unknown θ^* , namely, the signal

$$\tilde{\theta}(t) = \hat{\theta}(t) - \theta^* \quad (1)$$

can be made to converge to any prescribed neighborhood of zero. Based on Fig. 1, the estimate is governed by the differential equation $\dot{\hat{\theta}} = k \sin(\omega t) f(\theta)$, which means that the estimation error is governed by

$$\frac{d\tilde{\theta}}{dt} = ka \sin(\omega t) \left[f^* + \frac{f''}{2} \left(\tilde{\theta} + a \sin(\omega t) \right)^2 \right] \quad (2)$$

Expanding the right-hand side one obtains

$$\begin{aligned} \frac{d\tilde{\theta}(t)}{dt} = & ka f^* \underbrace{\sin(\omega t)}_{\text{mean}=0} + ka^3 \frac{f''}{2} \underbrace{\sin^3(\omega t)}_{\text{mean}=0} \\ & + ka \frac{f''}{2} \underbrace{\sin(\omega t)}_{\text{fast, mean}=0} \underbrace{\tilde{\theta}(t)^2}_{\text{slow}} \\ & + ka^2 f'' \underbrace{\sin^2(\omega t)}_{\text{fast, mean}=1/2} \underbrace{\tilde{\theta}(t)}_{\text{slow}}. \end{aligned} \quad (3)$$

A theoretically rigorous time-averaging procedure allows to replace the above sinusoidal signals by their means, yielding the “average system”

$$\frac{d\tilde{\theta}_{\text{ave}}}{dt} = \frac{\overbrace{k f'' a^2}^{<0}}{2} \tilde{\theta}_{\text{ave}}, \quad (4)$$

which is exponentially stable. The averaging theory guarantees that there exists sufficiently large ω such that, if the initial estimate $\hat{\theta}(0)$ is

sufficiently close to the unknown θ^* ,

$$\begin{aligned} |\theta(t) - \theta^*| \leq & |\theta(0) - \theta^*| e^{\frac{k f'' a^2}{2} t} \\ & + O\left(\frac{1}{\omega}\right) + a, \quad \forall t \geq 0. \end{aligned} \quad (5)$$

For the user, the inequality (5) guarantees that, if a is chosen small and ω is chosen large, the input $\theta(t)$ exponentially converges to a small interval around the unknown θ^* and, consequently, the output $f(\theta(t))$ converges to the vicinity of the optimal output f^* .

ES for Multivariable Static Maps

For static maps, ES extends in a straightforward manner from the single-input case shown in Fig. 1 to the multi-input case shown in Fig. 2.

The algorithm measures the scalar signal $y(t) = Q(\theta(t))$, where $Q(\cdot)$ is an unknown map whose input is the vector $\theta = [\theta_1, \theta_2, \dots, \theta_n]^T$. The gradient is estimated with the help of the signals

$$S(t) = \begin{bmatrix} a_1 \sin(\omega_1 t) & \cdots & a_n \sin(\omega_n t) \end{bmatrix}^T \quad (6)$$

$$M(t) = \begin{bmatrix} \frac{2}{a_1} \sin(\omega_1 t) & \cdots & \frac{2}{a_n} \sin(\omega_n t) \end{bmatrix}^T \quad (7)$$

with nonzero perturbation amplitudes a_i and with a gain matrix K that is diagonal. To guarantee convergence, the user should choose $\omega_i \neq \omega_j$. This is a key condition that differentiates the multi-input case from the single-input case. In addition, for simplicity in the convergence analysis, the user should choose ω_i/ω_j as rational and $\omega_i + \omega_j \neq \omega_k$ for distinct i, j , and k .

If the unknown map is quadratic, namely, $Q(\theta) = Q^* + \frac{1}{2}(\theta - \theta^*)^T H(\theta - \theta^*)$, the averaged system is

$$\dot{\tilde{\theta}}_{\text{ave}} = K H \tilde{\theta}_{\text{ave}}, \quad H = \text{Hessian}. \quad (8)$$

If, for example, the map $Q(\cdot)$ has a maximum that is locally quadratic (which implies $H = H^T < 0$), and if the user chooses the elements of the diagonal gain matrix K as positive, the ES algorithm is guaranteed to be locally convergent. However, the convergence rate depends on

the unknown Hessian H . This weakness of the gradient-based ES algorithm is removed with the Newton-based ES algorithm.

A stochastic version of the algorithm in Fig. 2 also exists, in which $S(t)$ and $M(t)$ are replaced by

$$S(\eta(t)) = [a_1 \sin(\eta_1(t)), \dots, a_n \sin(\eta_n(t))]^T, \quad (9)$$

$$M(\eta(t)) = \left[\frac{2}{a_1(1 - e^{-q_1^2})} \sin(\eta_1(t)), \dots, \frac{2}{a_n(1 - e^{-q_n^2})} \sin(\eta_n(t)) \right]^T \quad (10)$$

where $\eta_i = \frac{q_i \sqrt{\varepsilon_i}}{\varepsilon_i s + 1} [\dot{W}_i]$ and $\dot{W}_i$ are independent unity-intensity white noise processes.

ES for Dynamic Systems

ES extends in a relatively straightforward manner from static maps to dynamic systems, provided the dynamics are stable and the algorithm's parameters are chosen so that the algorithm's dynamics are slower than those of the plant. The algorithm is shown in Fig. 3.

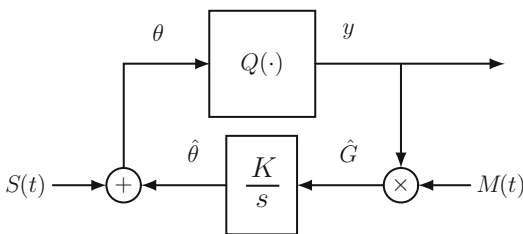
The technical conditions for convergence in the presence of dynamics are that the equilibria $x = l(\theta)$ of the system $\dot{x} = f(x, \alpha(x, \theta))$, where $\alpha(x, \theta)$ is the control law of an internal feedback loop, are locally exponentially stable uniformly in θ and that, given the output map $y = h(x)$, there exists at least one $\theta^* \in \mathbb{R}^n$

such that $\frac{\partial}{\partial \theta}(h \circ l)(\theta^*) = 0$ and $\frac{\partial^2}{\partial \theta^2}(h \circ l)(\theta^*) = H < 0$, $H = H^T$.

The stability analysis in the presence of dynamics employs both averaging and singular perturbations, in a specific order. The design guidelines for the selection of the algorithm's parameters follow the analysis, ensuring that the plant's dynamics are on a fast time scale, the perturbations are on a medium time scale, and the ES algorithm is on a slow time scale. Such a three-time-scale separation is achieved by the following selection process for the algorithm's parameters.

First, the user starts from some approximate knowledge of the time constants of the plant, even though the plant's model is largely uncertain. Second, the user selects the perturbation frequency whose period is relatively large compared to the time constants of the plant. That ensures that the perturbation is on a slower time scale than the plant, letting the plant dynamics “nearly settle” after any significant change in $\hat{\theta}(t)$. Finally, i.e., the parameters of the algorithm, including the gain K , perturbation amplitude a , and the filter poles ω_h and ω_l , are chosen by the user to be relatively small relative to the perturbation frequency ω . This ensures that the algorithm evolves on a slower time scale than the perturbation frequency. In summary, the overall system evolves on three time scales.

On the analysis side, the slowness of the perturbation allows the applicability of a singular perturbation argument in the analysis, treating the



Extremum Seeking Control, Fig. 2 Extremum seeking algorithm for a multivariable map $y = Q(\theta)$, where θ is the input vector $\theta = [\theta_1, \theta_2, \dots, \theta_n]^T$. The algorithm employs the additive perturbation vector signal $S(t)$ given in (6) and the multiplicative demodulation vector signal $M(t)$ given in (7)

plant dynamics as “instantaneous.” The slowness of the algorithm relative to its perturbation ω allows the applicability of an averaging argument in the analysis of the reduced system with the plant dynamics treated as instantaneous.

$$N_{ii}(t) = \frac{16}{a_i^2} \left(\sin^2(\omega_i t) - \frac{1}{2} \right),$$

$$N_{ij}(t) = \frac{4}{a_i a_j} \sin(\omega_i t) \sin(\omega_j t) \quad (11)$$

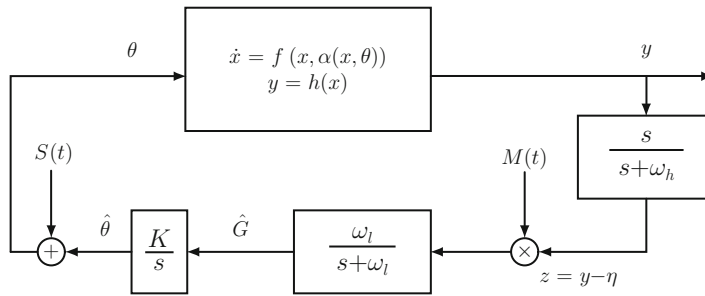
For a quadratic map, the averaged system in error variables $\tilde{\theta} = \hat{\theta} - \theta^*$, $\tilde{\Gamma} = \Gamma - H^{-1}$ is

Newton ES Algorithm for Static Map

A Newton version of the ES algorithm, shown in Fig. 4, ensures that the convergence rate be user-assignable, rather than being dependent on the unknown Hessian of the map.

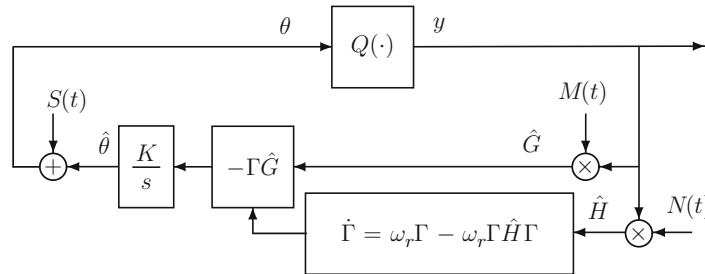
The elements of the demodulating matrix $N(t)$ for generating the estimate of the Hessian are given by

$$\begin{aligned} \frac{d\tilde{\theta}^{\text{ave}}}{dt} &= -K\tilde{\theta}^{\text{ave}} - \underbrace{K\tilde{\Gamma}^{\text{ave}}H\tilde{\theta}^{\text{ave}}}_{\text{quadratic}}, \\ \frac{d\tilde{\Gamma}^{\text{ave}}}{dt} &= -\omega_r\tilde{\Gamma}^{\text{ave}} - \omega_r\tilde{\Gamma}^{\text{ave}}\underbrace{H\tilde{\Gamma}^{\text{ave}}}_{\text{quadratic}}. \end{aligned} \quad (12)$$



Extremum Seeking Control, Fig. 3 The ES algorithm in the presence of dynamics with an equilibrium map $\theta \mapsto y$ that satisfies the same conditions as in the static case. If the dynamics are stable and the user employs parameters in the ES algorithm that make the algorithm dynamics

slower than the dynamics of the plant, convergence is guaranteed (at least locally). The two filters are useful in the implementation to reduce the adverse effect of the perturbation signals on asymptotic performance but are not needed in the stability analysis



Extremum Seeking Control, Fig. 4 A Newton-based ES algorithm for a static map. The multiplicative excitation $N(t)$ helps generate the estimate of Hessian $\frac{\partial^2 Q(\theta)}{\partial \theta^2}$ as $\hat{H}(t) = N(t)y(t)$. The Riccati matrix differential equation

$\dot{\Gamma}(t)$ generates an estimate of the Hessian's *inverse* matrix, avoiding matrix inversions of Hessian estimates that may be singular during the transient

Since the eigenvalues are determined by K and ω_r and are therefore independent of the unknown H , the (local) convergence rate is user-assignable.

Further Reading on Extremum Seeking

Since the publication of the first proof of stability of extremum seeking (Krstic and Wang 2000), thousands of papers have been published on this topic, presenting further theoretical developments and applications of ES. A proof that expands the validity of extremum seeking from local to global stability was published in Tan et al. (2006). The references Liu and Krstic (2010a, 2012) present stochastic versions of the algorithms in this entry, where the sinusoids are replaced by filtered white noise perturbation signals.

Among the natural applications of extremum seeking is the maximization of efficiency of renewable energy systems, notably, maximum power point tracking (MPPT) for photovoltaic (Ghaffari et al. 2014a) and wind (Ghaffari et al. 2014b) energy conversion.

“Source seeking” with autonomous vehicles has been one of the major applications of real-time optimization for dynamic systems. The first result in seeking the location of a signal source with a nonholonomic vehicle is Cochran and Krstic (2009), subsequently also achieved with stochastic perturbations and averaging (Liu and Krstic 2010b), whereas a more general problem of following any level set other than the trivial level set, at the extremum, is shown to be

solvable without perturbation-based extremum seeking but, instead, with an elegant use of PI feedback (Baronov and Baillieul 2011).

Cross-References

- [Adaptive Control: Overview](#)
- [Stochastic Approximation with Applications](#)

Bibliography

- Baronov D, Baillieul J (2011) A motion description language for robotic reconnaissance of unknown fields. *Eur J Control* 17:512–525
- Cochran J, Krstic M (2009) Nonholonomic source seeking with tuning of angular velocity. *IEEE Trans Autom Control* 54:717–731
- Ghaffari A, Krstic M, Seshagiri S (2014a) Power optimization for photovoltaic micro-converters using multivariable Newton-based extremum seeking. *IEEE Trans Control Syst Technol* 22:2141–2149
- Ghaffari A, Krstic M, Seshagiri S (2014b) Power optimization and control in wind energy conversion systems using extremum seeking. *IEEE Trans Control Syst Technol* 22:1684–1695
- Krstic M, Wang HH (2000) Stability of extremum seeking feedback for general dynamic systems. *Automatica* 36:595–601
- Liu S-J, Krstic M (2010a) Stochastic averaging in continuous time and its applications to extremum seeking. *IEEE Trans Autom Control* 55: 2235–2250
- Liu S-J, Krstic M (2010b) Stochastic source seeking for nonholonomic unicycle. *Automatica* 46: 1443–1453
- Liu S-J, Krstic M (2012) *Stochastic averaging and stochastic extremum seeking*. Springer, London/New York
- Tan Y, Nesic D, Mareels I (2006) On non-local stability properties of extremum seeking control. *Automatica* 42:889–903

Facility Control and Optimization Problems in Data Centers

Zhikui Wang
Cupertino, CA, USA

Abstract

Facility design and operation of data centers evolved through a few generations in the last decade, resulting in significant improvement of energy efficiency and operation reliability. The real-time control and optimization problems are challenging because of the heterogeneous architectures, nonlinearity of the systems under control, spatial and temporal variability of the dynamics, and tight coupling of the subsystems. This entry provides a brief introduction of the challenges through discussion about one server enclosure system and one typical data center cooling architecture.

Keywords

Data center facility · Thermal dynamics · Flow dynamics · Optimal control

Introduction

Data centers are special buildings. They run the critical Information and Communication Technology (ICT) infrastructure that hosts the Internet or enterprise applications and serves the operation of governments, schools, enterprises, business, and industries. A typical data center contains one or more computer or machine rooms, hosting the ICT equipment including computing, networking, and storage hardware, together with the power and cooling facility. It is a complicated system running 24×7 , for which reliability is of the first concern. High reliability is first considered through redundant designs in power, cooling, and ICT architectures. During the operation time, feedback control is applied to maintain the stability of the systems upon external disturbance, e.g., environmental changes, failures of equipment, or resource demand changes from the ICT applications. Energy efficiency is critical due to the cost and emerging environmental concerns. It has been improved significantly in the past decade with adoption of new architectures, systems, and products from chips, ICT systems to power/cooling facility (Barroso et al. 2018). Dynamic control and optimization also play important roles to reduce energy consumption while maintaining the operation performance.

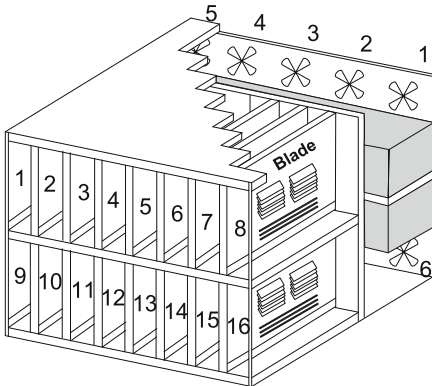
Modeling and Control of a Computer

A data center runs like a computer (Barroso et al. 2018). The main subsystems, including power, cooling, and computing, are tightly coupled. The interaction among these subsystems involves complicated thermal and fluid dynamics. Figure 1 illustrates one example of a computer, an enclosure with 16 blade servers in the front cooled by ten fans in the back. The fans pull air flows through the blades toward the back of the enclosure. Building numerical models are viable for such a small-scale system with appropriate approximation and simplification (Wang et al. 2009).

Based on energy balance, the change rate of the CPU temperature is approximately the sum of the rate at which heat is generated within the CPU device and the rate at which heat is transferred from the CPU to the ambient air flow, as represented below for the blade j .

$$C_1 \frac{dT_{CPU,j}}{dt} = \frac{C_2}{R_j} (T_{inlet,j} - T_{CPU,j}) + Q_j, \quad (1)$$

where C_1 and C_2 are constants related to the fluid properties of air, CPU package geometry, and material properties and T_{inlet} is the temperature of the cold air. Q is the heat transferred from the processor. It is not measurable but can be represented by the CPU power consumption, P_{CPU} ,



Facility Control and Optimization Problems in Data Centers, Fig. 1 A blade server enclosure (Wang et al. 2009)

which is usually modeled as a linear function of its utilization $Util$ with a bias of the idle power $P_{CPU, idle}$: $P_{CPU,j} = g_{CPU,j} * Util_j + P_{CPU,j, idle}$. R_j is the lumped thermal resistance between CPU and the cold air. Approximately,

$$R_j = \frac{C_3}{\dot{V}_j^{n_R}} + C_4, \quad \text{for blade } j, \quad (2)$$

where C_3 and C_4 are constants related to the fluid and material properties of the air, the CPU package, and the heatsink. The parameter n_R defines the shape of the thermal resistance curve as a function of the air flow rate $\dot{V}$.

On the flow dynamic side, the air flow through one blade may come from multiple fans, and the rate of the air flow generated by each fan $\dot{V}$ is about proportional to the fan speed FS . That is,

$$\dot{V}_j = \sum_i \eta_{ij} \times FS_i, \quad \text{for blade } j, \quad (3)$$

where η_{ij} is the correlation index between the speed of fan i and the air flow rate in blade j .

The models (1)–(3) could have been oversimplified but are still identifiable through system identification experiments (Wang et al. 2009). They capture the interaction among power, cooling, and computing workload in the enclosure, representing the non-uniform spatial and temporal cooling efficiency of the fans. The models imply a few ways to improve both efficiency and reliability. Below are a few examples.

- The total power consumption of the fans may be optimized because their capacity is shared by the blades. For instance, four fans running at 25% speed can provide the same volume of air flow as two fans running at 50% speed; however, the total fan power is reduced to 25% since the fan power is about a cubic function of the speed. An optimal fan controller is viable with the models identified (Wang et al. 2009).
- Moving workload around to take advantage of the spatial or temporal variance of cooling efficiency may improve the overall efficiency. It is usually formulated as a constrained

mixed-integer optimization problem (Tolia et al. 2009).

- Reliability can be improved because of sharing of the cooling capacity. Upon failures of one or more fans, a feedback controller can still drive the other fans to complement the capacity loss due to the failures.

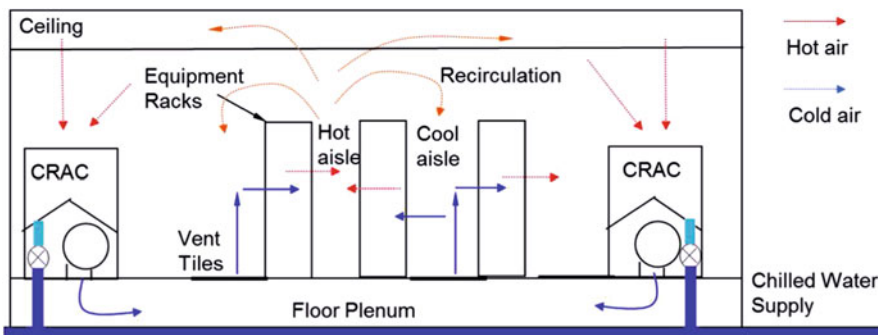
Data Center Level Modeling and Control

Figure 2 shows a traditional raised-floor cooling architecture inside a data center. The ICT equipment is housed in standard racks sitting atop a raised floor. The floor creates a plenum that is pressurized with cold air from CRAC (Computer Room Air Conditioning) units. Cold air is delivered to the ICT equipment via perforated or grated vent tiles in the floor. ICT equipment is typically designed to move air from the front of the rack to the back. Racks are arranged to form hot aisles and cold aisles to reduce the likelihood of equipment ingesting exhaust hot air from nearby racks. The exhaust air returns to the CRAC units either through the room or via a ceiling plenum. Within the CRAC units, heat is removed from the air stream via a refrigerant-cooled evaporator or a water-cooled heat exchanger connected to typically a hydronic network, which rejects heat to the external environment via cooling towers.

In the environment as in Fig. 2, one typical issue is the “short circuit” of air flows from the

hot aisle to the cold aisle, due to the uneven distribution of the pressure around the racks. This is usually caused by the lack of enough cold air to meet the demand of the racks. Re-circulation of the exhaust air causes “hot spots” in the front of the racks. From operation point of view, the temperatures of these “hot spots” are proxies of the cooling demand at the spots. They can be used as the metrics to drive cooling facility control. To measure the inlet temperatures of the racks, a distributed network of sensors may be built (Bash et al. 2006). A MIMO controller would be ideal to manipulate the CRAC units and maintain the inlet temperatures at or below the thresholds while minimizing the cooling cost, e.g., the total blower power of the CRAC units. However, it is almost impossible to derive the models similar as in (1) because of the large scale of data center cooling facility. Given that one CRAC unit may provide cooling air for certain zone, and the boundary of the zones varies slowly when the system runs around equilibrium, the cooling systems may be decomposed into multiple zones, and each is cooled approximately by one CRAC unit and controlled by a SISO controller. In Bash et al. (2006), the effective zone of each CRAC unit was identified. A dominant temperature was then discovered from each zone to drive the controller. PID controllers, together with a set of rules to discover the dominant sensors, were shown effective for tuning the blower speeds and supplied air temperatures of the CRAC units.

The open environment as in Fig. 2 was popular due to its low cost to build. However, the



Facility Control and Optimization Problems in Data Centers, Fig. 2 Raised floor data center with isolated and open aisles

re-circulation and “hot spots” make the dynamic thermal control difficult. Cooling capacity can still be over-provisioned since the operation of the CRAC unit for each zone is dominated by one local “hot spot.” To address these issues, new architecture designs were introduced in the last decade. In order to prevent re-circulation of hot air, the hot or cold aisles are contained by walls or other barriers. In the ideal case, what the CRAC units need to do is then to provide enough volume of cold air, e.g., to maintain constant pressure drops between cold aisles and hot aisles. Containment of hot aisles has been popular in practice. Other architecture changes include moving the cooling units closer to the racks, e.g., in-row cooling (Barroso et al. 2018) where the fan coils are installed on the top of the hot aisles or at the end of racks together with hot aisle containment. This new architecture can support higher power density and help reduce the mechanical work to move the cold air flows. From dynamic control point of view, local SISO temperature and pressure control may be good enough.

Run-Time Optimization of Cooling Efficiency

The standard organization ASHRAE created the guideline for the inlet air for ICT equipment (ASHRAE 2016), where ranges of recommended or allowable temperature and humidity are specified. Following the guideline will guarantee thermal safety for the ICT equipment, but not necessarily result in the highest efficiency. There are a few tradeoffs at data center level for efficiency optimization as represented below.

- Intuitively, increasing the inlet air temperature setpoints is desirable since it will reduce the power by the cooling facility. However, as implied in (1), raising the inlet air temperature will result in higher server fan power, higher volume rate of air flows, and then more power consumed by the blowers of the CRAC units. It is also well-known that the leakage power of integrated circuits increases with temperature,

which is not negligible at the data center scale (Walsh et al. 2010).

- As implied in (1)–(3), cooling efficiency at data center level is also a spatial and temporal variant especially in the open environment. Moving workload around can potentially improve the overall energy efficiency.
- Nonlinearity exists with almost all of the facility equipment, such as the efficiency and capacity curves of the equipment like blowers, compressors, and pumps. When the capacity of multiple units is shared, minimizing the aggregate energy consumption while meeting the cooling demand is an opportunity.

However, online optimization at data center scale is difficult due to lack of models. First-principal models may be developed for offline analysis for small-scale data centers. One such set of thermal and fluid dynamic models were developed in Breen et al. (2010) for facility systems from chips to cooling towers. This “white-box” approach delivers insights but is hard to be applied to large-scale data centers. In data centers that are measured extensively, large datasets are available for machine-learning-based approaches to build “black-box” models. As discussed in Gao (2014), neural network models were developed from the big data sets collected in Google data centers throughout the years. The models can then be used to guide configuration of facilities, e.g., temperature setpoints, and run what-if analysis.

More Control and Optimization Issues

Facility equipment such as the hydronic network of chillers, cooling towers, water economizers, and air economizers usually have their own control kits from the manufactures. These units, however, can still be manipulated through the knobs exposed by the kits, e.g., chilled water setpoints or the interface to turn on or off the units. Understanding and modeling the units themselves, the interaction between the units, and the interaction between the facility and the ICT equipment are

always important for development of the data center level control and optimization systems.

Power control is another important topic for data center facility management. There are multiple knobs for manipulating power consumption, e.g., DVFS (Dynamic Voltage/Frequency Scaling) of processors, turning on/off servers. Workload migration, consolidation, or reduction may affect the power demand as well. Power may be regulated to follow the change of the workload or capped for thermal or electrical safety purposes. The reference Wang and Ranganathan (2014) provides comprehensive discussions on related problems, architecture, model development, and control/optimization algorithms.

Future Directions for Research

Online efficiency optimization is still challenging. Machine-learning-based approaches are promising, not only for data center operation but also for the operation of the individual facility units such as heat exchangers or chiller sets.

Reliability is a critical concern for which dynamic control can play an important role. As one example, upon the failure of one or more facility units, e.g., CRAC units, a controller needs to respond promptly to deliver enough cooling capacity. Designs are usually intuitive. Formal investigation is needed for the effectiveness and the stability. In another example, many facility units are managed by individual kits. Formal study is desirable on the interaction between the units and the global stability of the overall system.

Data center is a typical cyber-physical system (CPS) where the physical facility, the ICT equipment, the hosted applications and services, and the life of people in the world are tightly connected. Opportunities exist to explore the data center control and optimization problems from CPS point of view (Parolini et al. 2012).

Cross-References

- [Adaptive Control: Overview](#)
- [Building Control Systems](#)

- [Building Energy Management System](#)
- [Model Predictive Control in Practice](#)

Recommended Reading

The reference Barroso et al. (2018) provides a comprehensive introduction for the state-of-art of data center designs and operation. Wang et al. (2009), Bash et al. (2006), and Breen et al. (2010) are useful references for modeling, control, and optimization problems, while Wang and Ranganathan (2014) can be taken as an introduction of data center power control.

Bibliography

- ASHRAE TC9.9 (2016) Data center power equipment thermal guidelines and best practices, whitepaper created by ASHRAE Technical Committee (TC) 9.9 Mission critical facilities, data centers, technology spaces, and electronic equipment, ASHRAE
- Barroso LA, Hölzle U, Ranganathan P (2018) The datacenter as a computer: an introduction to the design of warehouse-scale machines, 3rd edn. In: Martonosi M (ed) Synthesis lectures on computer architecture. Morgan & Claypool Publishers, San Rafael
- Bash CB, Patel CD, Sharma RK (2006) Dynamic thermal management of air cooled data centers. In: Thermal and thermomechanical proceedings 10th intersociety conference on phenomena in electronics systems, ITherm 2006, San Diego, pp 8–452
- Breen TJ, Walsh EJ, Punch J, Shah AJ, Bash CE (2010) From chip to cooling tower data center modeling: part I Influence of server inlet temperature and temperature rise across cabinet. In: Proceedings of 12th IEEE intersociety conference on thermal and thermomechanical phenomena in electronic systems, June 2010, Las Vegas
- Gao J (2014) Machine learning applications for data center optimization, Google White Paper, Mountain View, CA
- Parolini L, Sinopoli B, Krogh BH, Wang Z (2012) A cyber-physical systems approach to data center modeling and control for energy efficiency. *Proc IEEE* 100(1):254–268
- Tolia N, Wang Z, Ranganathan R, Bash CE, Marwah M, Zhu X (2009) Unified thermal and power management in server enclosures. In: ASME 2009 InterPACK conference international electronic packaging technical conference and exhibition, vol 2, pp 721–730. <https://doi.org/10.1115/InterPACK2009-89075>
- Walsh EJ, Breen TJ, Punch J, Shah AJ, Bash CE (2010) From chip to cooling tower data center modeling: part

II Influence of chip temperature control philosophy. In: Proceedings of 12th IEEE intersociety conference on thermal and thermomechanical phenomena in electronic systems, June 2010, Las Vegas

- Wang Z, Ranganathan P (2014) Chapter 6: cross-layer power management. In: Feng W-C (ed) The green computing book: tackling energy efficiency at large scale. CRC Press/Taylor & Francis Group, Boca Raton
- Wang Z, Bash CE, Tolia N, Marwah M, Zhu X, Ranganathan R (2009) Optimal fan speed control for thermal management of servers. In: ASME 2009 InterPACK conference international electronic packaging technical conference and exhibition, vol 2, pp 709–719. <https://doi.org/10.1115/InterPACK2009-89074>

Fault Detection and Diagnosis

Janos Gertler

George Mason University, Fairfax, VA, USA

Abstract

The fundamental concepts and methods of fault detection and diagnosis are reviewed. Faults are defined and classified as additive or multiplicative. The model-free approach of alarm systems is described and critiqued. Residual generation, using the mathematical model of the plant, is introduced. The propagation of additive and multiplicative faults to the residuals is discussed, followed by a review of the effect of disturbances, noise, and model errors. Enhanced residuals (structured and directional) are introduced. The main residual generation techniques are briefly described, including direct consistency relations, parity space, and diagnostic observers. Principal component analysis and its application to fault detection and diagnosis are outlined. The entry closes with some thoughts about future directions.

Keywords

Fault detection · Fault diagnosis · Residual generation · Consistency relations · Parity space · Diagnostic observers · Principal component analysis

Introduction

Faults are malfunctions of various elements of technical systems. Extreme cases of faults, called *failures*, are catastrophic breakdowns of the same. The technical systems (*the “plant”*) we are concerned with range from complex production systems (chemical plants, oil refineries, power stations) through major transportation equipment (airplanes, ships) to consumer machines (automobiles, home-heating systems, etc.). The faults may affect various parts of the main technical system (motors, pumps, storage tanks, pipelines) or devices interfacing the main technical system with computers providing for control, monitoring, and operator information. These latter include *sensors* (measuring devices) and *actuators* (devices acting on the process, such as valves).

The objective of *fault detection* is to determine and signal if there is a fault anywhere in the system. *Fault diagnosis* is aimed at providing more specific information about the fault; *fault isolation* is to pinpoint the component(s) (sensors, actuators, or plant components) where the fault is located, while *fault identification* is to determine (estimate) the size of the fault and, in some cases, the time of its arrival. With the ubiquitous presence of the computer, fault detection and diagnosis (FDD) is, in general, a function of the computer interfaced to the plant.

The simplest approaches to FDD consist of comparing individual plant measurements to preset limits, without utilizing any knowledge of the plant model (*limit checking* or “*alarm systems*”). More sophisticated techniques rely on an explicit mathematical model of the plant. They compare plant measurements to estimates obtained, from other measurements, by the model; any discrepancy may be an indication of faults. Another class of techniques (generally but incorrectly called “*data driven*”), most notably principal component analysis (PCA), include the estimation of an implicit model, from empirical plant data, and then use this in ways similar to the model-based methods. These approaches will be described in more detail in the sequel.

Alarm Systems

Alarm systems rely on the comparison of individual plant measurements to their respective limits. The limits may be two- or one-sided (upper and lower limit or upper limit only) and may have one or two levels (preliminary and full alarm). Momentary comparisons may be extended to include trend checks. Alarm systems are relatively simple but suffer from two major shortcomings:

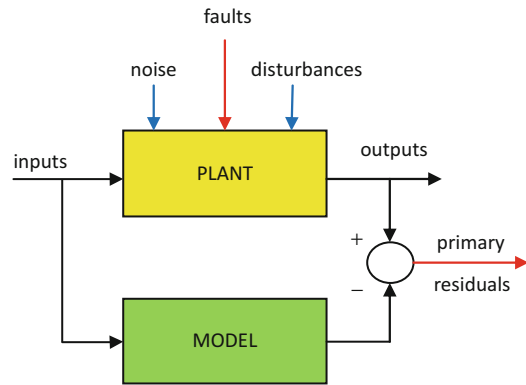
- They have very limited fault specificity. A variable exceeding its limit is not a fault but a symptom of faults. A single component fault may cause alarm on many variables and a particular alarm may be due to various component faults.
- They have limited fault sensitivity. What is “normal” for a plant output variable depends on the value of the plant inputs. Such relationship, however, cannot be considered without a plant model; therefore, the alarm thresholds need to be set conservatively high.

Because of their simplicity, and despite the above shortcomings, alarm systems are widely used in industrial applications.

Model-Based FDD Concepts

Model-based methods utilize an explicit mathematical model of the plant. Such model is obtained usually from empirical plant data by systems identification methods or, exceptionally, from the “first principles” understanding of the plant. Model building, though critical to the success of model-based FDD, is usually not considered part of the FDD effort. The models may be linear or nonlinear, static or dynamic, and continuous- or discrete-time. In FDD, most frequently linear discrete-time dynamic models are used.

The fundamental idea of model-based FDD is the comparison of measured plant outputs to



Fault Detection and Diagnosis, Fig. 1 The concept of residual generation

their estimates, obtained, via the mathematical model, from measured or actuated plant inputs (Fig. 1).

Any discrepancy is (at least ideally) an indication that a fault (or faults) is (are) present in the system. Mathematically, the difference between the measured output $y_i(t)$ and its estimate $\hat{y}_i(t)$ is a (primary) residual (Willsky 1976):

$$e_i(t) = y_i(t) - \hat{y}_i(t)$$

In general, residuals are quantities that are zero in the absence of faults and nonzero in their presence.

Unfortunately, it is not only the faults that can make the residuals nonzero. Usually the plant is subject to disturbances (unmeasured deterministic inputs) and noise (unmeasured random inputs) (Fig. 1). In addition, and most importantly, model-based FDD is subject to model errors (either due to initial inaccuracies in model building or to changes in the physical plant) (Isermann 1984). The FDD algorithm should be designed, as much as possible, to be insensitive to noise and “robust” in face of disturbances and model errors.

Additive and multiplicative faults Depending on the way they appear in the system equations, faults may be additive or multiplicative. Additive faults are sensor and actuator biases,

leaks in the plant, etc. Multiplicative faults are changes in the plant parameters. In the following input-output relationship, $\mathbf{u}(t)$ is the vector of observed (measured or commanded) plant inputs, $\mathbf{y}(t)$ is the vector of measured plant outputs, and $\mathbf{p}(t)$ is the vector of additive faults; t is the discrete time. $\mathbf{M}(q)$ and $\mathbf{S}(q)$ are transfer function matrices in the shift operator q , and θ is the vector of plant parameters. Then,

$$\mathbf{y}(t) = \mathbf{M}(q, \theta)\mathbf{u}(t) + \mathbf{S}(q, \theta)\mathbf{p}(t)$$

The (“primary”) residual vector $\mathbf{e}(t)$, in response to additive faults, is

$$\mathbf{e}(t) = \mathbf{y}(t) - \mathbf{M}(q, \theta)\mathbf{u}(t) = \mathbf{S}(q, \theta)\mathbf{p}(t)$$

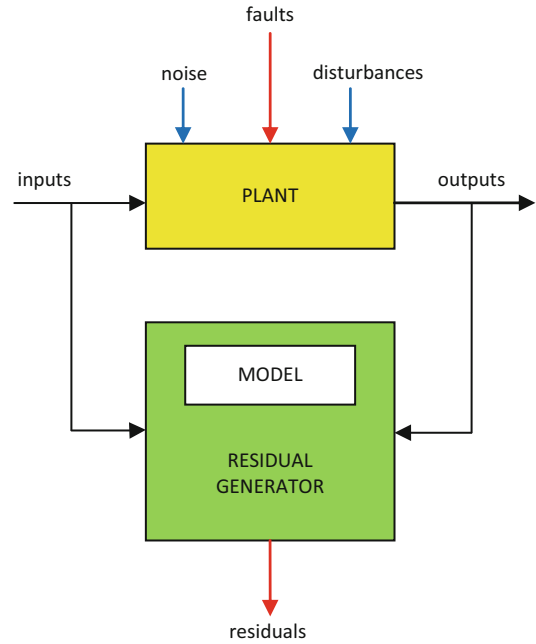
If there are multiplicative faults, then $\theta = \theta^\circ + \Delta\theta$, where θ° is the nominal parameter vector and $\Delta\theta$ is its change (the parametric fault); now the residual vector $\mathbf{e}(t)$ is (Gertler 1998)

$$\begin{aligned}\mathbf{e}(t) &= \mathbf{y}(t) - \mathbf{M}(q, \theta^\circ)\mathbf{u}(t) \\ &= \sum_j (\partial \mathbf{M}(q, \theta) / \partial \theta_j) \mathbf{u}(t) \Delta \theta_j\end{aligned}$$

Enhanced residuals To facilitate the isolation of faults, the primary residuals $\mathbf{e}(t)$ are subject to some enhancement manipulation. The three widely used enhancement techniques are:

- *Structured residuals*, whereas each residual is selectively sensitive to a subset of faults, resulting in a fault-specific set of zero/nonzero residuals upon a particular fault (fault codes);
- *Directional residuals*, whereas the residual vector maintains a fault-specific direction in response to each particular fault;
- *Diagonal residuals*, whereas each residual responds only to a particular fault.

Residual generators take the input and output observations from the plant and generate enhanced residuals by one of the above schemes, utilizing the mathematical model of the plant (Fig. 2).



Fault Detection and Diagnosis, Fig. 2 Residual generator

Dealing with noise Noise is practically unavoidable in physical systems. In FDD, basically two steps may be taken to reduce the effect of noise:

- *Residual filtering*. This can be achieved by basing decisions on moving averages of the residuals, or by applying explicit low-pass filters to the residuals, or by designing the residual generators in such a way that they have built-in low-pass behavior.
- *Statistical testing of the residuals*. Structured residuals are tested individually, as scalars, against thresholds; each residual is then represented by a Boolean 1 or 0, depending on the outcome of the test. Directional residuals are tested as vectors against multivariable distributions. The test thresholds are determined either theoretically, using assumptions for the source noise, or empirically based on measurements from fault-free operating conditions.

Dealing with disturbances Additive disturbances are unmeasurable inputs. If the disturbance-to-output transfer function (or

equivalent state-space representation) is known, then it is possible to design residuals that are completely decoupled from (insensitive to) those disturbances. However, the FDD algorithm is subject to a certain degree of “design freedom,” defined by the number of outputs in the physical system; disturbance decoupling is competing for this freedom with fault isolation enhancement. If there are too many disturbances, or if their path to the outputs is unknown, then only approximate decoupling is possible, making FDD also approximate, usually designed to optimize some (H-infinity) performance index.

Dealing with model errors Model errors are also unavoidable in most practical situations. This is the most serious obstacle in the application of model-based FDD techniques. In some very special cases, uncertainty of a particular plant parameter may be handled as a “multiplicative disturbance,” and residuals designed to be explicitly decoupled from it. In general, however, only approximate solutions are possible, reducing the residuals’ sensitivity to modeling errors, at the expense of also reducing their sensitivity to faults. Design methods utilizing some optimization techniques, mostly based on H-infinity or similar performance indices, are available in the literature (Edelmayer et al. 1994).

Residual Generation Methods For linear dynamic systems, provided exact (non-approximate) solution is possible, there are three major techniques to design residual generators: (i) direct consistency (parity) relations, (ii) parity space, and (iii) diagnostic observers. We will briefly introduce the three methods, for discrete-time plant models and additive faults. Note that though they look formally different, if designed for the same plant under the same design conditions, the three methods yield identical residuals (Gertler 1991).

Direct consistency (parity) relations (Gertler 1998). The input-output model of the plant is utilized directly in the design. The enhanced residuals are obtained from the primary residuals by a transformation $\mathbf{W}(q)$:

$$\begin{aligned}\mathbf{r}(t) &= \mathbf{W}(q)\mathbf{e}(t) = \mathbf{W}(q)[\mathbf{y}(t) - \mathbf{M}(q)\mathbf{u}(t)] \\ &= \mathbf{W}(q)\mathbf{S}(q)\mathbf{p}(t)\end{aligned}$$

The desired behavior of the residuals is specified as $\mathbf{r}(t) = \mathbf{Z}(q)\mathbf{p}(t)$, where the specification $\mathbf{Z}(q)$ contains the basic residual properties (structure or directions) plus the residual dynamics. The resulting design condition is $\mathbf{W}(q)\mathbf{S}(q) = \mathbf{Z}(q)$. If the $\mathbf{S}(q)$ matrix is square, which is usually the case (Gertler 1998), then this can be solved for $\mathbf{W}(q)$ by direct inversion. The residual generator has to be causal and stable; this can always be achieved by the appropriate modification of the dynamics in $\mathbf{Z}(q)$.

Parity space (Chow and Willsky 1984). This method, also known as the “Chow-Willsky scheme,” relies on the state-space description of the system:

$$\begin{aligned}\mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{E}\mathbf{p}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{F}\mathbf{p}(t)\end{aligned}$$

Stacking n consecutive output vectors $\mathbf{y}(t)$ (where n is the order of the model), and chain-substituting the state $\mathbf{x}(t)$, yields the equation

$$\mathbf{Y}(t) = \mathbf{J}\mathbf{x}(t-n) + \mathbf{K}\mathbf{U}(t) + \mathbf{L}\mathbf{P}(t)$$

where $\mathbf{Y}(t)$, $\mathbf{U}(t)$ and $\mathbf{P}(t)$ are stacked vectors and $\mathbf{J}$, $\mathbf{K}$ and $\mathbf{L}$ are hyper-matrices composed of the $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{E}$, $\mathbf{F}$ matrices. Now

$$\mathbf{E}^*(t) = \mathbf{Y}(t) - \mathbf{K}\mathbf{U}(t) = \mathbf{L}\mathbf{P}(t) + \mathbf{J}\mathbf{x}(t-n)$$

would be a stacked vector of primary residuals, was it not for the presence of the inaccessible initial state $\mathbf{x}(t-n)$. To obtain true residuals, a transformation $r_i(t) = \mathbf{w}_i^T \mathbf{E}^*(t)$ is necessary, so that $\mathbf{w}_i^T \mathbf{J} = \mathbf{0}$. Any vector $\mathbf{w}_i$ satisfying this orthogonality condition is a *parity vector*, together spanning the *parity space*. Any parity vector yields a true residual $r_i(t)$; they can be so chosen that a set of residuals possesses structured behavior.

Diagnostic observers Various observer schemes have been extensively investigated as possible residual generator algorithms. The basic full-order *Luenberger observer* (assuming $\mathbf{D}=\mathbf{0}$) is

$$\hat{\mathbf{x}}(t+1) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{K}\mathbf{e}(t)$$

where $\mathbf{K}$ is the observer gain matrix and

$$\mathbf{e}(t) = \mathbf{y}(t) - \mathbf{C}\hat{\mathbf{x}}(t)$$

is the innovation vector. If the observer is stable then, apart from the start-up transient of the observer, the innovation qualifies as the primary residual. The gain matrix $\mathbf{K}$ is the major design parameter; it is chosen to place the observer poles, thus achieving stability and desired dynamic behavior (e.g., transient response and noise suppression). The remaining design freedom can be utilized to influence residual properties. The latter can be further affected by the transformation $\mathbf{r}(t) = \mathbf{H}\mathbf{e}(t)$, where the $\mathbf{H}$ matrix is an additional design parameter. Diagnostic observers can be designed for both structured and directional residuals (Chen and Patton 1999; White and Speyer 1987). Other observer schemes, most notably the *unknown input observer*, have also been proposed (Frank and Wunnenberg 1989). Because of their complexity, the detailed design procedures of diagnostic observers go beyond the scope of this entry.

Principal Component Analysis

Principal component analysis is extensively used in the monitoring of complex plants with hundreds of variables because, by revealing linear relations among the variables, it significantly reduces the dimensionality of the plant model (Kresta et al. 1991). When PCA is applied for FDD, it implies two phases. In the training phase, an implicit plant model is created from empirical plant data. In the monitoring phase, this model is used for FDD.

Training data (measured inputs and outputs) are collected from the plant during fault-free operation. The covariance matrix of the data is formed and its eigenstructure obtained. Due to

linear relations among the data, some of the eigenvalues will be zero (or near-zero, in the presence of noise). The eigenvectors belonging to the nonzero eigenvalues form the *data space*, where the fault-free data exist, while those belonging to the zero eigenvalues form the *residual space*.

It is the residual space that is utilized for FDD. The projection of a measurement vector onto the residual space is the (primary) residual. A statistical test on its size leads to a detection decision (absence or presence of faults). A threshold test is necessary because noise also causes nonzero residuals. An analysis of the eigenvectors spanning the residual space shows how the various faults propagate to the primary residual. This allows for the design of residual manipulations yielding structured or directional residuals, just like in the FDD methods based on exact models (Gertler et al. 1999).

The procedure as described above applies to sensor and actuator faults; inclusion of plant faults requires extra effort (and experiments). Also, PCA is primarily meant for static models. Its extension to discrete-time dynamic models is straightforward, but it increases the size of the model, proportionally to the dynamic order of the model.

Summary and Future Directions

The theoretical and methodological foundations of fault detection and diagnosis were laid in the 1980s and early 1990s. Today FDD is a mature field within systems and control engineering. There is a very significant level of activity, as measured in published papers and conference contributions.

There are still open problems in various areas, such as extensions to nonlinear or parameter varying systems (Senname et al. 2013). Also, the proliferation of networked, distributed, and cooperative systems is raising a whole range of exciting challenges for FDD, calling for decentralized, distributed, or cooperative diagnosis techniques (Blanke et al. 2016; Fang et al. 2007).

Several “classical” application areas of FDD have gained new significance with the dramatic development of the target system: on-board

automotive FDD with the arrival of self-driving cars or aerospace FDD with the proliferation of unmanned aerial vehicles (Isermann 2010). Electric cars bring into new focus the diagnosis of classical electrical systems. The spread of wind turbines and turbine farms has opened a whole new field for FDD applications, both at the individual device level and as a network problem (Simani and Castaldi 2019). With tighter water resources management, FDD in water distribution networks is gaining significance (Puig et al. 2017). These application problems are visible in conference programs and technical journals. What is more important, with the ever-growing availability of computing power and a welcome evolution of industrial attitude, FDD technology now increasingly finds its way to a broad spectrum of real-life applications.

Cross-References

- [Fault Detection and Diagnosis: Computational Issues and Tools](#)
- [Robust Fault Diagnosis and Control](#)

Bibliography

- Blanke M, Kinnaert M, Lunze J, Staroswiecki M (2016) *Diagnosis and fault tolerant control*, 3rd edn. Springer, Berlin/Heidelberg
- Chen J, Patton RJ (1999) *Robust model-based fault diagnosis for dynamic systems*. Kluwer, Boston/Dordrecht/Amsterdam
- Chow EJ, Willsky AS (1984) Analytical redundancy and the design of robust failure detection systems. *IEEE Trans Autom Control* AC-29:603–614
- Edelmayer A, Bokor J, Keviczky L (1994) An H-infinity filtering approach to robust detection of failures in dynamic systems. In: 33rd IEEE conference on decision and control, Lake Buena Vista
- Fang H, Ye H, Zhang M (2007) Fault diagnosis of networked control systems. *Annu Rev Control* 31:55–68
- Frank PM, Wunnenberg J (1989) Robust fault diagnosis using unknown input observer schemes. In: Patton R, Frank P, Clark R (eds) *Fault diagnosis in dynamic systems*. Prentice Hall, Upper Saddle River
- Gertler J (1991) A survey of analytical redundancy methods in fault detection and isolation. Plenary paper, IFAC safeprocess symposium, Baden-Baden
- Gertler J (1998) *Fault detection and diagnosis in engineering systems*. Marcel Dekker, New York
- Gertler J, Li W, Huang Y, McAvoy T (1999) Isolation enhanced principal component analysis. *AICHE J* 45:323–334
- Isermann R (1984) Process fault detection based on modeling and estimation methods. *Automatica* 20:387–404
- Isermann R (2010) Diagnosis methods for electronic controlled vehicles. *Int J Veh Dyn Mobil* 36:77–117
- Kresta JV, MacGregor JF, Marlin TE (1991) Multivariate statistical monitoring of processes. *Can J Chem Eng* 69:35
- Puig V, Ocampo-Martinez C, Perez R, Cembrano G, Quevedo J, Escobet T (eds) (2017) *Real-time monitoring and operational control of drinking water systems*. Springer, Cham
- Senname O, Gaspar P, Bokor J (eds) (2013) *Robust control and parameter varying approaches*. Springer, Berlin/Heidelberg
- Simani S, Castaldi P (2019) Fault diagnosis techniques for a wind turbine system. *IntechOpen*. <https://doi.org/10.5772/intechopen.83810>
- White JE, Speyer JL (1987) Detection filter design: spectral theory and algorithm. *IEEE Trans Autom Control* AC-32:593–603
- Willsky AS (1976) A survey of design methods for failure detection in dynamic systems. *Automatica* 12:601–611

Fault Detection and Diagnosis: Computational Issues and Tools

Andreas Varga
Gilching, Germany

Abstract

Two basic fault diagnosis problems are formulated for linear time-invariant systems with additive faults, and general existence conditions of their solutions are given. Several computational paradigms are discussed, which are instrumental for the development of efficient and numerically reliable synthesis procedures for the design of fault detection filters. An overview of the available computational software tools is presented.

Keywords

CACSD · Fault detection and diagnosis · Fault detection and isolation · Numerical methods · Software tools

Introduction

The theoretical developments for the model-based fault detection and diagnosis for linear time-invariant systems are essentially completed. Also, the development of numerically reliable computational procedures for the synthesis of fault detection filters has been mostly finalized in the last decade. The new generation of synthesis procedures is well suited as the basis of implementation of versatile software tools, which allow the solution of synthesis problems in the most general setting using the best available numerical methods. This entry presents an overview of these recent developments both in the computational synthesis procedures and associated software tools. Therefore, this presentation complements two other entries of this encyclopedia: ► [Fault Detection and Diagnosis](#) and ► [Robust Fault Diagnosis and Control](#).

Two basic synthesis problems of fault detection filters are formulated and general solvability conditions are given, which guarantee the existence of the solutions in the most general setting, for both continuous- and discrete-time systems. For each of the formulated problems, general synthesis procedures are presented. The development of these procedures relies on several computational paradigms, which are discussed in details.

Plant Models with Additive Faults

The plant models underlying the discussed synthesis methods (also called *synthesis models*) are linear time-invariant (LTI) system models, where the faults are equated with special (unknown) disturbance inputs. An important class of models with additive faults arises when defining the fault signals for two main categories of faults, namely, actuator and sensor faults. Two basic forms of synthesis models are used.

The input-output plant model with additive faults has the form

$$\mathbf{y}(\lambda) = G_u(\lambda)\mathbf{u}(\lambda) + G_d(\lambda)\mathbf{d}(\lambda) + G_f(\lambda)\mathbf{f}(\lambda) + G_w(\lambda)\mathbf{w}(\lambda), \quad (1)$$

where $\mathbf{y}(\lambda)$, $\mathbf{u}(\lambda)$, $\mathbf{d}(\lambda)$, $\mathbf{f}(\lambda)$, and $\mathbf{w}(\lambda)$, with boldface notation, are the Laplace-transformed

(in the continuous-time case) or Z-transformed (in the discrete-time case) p -dimensional system output vector $y(t)$; m_u -dimensional control input vector $u(t)$; m_d -dimensional disturbance vector $d(t)$; m_f -dimensional fault vector $f(t)$; and m_w -dimensional noise vector $w(t)$, respectively, and where $G_u(\lambda)$, $G_d(\lambda)$, $G_f(\lambda)$, and $G_w(\lambda)$ are proper *transfer function matrices* (TFMs) from the respective inputs to outputs. Input-output models with additive faults of the form (1) are useful in formulating various fault diagnosis problems, in deriving general solvability conditions and in describing conceptual synthesis procedures. However, these models are generally not suited for numerical computations, due to the potentially high sensitivity of polynomial-based model representations.

For computational purposes, instead of the input-output model (1) with the compound TFM $[G_u(\lambda) \ G_d(\lambda) \ G_f(\lambda) \ G_w(\lambda)]$, an equivalent state-space model is used having the form

$$\begin{aligned} E\lambda x(t) &= Ax(t) + B_u u(t) + B_d d(t) \\ &\quad + B_f f(t) + B_w w(t), \\ y(t) &= Cx(t) + D_u u(t) + D_d d(t) \\ &\quad + D_f f(t) + D_w w(t), \end{aligned} \quad (2)$$

with the n -dimensional state vector $x(t)$, where $\lambda x(t) := \dot{x}(t)$ or $\lambda x(t) := x(t+1)$ depending on the type of the system, continuous- or discrete-time, respectively. The matrix E is generally invertible and is frequently taken as $E = I_n$. Plant models of the form (2) often arise from the linearization of nonlinear dynamic plant models in specific operation points and for fixed values of plant parameters. The noise inputs generally account for the effects of uncertainties (e.g., inherent variabilities in operating points and parameters).

Notation: To indicate the input-output equivalence of the models in (1) and (2), we use the notation

$$\begin{aligned} &[G_u(\lambda) \ G_d(\lambda) \ G_f(\lambda) \ G_w(\lambda)] \\ &= \left[\begin{array}{c|c} A - \lambda E & B_u \ B_d \ B_f \ B_w \\ \hline C & D_u \ D_d \ D_f \ D_w \end{array} \right]. \end{aligned}$$

Residual Generation

A nonzero fault signal $f \neq 0$ in (1) signifies a deviation from the normal behavior of the plant due to an unexpected event (e.g., physical component failure or supply breakdown). Generally, the occurrence of a fault must be detected as early as possible to prevent further degradation of the plant behavior. *Fault detection and diagnosis* (FDD) is concerned with one or more of the following aspects: the detection of the occurrence of any fault (*fault detection*), the localization of detected faults (*fault isolation*), the reconstruction of the fault signal (*fault estimation*), and the classification of the detected faults and determination of their characteristics (*fault identification*). In this entry we constrain the FDD problem only to the *fault detection* and *fault detection and isolation* (FDI) aspects.

A FDD system is a device (usually based on a collection of real-time processing algorithms) suitably set up to fulfill the above tasks. The minimal functionality of any FDD system is illustrated in Fig. 1.

The main component of any FDD system is the *residual generator* (or *fault detection filter*), which produces residual signals grouped in a q -dimensional vector r by processing the available measurements y and the known values of control inputs u . The role of the residual signals is to indicate the presence or absence of faults, and therefore the residual r must be equal (or close) to zero in the absence of faults and significantly different from zero after a fault occurs.

For decision-making when solving fault detection problems, a suitable measure of the residual magnitude is generated in a scalar evaluation signal θ (e.g., $\theta = \|r\|$), which is then used to set the corresponding decision variable ι as follows: $\iota = 1$, if $\theta > \tau$ for a detected fault, and $\iota = 0$ if $\theta \leq \tau$, for the lack of faults, where τ is a given detection threshold.

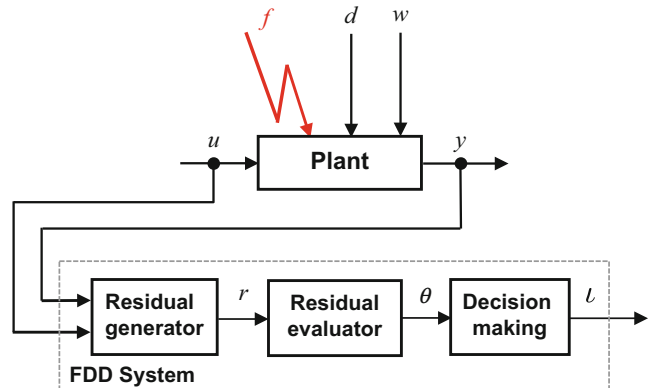
For decision-making when solving FDI problems, $r(t)$ is generally a structured vector with, say n_b components $r^{(i)}(t)$, $i = 1, \dots, n_b$, and θ and ι are n_b -dimensional vectors, with θ_i representing a measure of the magnitude of the i -th residual component (e.g., $\theta_i = -\|r^{(i)}\|$). The i -th component of the binary signature vector ι is set $\iota_i = 1$ or $\iota_i = 0$ corresponding to a fired (i.e., $\theta_i > \tau$) or not fired (i.e., $\theta_i \leq \tau$) component $r^{(i)}(t)$, respectively.

A linear residual generator employed in the FDD system in Fig. 1 has the input-output form

$$\mathbf{r}(\lambda) = Q(\lambda) \begin{bmatrix} \mathbf{y}(\lambda) \\ \mathbf{u}(\lambda) \end{bmatrix}, \quad (3)$$

where $Q(\lambda)$ is the TFM of the filter. For a physically realizable filter, $Q(\lambda)$ must be *stable* (i.e., with all its poles having negative real parts for a continuous-time system or magnitudes less than one for a discrete-time system). The *order* of $Q(\lambda)$ is the dimension of the state vector of a minimal state-space realization of $Q(\lambda)$. The dimension q of the residual vector $r(t)$ depends on the fault diagnosis problem to be solved. The input-output representation (3) is called the

Fault Detection and Diagnosis: Computational Issues and Tools, Fig. 1 Basic fault diagnosis setup



implementation form of the fault detection filter and is the basis of its real-time implementation.

The residual signal $r(t)$ in (3) generally depends via the system outputs $y(t)$ of all system inputs $u(t)$, $d(t)$, $f(t)$, and $w(t)$. The *internal form* of the filter is obtained by replacing in (3) $y(\lambda)$ by its expression in (1), and is given by

$$\begin{aligned} \mathbf{r}(\lambda) = & R_u(\lambda)\mathbf{u}(\lambda) + R_d(\lambda)\mathbf{d}(\lambda) \\ & + R_f(\lambda)\mathbf{f}(\lambda) + R_w(\lambda)\mathbf{w}(\lambda), \end{aligned} \quad (4)$$

where

$$\begin{aligned} & [R_u(\lambda) \mid R_d(\lambda) \mid R_f(\lambda) \mid R_w(\lambda)] \\ & := Q(\lambda) \begin{bmatrix} G_u(\lambda) & G_d(\lambda) & G_f(\lambda) & G_w(\lambda) \\ I_{m_u} & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \quad (5)$$

For a successfully designed filter $Q(\lambda)$, all TFMs in the corresponding internal form (4) are stable and fulfill specific fault diagnosis requirements.

The basic functionality of a well-designed fault detection filter is to ensure the lack of false alarms, in the case when no faults occurred, and the lack of missed detection of faults, in the case of occurrence of a fault. The first requirement is fulfilled if, in the presence of noise, the signal norm $\|r\|$ is sufficiently small for all possible control, disturbance, and noise inputs. The requirement on the lack of missed detections is fulfilled provided $\|r\|$ is sufficiently large for any fault of sufficiently large magnitude for all possible control, disturbance, and noise inputs.

Fault Diagnosis Problems

Two basic fault diagnosis problems are formulated in what follows. To fulfill the requirement for the lack of false alarms, for both problems we require that by a suitable choice of a stable fault detection filter $Q(\lambda)$, we achieve that the residual signal $r(t)$ is fully decoupled from the control input $u(t)$ and disturbance input $d(t)$. Thus, the following *decoupling conditions* must be generally fulfilled:

$$\begin{aligned} (i) \quad & R_u(\lambda) = 0, \\ (ii) \quad & R_d(\lambda) = 0. \end{aligned} \quad (6)$$

Since the effect of a nonzero noise input $w(t)$ can usually not be fully decoupled from the residual $r(t)$, an additional requirement is that the influence of the noise signal $w(t)$ is negligible. Thus, the following *noise attenuation condition* has to be fulfilled:

$$(iv) \quad R_w(\lambda) \approx 0, \text{ with } R_w(\lambda) \text{ stable.} \quad (7)$$

The condition $R_w(\lambda) \approx 0$ expresses the requirement that the transfer gain $\|R_w(\lambda)\|$ (measured by any suitable norm) can be made arbitrarily small.

For particular fault diagnosis problems specific requirements on $R_f(\lambda)$ have to be additionally fulfilled. In what follows, we formulate two basic fault diagnosis problems, for which we also give necessary and sufficient conditions for the existence of their solutions (for proofs see Varga 2017).

Fault Detection Problem: FDP

The basic additional requirement is simply to achieve, by a suitable choice of a fault detection filter $Q(\lambda)$, that in the absence of noise input (i.e., $w(t) = 0$), the residual $r(t)$ is influenced by all fault components $f_j(t)$, $j = 1, \dots, m_f$. Let $R_{f_j}(\lambda)$ denote the j -th column of $R_f(\lambda)$. This requirement can be expressed as the following *detection condition* to be fulfilled for all faults:

$$\begin{aligned} (iv) \quad & R_{f_j}(\lambda) \neq 0, \\ & j = 1, \dots, m_f \text{ with } R_f(\lambda) \text{ stable.} \end{aligned} \quad (8)$$

The solvability conditions of the FDP are simple rank conditions involving only the TFMs from the disturbances and faults:

Theorem 1 *For the system (1), the FDP is solvable if and only if*

$$\begin{aligned} & \text{rank}[G_d(\lambda) \ G_{f_j}(\lambda)] > \text{rank } G_d(\lambda), \\ & j = 1, \dots, m_f, \end{aligned} \quad (9)$$

where $G_{f_j}(\lambda)$ denotes the j -th column of $G_f(\lambda)$. Here, $\text{rank } G(\lambda)$ denotes the normal rank of the rational TFM $G(\lambda)$, representing the maximal rank of the complex matrix $G(\lambda)$ over all values of $\lambda \in \mathbb{C}$ such that $G(\lambda)$ has finite norm.

For fault diagnosis applications, the importance of solving FDPs primarily lies in the fact that the solution of the (more involved) fault isolation problems can be addressed by successively solving several appropriately formulated FDPs.

Fault Detection and Isolation Problem: FDIP

For the isolation of faults, we employ residual generator filters formed by stacking a bank of n_b filters of the form

$$\mathbf{r}^{(i)}(\lambda) = Q^{(i)}(\lambda) \begin{bmatrix} \mathbf{y}(\lambda) \\ \mathbf{u}(\lambda) \end{bmatrix}, \quad (10)$$

where the i -th filter $Q^{(i)}(\lambda)$ generates the corresponding i -th residual component $r^{(i)}(t)$ (scalar or vector). This leads to the following structured residual vector $r(t)$ and block-structured filter $Q(\lambda)$

$$r(t) = \begin{bmatrix} r^{(1)}(t) \\ \vdots \\ r^{(n_b)}(t) \end{bmatrix}, \quad Q(\lambda) = \begin{bmatrix} Q^{(1)}(\lambda) \\ \vdots \\ Q^{(n_b)}(\lambda) \end{bmatrix}. \quad (11)$$

The resulting $R_f(\lambda)$ is an $n_b \times m_f$ block-structured TFM of the form

$$R_f(\lambda) = \begin{bmatrix} R_{f_1}^{(1)}(\lambda) & \cdots & R_{f_{m_f}}^{(1)}(\lambda) \\ \vdots & \ddots & \vdots \\ R_{f_1}^{(n_b)}(\lambda) & \cdots & R_{f_{m_f}}^{(n_b)}(\lambda) \end{bmatrix}, \quad (12)$$

where the (i, j) -th block of $R_f(\lambda)$ is defined as

$$R_{f_j}^{(i)}(\lambda) := Q^{(i)}(\lambda) \begin{bmatrix} G_{f_j}(\lambda) \\ 0 \end{bmatrix}$$

and describes how the j -th fault f_j influences the i -th residual component $r^{(i)}(t)$.

We associate to the block-structured $R_f(\lambda)$ in (12) the $n_b \times m_f$ binary structure matrix S_{R_f} ,

whose (i, j) -th element is defined as

$$\begin{aligned} S_{R_f}(i, j) &= 1 \quad \text{if } R_{f_j}^{(i)}(\lambda) \neq 0, \\ S_{R_f}(i, j) &= 0 \quad \text{if } R_{f_j}^{(i)}(\lambda) = 0. \end{aligned} \quad (13)$$

If $S_{R_f}(i, j) = 1$ then we say that the residual component $r^{(i)}$ is sensitive to the j -th fault f_j , while if $S_{R_f}(i, j) = 0$ then the j -th fault f_j is decoupled from $r^{(i)}$. The m_f columns of S_{R_f} are called *fault signatures*. Since each nonzero column of S_{R_f} is associated with the corresponding fault input, fault isolation can be performed by comparing the resulting binary decision vector ι in Fig. 1 (i.e., the signatures of fired or not fired residual components) with the fault signatures coded in the columns of S_{R_f} .

For a given $n_b \times m_f$ structure matrix S , the FDIP requires the determination of a stable filter $Q(\lambda)$ of the form (11) such that for the corresponding block-structured $R_f(\lambda)$ in (12), the following condition is additionally fulfilled:

$$(iv) \quad S_{R_f} = S, \quad \text{with } R_f(\lambda) \text{ stable.} \quad (14)$$

The solution of the FDIP can be addressed by solving n_b suitably formulated FDPs. The i -th FDP arises by reformulating the i -th FDIP for determining the i -th filter $Q^{(i)}(\lambda)$ in (10) for a structure matrix which is the i -th row of S . This can be accomplished by redefining the fault components corresponding to zero entries in the i -th row of S as additional disturbance inputs to be decoupled in the i -th residual component $r^{(i)}(t)$. Let $\widehat{G}_d^{(i)}(\lambda)$ be the corresponding TFM formed from the columns of $G_f(\lambda)$ for which $S_{ij} = 0$. We have the following solvability conditions for the FDIP, which simply express the solvability of the n_b FDPs formulated for the n_b rows of S :

Theorem 2 For the system (1) and a given $n_b \times m_f$ structure matrix S , the FDIP is solvable if and only if for $i = 1, \dots, n_b$

$$\begin{aligned} &\text{rank} [G_d(\lambda) \quad \widehat{G}_d^{(i)}(\lambda) \quad G_{f_j}(\lambda)] \\ &> \text{rank} [G_d(\lambda) \quad \widehat{G}_d^{(i)}(\lambda)] \end{aligned} \quad (15)$$

for all j such that $S_{ij} \neq 0$.

Synthesis of Fault Detection Filters

Several computational paradigms are instrumental in developing generally applicable, numerically reliable, and computationally efficient synthesis methods of residual generators. In what follows we shortly review these paradigms and discuss their roles in the synthesis procedures.

Nullspace-Based Synthesis

An important synthesis paradigm is the use of the nullspace method as a first synthesis step to ensure the fulfillment of the decoupling conditions $R_u(\lambda) = 0$ and $R_d(\lambda) = 0$ in (6). This can be done by choosing $Q(\lambda)$ of the form

$$Q(\lambda) = \overline{Q}_1(\lambda) Q_1(\lambda), \quad (16)$$

where the factor $Q_1(\lambda)$ is a left nullspace basis of the rational matrix

$$G(\lambda) := \begin{bmatrix} G_u(\lambda) & G_d(\lambda) \\ I_{m_u} & 0 \end{bmatrix}. \quad (17)$$

For any $Q(\lambda)$ of the form (16), we have

$$[R_u(\lambda) \ R_d(\lambda)] = Q(\lambda)G(\lambda) = 0.$$

With (16), the fault detection filter in (3) can be rewritten in the alternative form

$$\mathbf{r}(\lambda) = \overline{Q}_1(\lambda) Q_1(\lambda) \begin{bmatrix} \mathbf{y}(\lambda) \\ \mathbf{u}(\lambda) \end{bmatrix} = \overline{Q}_1(\lambda) \overline{\mathbf{y}}(\lambda), \quad (18)$$

where

$$\begin{aligned} \overline{\mathbf{y}}(\lambda) &:= Q_1(\lambda) \begin{bmatrix} \mathbf{y}(\lambda) \\ \mathbf{u}(\lambda) \end{bmatrix} \\ &= \overline{G}_f(\lambda) \mathbf{f}(\lambda) + \overline{G}_w(\lambda) \mathbf{w}(\lambda), \end{aligned} \quad (19)$$

with

$$[\overline{G}_f(\lambda) \ \overline{G}_w(\lambda)] := Q_1(\lambda) \begin{bmatrix} G_f(\lambda) & G_w(\lambda) \\ 0 & 0 \end{bmatrix}. \quad (20)$$

With this first preprocessing step, we reduced the original problems formulated for the system (1) to simpler ones, which can be formulated for the

reduced system (19) (without control and disturbance inputs), for which we have to determine the TFM $\overline{Q}_1(\lambda)$ of the simpler fault detection filter (18).

At this stage we can assume that both $Q_1(\lambda)$ and the TFMs of the reduced system (19) are proper and even stable. This can be always achieved by replacing $Q_1(\lambda)$, with a stable basis $M(\lambda)Q_1(\lambda)$, where $M(\lambda)$ is an invertible, stable TFM, such that $M(\lambda)[Q_1(\lambda) \ \overline{G}_f(\lambda) \ \overline{G}_w(\lambda)]$ is stable. Such an $M(\lambda)$ can be determined as the (minimum-degree) denominator of a stable left coprime factorization of $[Q_1(\lambda) \ \overline{G}_f(\lambda) \ \overline{G}_w(\lambda)]$.

Both synthesis problems can be equivalently reformulated to determine a filter $\overline{Q}_1(\lambda)$ for the reduced system (19), for which simpler solvability conditions hold. For the solvability of the FDP, we have the following corollary to Theorem 1.

Corollary 1 *For the system (1), the FDP is solvable if and only if*

$$\overline{G}_{f_j}(\lambda) \neq 0, \quad j = 1, \dots, m_f. \quad (21)$$

Similar simpler conditions hold for the solvability of the FDIP.

Filter Updating Techniques

In many synthesis procedures, the TFM of the resulting filter $Q(\lambda)$ can be expressed in a factored form as

$$Q(\lambda) = Q_K(\lambda) \cdots Q_2(\lambda) Q_1(\lambda), \quad (22)$$

where $Q_1(\lambda)$ is a left nullspace basis of $G(\lambda)$ in (17) satisfying $Q_1(\lambda)G(\lambda) = 0$, and $Q_1(\lambda)$, $Q_2(\lambda)Q_1(\lambda)$, ..., can be interpreted as partial syntheses addressing specific requirements. Since each partial synthesis may represent a valid fault detection filter, this approach can be flexibly used for employing or combining different synthesis techniques.

The determination of $Q(\lambda)$ in the factored form (22) can be formulated as a K -step synthesis procedure based on successive updating of an initial filter $Q = Q_1(\lambda)$ and the nonzero terms of its corresponding internal form

$$\begin{aligned} R(\lambda) &:= [R_f(\lambda) \ R_w(\lambda)] \\ &= Q_1(\lambda) \begin{bmatrix} G_f(\lambda) & G_w(\lambda) \\ 0 & 0 \end{bmatrix} \end{aligned}$$

as follows: for $k = 2, \dots, K$, do

Step k: Determine $Q_k(\lambda)$ using the current $Q(\lambda)$ and $R(\lambda)$ and then perform the updating as

$$Q(\lambda) \leftarrow Q_k(\lambda)Q(\lambda), \quad R(\lambda) \leftarrow Q_k(\lambda)R(\lambda).$$

The updating operations are efficiently performed using state-space description-based formulas. The state-space realizations of the TFMs $Q(\lambda)$ and $R(\lambda)$ in the implementation and internal forms can be jointly expressed in the generalized state-space (or descriptor system) representation

$$[Q(\lambda) \ R(\lambda)] = \left[\begin{array}{c|cc} A_Q - \lambda E_Q & B_Q & B_R \\ \hline C_Q & D_Q & D_R \end{array} \right]. \quad (23)$$

The state-space realizations of $Q(\lambda)$ and $R(\lambda)$ share the matrices A_Q , E_Q , and C_Q . The preservation of these forms at each updating step is the basis of the filter updating-based synthesis paradigm. To allow operations with non-proper TFMs (i.e., with singular E_Q), the use of the descriptor system representation is instrumental for developing numerically reliable computational algorithms.

The main benefit of using explicit state-space based updating formulas is the possibility to ensure at each step the cancellation of a maximum number of poles and zeros between the factors. This allows to keep the final order of the filter $Q(\lambda)$ as low as possible. In this way, the updating-based synthesis approach naturally leads to so-called *integrated computational algorithms*, with strongly coupled successive computational steps. Since the form of the realizations of $Q(\lambda)$ and $R(\lambda)$ in (23) are preserved, the stability is simultaneously achieved for both $Q(\lambda)$ and $R(\lambda)$ at the final synthesis step. At the last step, the standard state-space realization of $Q(\lambda)$ is usually recovered, to facilitate its real-time implementation.

Least Order Synthesis

The least order synthesis of fault detection filters means to determine filters $Q(\lambda)$ with the least possible orders, to help in reducing the computational burden associated with their real-time implementations. The main tool to achieve least order synthesis is the solution of suitable *minimal cover problems*. If $X_1(\lambda)$ and $X_2(\lambda)$ are rational matrices of the same column dimension, then the *left minimal cover problem* is to find $X(\lambda)$ and $Y(\lambda)$ such that

$$X(\lambda) = X_1(\lambda) + Y(\lambda)X_2(\lambda), \quad (24)$$

and the order of $[X(\lambda) \ Y(\lambda)]$ is minimal.

A typical second step in many synthesis procedures is to choose $Q_2(\lambda)$ such that the product $Q_2(\lambda)Q(\lambda)$ has least dynamical order and, simultaneously, a certain *admissibility* condition is fulfilled (usually involving the nonzero TFMs $R_f(\lambda)$ and $R_w(\lambda)$). For the solution of the FDP, the rows of $Q(\lambda) := Q_1(\lambda)$ form a left nullspace basis and the employed admissibility conditions are

$$Q_2(\lambda)R_{f_j}(\lambda) \neq 0, \quad j = 1, \dots, m_f, \quad (25)$$

which guarantee the detectability of all fault components, and additionally to (25), the full row rank requirement on $Q_2(\lambda)R_w(\lambda)$. The latter condition is only imposed for convenience, to simplify the subsequent computational steps.

The determination of candidate solutions $Q_2(\lambda)$ such that $Q_2(\lambda)Q(\lambda)$ has least order can be done by solving left minimal cover problems of the form (24), where $X_1(\lambda)$ and $X_2(\lambda)$ represent disjoint subsets of basis vectors, such that: $Q(\lambda) = \begin{bmatrix} X_1(\lambda) \\ X_2(\lambda) \end{bmatrix}$, $Q_2(\lambda) = [I \ Y(\lambda)]$, and $X(\lambda) = Q_2(\lambda)Q(\lambda)$ together with $Y(\lambda)$ represent the solution of the left cover problem. A systematic search over increasing orders of candidate solutions can be performed, and the search stops when the admissibility conditions are fulfilled.

Coprime Factorization Techniques

A desired dynamics of the resulting final filters $Q(\lambda)$ and $R(\lambda)$ can be enforced by choosing

a suitable invertible factor $M(\lambda)$, such that $M(\lambda)[Q(\lambda) \ R(\lambda)]$ has desired poles. This can be achieved by computing a left coprime factorization

$$[Q(\lambda) \ R(\lambda)] = M^{-1}(\lambda)[N_Q(\lambda) \ N_R(\lambda)]$$

with $M(\lambda)$ and $[N_Q(\lambda) \ N_R(\lambda)]$ coprime and having arbitrary stable poles, and then performing the updating operation

$$[Q(\lambda) \ R(\lambda)] \leftarrow [N_Q(\lambda) \ N_R(\lambda)].$$

To illustrate the state-space formulas based updating technique, we employ a realization of the denominator factor $M(\lambda)$ of the form

$$M(\lambda) = \left[\begin{array}{c|c} \frac{A_Q + KC_Q - \lambda E_Q}{C_Q} & \frac{K}{I} \end{array} \right],$$

where K (the so-called output injection gain matrix) is determined by eigenvalue assignment techniques to ensure stability (i.e., to assign the poles of $M(\lambda)$ to lie in a suitable stable region). Then, the numerator factors are obtained in forms compatible with $Q(\lambda)$ and $R(\lambda)$ in (23) as

$$N_Q(\lambda) = \left[\begin{array}{c|c} \frac{A_Q + KC_Q - \lambda E_Q}{C_Q} & \frac{B_Q + KD_Q}{D_Q} \end{array} \right],$$

$$N_R(\lambda) = \left[\begin{array}{c|c} \frac{A_Q + KC_Q - \lambda E_Q}{C_Q} & \frac{B_R + KD_R}{D_R} \end{array} \right].$$

Outer-Co-inner Factorization

The outer-co-inner factorization plays an important role in the solution of the FDP and FDIP. We only consider the particular case of a stable and full row rank rational matrix $G(\lambda)$, without zeros on the boundary of the stability region, which can be factored as

$$G(\lambda) = G_o(\lambda)G_i(\lambda), \quad (26)$$

where $G_o(\lambda)$ is the (invertible) minimum-phase outer factor (i.e., stable and having only stable zeros), and $G_i(\lambda)$ is co-inner (i.e., stable and $G_i(\lambda)G_i^{\sim}(\lambda) = I$; recall that $G_i^{\sim}(s) = G_i^T(-s)$

for a continuous-time system and $G_i^{\sim}(z) = G_i^T(1/z)$ for a discrete-time system).

In solving the formulated synthesis problems, $G(\lambda) = R_w(\lambda)$ and the optimal choice of the corresponding factor, say $Q_4(\lambda)$, is $Q_4(\lambda) = G_o^{-1}(\lambda)$, which is stable, because $G_o(\lambda)$ is minimum-phase.

Remark: In the case when $G(\lambda)$ has zeros on the boundary of the stability domain, a factorization as in (26) can still be computed, with $G_o(\lambda)$ containing, besides the stable zeros, also all zeros of $G(\lambda)$ on the boundary of the stability domain. In this case, the inverse $G_o^{-1}(\lambda)$ is unstable and/or improper. Such a case is called *non-standard* and, when it is encountered, requires a special treatment. We should emphasize that non-standard cases occur only in conjunction with employing a particular solution method and are generally not related to the solvability of the fault diagnosis problems.

Synthesis Procedures

To illustrate the application of the computational paradigms discussed in the previous section, we present in this section the detailed synthesis procedure to solve the FDP.

Procedure AFD

Inputs: $\{G_u(\lambda), G_d(\lambda), G_f(\lambda), G_w(\lambda)\}$

Outputs: $Q(\lambda), R_f(\lambda), R_w(\lambda)$

1. Compute a proper minimal left nullspace basis $Q_1(\lambda)$ of $G(\lambda)$ in (17); set $Q(\lambda) = Q_1(\lambda)$ and compute

$$R(\lambda) := [R_f(\lambda) \ R_w(\lambda)]$$

$$= Q_1(\lambda) \begin{bmatrix} G_f(\lambda) & G_w(\lambda) \\ 0 & 0 \end{bmatrix}.$$

2. **Exit** if $\exists j$ such that $R_{f_j}(\lambda) = 0$ (no solution). Choose a proper $Q_2(\lambda)$ such that $Q_2(\lambda)Q(\lambda)$ has least order, satisfies $Q_2(\lambda)R_{f_j}(\lambda) \neq 0 \ \forall j$, and $Q_2(\lambda)R_w(\lambda)$ has full row rank (admissibility); update $[Q(\lambda) \ R(\lambda)] \leftarrow Q_2(\lambda)[Q(\lambda) \ R(\lambda)]$.

3. Compute the left coprime factorization

$$[Q(\lambda) \ R(\lambda)] = Q_3^{-1}(\lambda) [N_Q(\lambda) \ N_R(\lambda)]$$

such that $N_Q(\lambda)$ and $N_R(\lambda)$ have desired stable dynamics; update $[Q(\lambda) \ R(\lambda)] \leftarrow [N_Q(\lambda) \ N_R(\lambda)]$.

4. Compute the outer-co-inner factorization $R_w(\lambda) = R_{wo}(\lambda)R_{wi}(\lambda)$, where the outer factor $R_{wo}(\lambda)$ is invertible; with $Q_4(\lambda) = R_{wo}^{-1}(\lambda)$ update $[Q(\lambda) \ R(\lambda)] \leftarrow Q_4(\lambda)[Q(\lambda) \ R(\lambda)]$.

This procedure also illustrates the flexibility of the factorization-based synthesis, which allows to skip some of the synthesis steps. For example, if the least order synthesis is not of interest, then $Q_2(\lambda) = I$ at Step 2, and if additionally, the resulting $Q_1(\lambda)$ at Step 1 has an already satisfactory dynamics, then $Q_3(\lambda) = I$ at Step 3. Step 4 need not be performed if no noise inputs are present. The occurrence of the non-standard case with $Q_4(\lambda) = R_{wo}^{-1}(\lambda)$ unstable at Step 4 can be simply handled, for example, by repeating the computations at Step 3 after performing Step 4.

The **Procedure AFD** serves as a computational kernel for solving the FDIP by computing repeatedly the solutions of appropriately formulated FDPs. The synthesis procedure described below applies at the first step the nullspace-based paradigm to enforce the decoupling conditions $R_u(\lambda) = 0$, $R_d(\lambda) = 0$, and to obtain the reduced system (19). Then, it solves, for each row of S , a FDP for a reduced system with redefined sets of disturbance and fault inputs. The synthesis procedure, additionally determines the block rows of $R_f(\lambda)$ and $R_w(\lambda)$ corresponding to $Q^{(i)}(\lambda)$ as

$$[R_f^{(i)}(\lambda) \ R_w^{(i)}(\lambda)] := Q^{(i)}(\lambda) \begin{bmatrix} G_f(\lambda) & G_w(\lambda) \\ 0 & 0 \end{bmatrix}.$$

Procedure AFDI

Inputs: $\{G_u(\lambda), G_d(\lambda), G_f(\lambda), G_w(\lambda)\}$, $S \in \mathbb{R}^{n_b \times m_f}$

Outputs: $Q^{(i)}(\lambda)$, $R_f^{(i)}(\lambda)$, $R_w^{(i)}(\lambda)$

1. Compute a minimal left nullspace basis $Q(\lambda)$ of $G(\lambda)$ in (17) and

$$[R_f(\lambda) \ R_w(\lambda)] = Q(\lambda) \begin{bmatrix} G_f(\lambda) & G_w(\lambda) \\ 0 & 0 \end{bmatrix},$$

such that $Q(\lambda)$, $R_f(\lambda)$ and $R_w(\lambda)$ are stable.

2. For $i = 1, \dots, n_b$

2.1 Form $\bar{G}_d^{(i)}(\lambda)$ from the columns $R_{f_j}(\lambda)$ for which $S_{ij} = 0$ and $\bar{G}_f^{(i)}(\lambda)$ from the columns $R_{f_j}(\lambda)$ for which $S_{ij} \neq 0$.

2.2 Apply **Procedure AFD** to the system described by $\{0, \bar{G}_d^{(i)}(\lambda), \bar{G}_f^{(i)}(\lambda), R_w(\lambda)\}$ to obtain the stable filter $\bar{Q}^{(i)}(\lambda)$.

Exit if no solution exists.

2.3 Compute $Q^{(i)}(\lambda) = \bar{Q}^{(i)}(\lambda)Q(\lambda)$, $R_f^{(i)}(\lambda) = \bar{Q}^{(i)}(\lambda)R_f(\lambda)$ and $R_w^{(i)}(\lambda) = \bar{Q}^{(i)}(\lambda)R_w(\lambda)$.

The existence conditions of Theorem 2 are implicitly checked when performing **Procedure AFD** at Step 2.2.

Software Tools

The development of dedicated software tools for the synthesis of fault detection filters must fulfill some general requirements for robust software implementation, but also some requirements specific to the field of fault diagnosis. In what follows, we shortly discuss some of these requirements.

A basic requirement is the use of general, numerically reliable, and computationally efficient numerical approaches as basis for the implementation of all computational functions, to guarantee the solvability of problems under the most general existence conditions of the solutions. Synthesis procedures suitable for robust software implementations are described in the book (Varga 2017).

For beginners, it is important to be able to easily obtain preliminary synthesis results. Therefore, for ease of use, simple and uniform user interfaces are desirable for all synthesis functions. This can be achieved by relying on mean-

ingful default settings for all problem parameters and synthesis options.

For experienced users, an important requirement is to provide an exhaustive set of options to ensure the complete freedom in choosing problem-specific parameters and synthesis options.

The following synthesis options are frequently used: we mention the number of residual signal outputs; stability degree for the poles of the resulting filters or the location of their poles; type of the employed nullspace basis (e.g., minimal proper, full-order observer based); performing least order synthesis, etc.

The *Fault Detection and Isolation Tools* (FDITOOLS) is a publicly available collection of MATLAB *m*-files for the analysis and solution of fault diagnosis problems and has been implemented along the previously formulated requirements. FDITOOLS supports six basic synthesis approaches of linear residual generation filters for both continuous-time and discrete-time LTI systems. The underlying synthesis techniques rely on reliable numerical algorithms developed by the author and are described in Chapters 5–7 of the book (Varga 2017). The functions of the FDITOOLS collection rely on the *Control System Toolbox* (MathWorks 2015, and later releases) and the *Descriptor System Tools* (DSTOOLS) (Varga 2018a). DSTOOLS provides the basic tools for the manipulation of rational TFMs via their descriptor system representations. The version V1.0 of the FDITOOLS collection covers all synthesis procedures described in (Varga 2017) and, additionally, includes several useful analysis functions, as well as functions for an easy setup of synthesis models.

A precursor of FDITOOLS was the Fault Detection Toolbox for MATLAB, a proprietary software of the German Aerospace Center (DLR), developed between 2004 and 2014 (for the status of this toolbox around 2006, see (Varga 2006)).

Cross-References

- [Basic Numerical Methods and Software for Computer Aided Control Systems Design](#)

- [Descriptor System Techniques and Software Tools](#)
- [Fault Detection and Diagnosis](#)
- [Robust Fault Diagnosis and Control](#)

Recommended Reading

The book (Varga 2017) provides extensive information on the mathematical background of solving synthesis problems of fault detection filters, gives detailed descriptions of the underlying synthesis procedures and contains an extensive list of references to the relevant literature. The numerical aspects of the synthesis procedures are also amply described (a novelty in the fault detection related literature). Additionally, this book addresses the related model detection problematic, a multiple model based approach to solve fault diagnosis problems (e.g., for parametric or multiplicative faults). An extended version of this entry, available in arXiv (Varga 2019b), additionally addresses fault diagnosis problems related to model-matching techniques and provides an extended list of references. For a concise presentation of the descriptor system techniques employed in the synthesis procedures, see the companion article (Varga 2019a).

FDITOOLS is distributed as a free software via the Bitbucket repository <https://bitbucket.org/DSVarga/fditools>. A comprehensive documentation of version V1.0 is available in arXiv (Varga 2018b).

Bibliography

- MathWorks (2015) Control system toolbox (R2015b), user's guide. The MathWorks Inc., Natick
- Varga A (2006) A fault detection toolbox for Matlab. In: Proceedings of the IEEE conference on computer aided control system design, Munich, pp 3013–3018
- Varga A (2017) Solving fault diagnosis problems – linear synthesis techniques. Studies in systems, decision and control, vol 84. Springer International Publishing
- Varga A (2018a) Descriptor system tools (DSTOOLS) V0.71, user's guide. <https://arxiv.org/abs/1707.07140>
- Varga A (2018b) Fault detection and isolation tools (FDITOOLS) V1.0, user's guide. <https://arxiv.org/abs/1703.08480>

- Varga A (2019a) Descriptor system techniques and software tools. <https://arxiv.org/abs/1902.00009>
- Varga A (2019b) Fault detection and diagnosis: computational issues and tools. <https://arxiv.org/abs/1902.11186>

estimation (FE) · Fault-tolerant control · Passive FTC · Reconfigurable control

Fault-Tolerant Control

Ron J. Patton

School of Engineering, University of Hull, Hull, UK

Abstract

A closed-loop control system for an engineering process may have unsatisfactory performance or even instability when faults occur in actuators, sensors, or other process components. Fault-tolerant control (FTC) involves the development and design of special controllers that are capable of tolerating the actuator, sensor, and process faults while still maintaining desirable and robust performance and stability properties. FTC designs involve knowledge of the nature and/or occurrence of faults in the closed-loop system either implicitly or explicitly using methods of fault detection and isolation (FDI), fault detection and diagnosis (FDD), or fault estimation (FE). FTC controllers are reconfigured or restructured using FDI/FDD information so that the effects of the faults are reduced or eliminated within each feedback loop in *active* or *passive* approaches or compensated in each control-loop using FE methods. A non-mathematical outline of the essential features of FTC systems is given with important definitions and a classification of FTC systems into either active/passive approaches with examples of some well-known strategies.

Keywords

Active FTC · Fault accommodation · Fault detection and diagnosis (FDD) · Fault detection and isolation (FDI) · Fault

Synonyms

FTC

Introduction

The complexity of modern engineering systems has led to strong demands for enhanced control system reliability, safety, and green operation in the presence of even minor anomalies. There is a growing need not only to determine the onset and development of process faults before they become serious but also to adaptively compensate for their effects in the closed-loop system or using hardware redundancy to replace faulty components by duplicate and fault-free alternatives. The title “failure tolerant control” was given by Eterno et al. (1985) working on a reconfigurable flight control study defining the meaning of control system tolerance to failures or faults. The word “failure” is used when a fault is so serious that the system function concerned fails to operate (Isermann 2006). The title failure detection has now been superseded by *fault detection*, e.g., in *fault detection and isolation* (FDI) or *fault detection and diagnosis* (FDD) (Gertler 1998; Chen and Patton 1999; Patton et al. 2000) motivated by studies in the 1980s on this topic (Patton et al. 1989). *Fault-tolerant control* (FTC) began to develop in the early 1990s (Patton 1993) and is now a standard in the literature (Patton 1997; Blanke et al. 2006; Zhang and Jiang 2008), based on the aerospace subject of reconfigurable flight control making use of redundant actuators and sensors (Steinberg 2005; Edwards et al. 2010).

Definitions Relating to Fault-Tolerant Control

FTC is a strategy in control systems architecture and design to ensure that a closed-loop system

can continue acceptable operation in the face of bounded actuator, sensor, or process faults. The goal of FTC design must ensure that the closed-loop system maintains satisfactory stability and acceptable performance during either one or more fault actions. When prescribed stability and closed-loop performance indices are maintained despite the action of faults, the system is said to be “fault tolerant,” and the control scheme that ensures the fault tolerance is the fault-tolerant controller (Patton 1997; Blanke et al. 2006).

Fault modelling is concerned with the representation of the real physical faults and their effects on the system mathematical model. Fault modelling is important to establish how a fault should be detected, isolated, or compensated. The faults illustrated in Fig. 1 act at system locations defined as follows (Chen and Patton 1999):

An *actuator fault* ($f_a(t)$) corresponds to variations of the control input $u(t)$ applied to the controlled system either completely or partially. The complete failure of an actuator means that it produces no actuation regardless of the input applied to it, e.g., as a result of breakage and burnout of wiring. For partial actuator faults, the actuator becomes less effective and provides the plant with only a part of the normal actuation signal.

A sensor is an item of equipment that takes a measurement or observation from the system, e.g., potentiometers, accelerometers, tachometers, pressure gauges, strain gauges, etc.; a *sensor fault* ($f_s(t)$) implies that incorrect

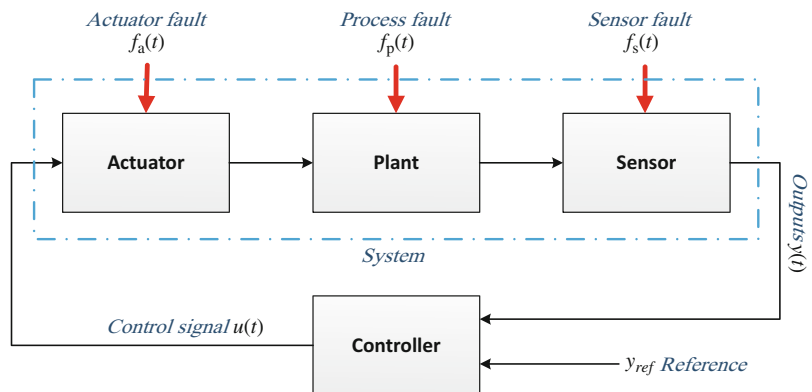
measurements are taken from the real system. This fault can also be subdivided into either a complete or partial sensor fault. When a sensor fails, the measurements no longer correspond to the required physical parameters. For a partial sensor fault the measurements give an inaccurate indication of required physical parameters.

A *process fault* ($f_p(t)$) directly affects the physical system parameters and in turn the input/output properties of the system. Process faults are often termed *component faults*, arising as variations from the structure or parameters used during system modelling, and as such cover a wide class of possible faults, e.g., dirty water having a different heat transfer coefficient compared to when it is clean, or changes in the viscosity of a liquid or components slowly degrading over time through wear and tear, aging, or environmental effects.

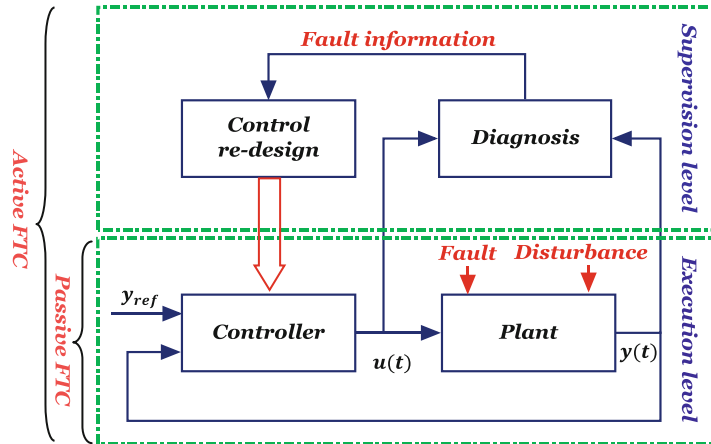
Architectures and Classification of FTC Schemes

FTC methods are classified according to whether they are “passive” or “active,” using fixed or reconfigurable control strategies (Eterno et al. 1985). Various architectures have been proposed for the implementation of FTC schemes, for example, the structure of reconfigurable control based on generalized internal model control (GIMC) has been proposed by Zhou and Ren (2001) and other studies by Niemann and

Fault-Tolerant Control,
Fig. 1 Closed-loop system
with actuator, process, and
sensor faults



Fault-Tolerant Control,
Fig. 2 Scheme of FTC
(Adapted from Blanke
et al. (2006))



Stoustrup (2005). Figure 2 shows a suitable architecture to encompass active and passive FTC methods in which a distinction is made between “execution” and “supervision” levels. The essential differences and requirements between the passive FTC (PFTC) and active FTC (AFTC).

PFTC is based solely on the use of robust control in which potential faults are considered as if they are uncertain signals acting in the closed-loop system. This can be related to the concept of reliable control (Veillette et al. 1992). PFTC requires no online information from the fault diagnosis (FDI/FDD/FE) function about the occurrence or presence of faults and hence it is not by itself an adaptive system and does not involve controller reconfiguration (Šiljak 1980; Patton 1993, 1997). PFTC approach can be used if the time window during which the system remains stabilizable in the presence of a fault is short; see, for example, the problem of the double inverted pendulum (Weng et al. 2007) which is unstable during a loop failure.

AFTC has *two* conceptual steps to provide the system with fault-tolerant capability (Patton 1997; Blanke et al. 2006; Zhang and Jiang 2008; Edwards et al. 2009):

- Equip the system with a mechanism to make it able to detect and isolate (or even estimate) the fault promptly, identify a faulty component, and select the required remedial action in to maintain acceptable operation performance.

With no fault a *baseline controller* attenuates disturbances and ensures good stability and closed-loop tracking performance (Patton 1997), and the diagnostic (FDI/FDD/FE) block recognizes that the closed-loop system is fault-free with no control law change required (supervision level).

- Make use of supervision level information and adapt or reconfigure/restructure the controller parameters so that the required remedial activity can be achieved (execution level).

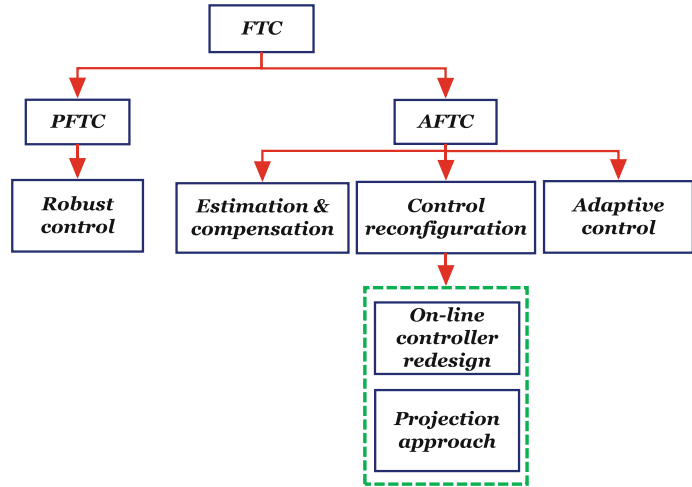
Figure 3 gives a classification of PFTC and AFTC methods (Patton 1997).

Figure 3 shows that AFTC approaches are divided into two main types of methods: projection-based methods and online automatic controller redesign methods. The latter involves the calculation of new controller parameters following control impairment, i.e., using reconfigurable control. In projection-based methods, a new precomputed control law is selected according to the required controller structure (i.e., depending on the type of isolated fault).

AFTC methods use online-fault accommodation based on unanticipated faults, classified as (Patton 1997):

- (a) Based on offline (pre-computed) control laws
- (b) Online-fault accommodating
- (c) Tolerant to unanticipated faults using FDI/FDD/FE
- (d) Dependent upon use of a baseline controller

Fault-Tolerant Control,
Fig. 3 General
 classification of FTC
 methods



AFTC Examples

One example of AFTC is model-based predictive control (MPC) which uses online computed control redesign. MPC is online-fault accommodating; it does not use an FDI/FDD unit and is not dependent on a baseline controller. MPC has a certain degree of fault tolerance against actuator faults under some conditions even if the faults are not detected. The representation of actuator faults in MPC is relatively natural and straightforward since actuator faults such as jams and slew-rate reductions can be represented by changing the MPC optimization problem constraints. Other faults can be represented by modifying the internal model used by MPC (Maciejowski 1998). The fact that online-fault information is not required means that MPC is an interesting method for flight control reconfiguration as demonstrated by Maciejowski and Jones in the GARTEUR AG 16 project on “fault-tolerant flight control” (Edwards et al. 2010).

Another interesting AFTC example that makes use of the concept of *model-matching* in explicit model following is the so-called pseudo-inverse method (PIM) (Gao and Antsaklis 1992) which requires the nominal or reference closed-loop system matrix to compute the new controller gain after a fault has occurred. The challenges are:

1. Guarantee of stability of the reconfigured closed-loop system

2. Minimization of the time consumed to approach the acceptable matching
3. Achieving perfect matching through use of different control methodologies

Exact model-matching may be too demanding, and some extensions to this approach make use of alternative, approximate (norm-based) model-matching through the computation of the required model-following gain. To relax the matching condition further, Staroswiecki (2005) proposed an *admissible model-matching* approach which was later extended by Tornil et al. (2010) using D-region pole assignment. The PIM approach requires an FDI/FDD/FE mechanism and is online-fault accommodating only in terms of *a priori* anticipated faults. This limits the practical value of this approach.

As a third example, feedback linearization can be used to compensate for nonlinear dynamic effects while also implementing control law reconfiguration or restructure. In flight control an aileron actuator fault will cause a strong coupling between the lateral and longitudinal aircraft dynamics. Feedback linearization is an established technique in flight control (Ochi and Kanai 1991). The faults are identified indirectly by estimating aircraft flight parameters online, e.g., using a recursive least-squares algorithm to update the FTC.

Hence, a AFTC system provides fault tolerance either by selecting a precomputed control law (*projection-based*) (Maybeck and Stevens 1991; Rauch 1995; Boskovic and Mehra 1999) or by synthesizing a new control strategy online (*online controller redesign*) (Ahmed-Zaid et al. 1991; Richter et al. 2007; Zou and Kumar 2011; Efimov et al. 2012).

Another widely studied AFTC method is the *estimation and compensation approach*, where a fault compensation input is superimposed on the nominal control input (Noura et al. 2000; Boskovic and Mehra 2002; Zhang et al. 2004; Sami and Patton 2013). There is a growing interest in robust FE methods based on sliding mode estimation (Edwards et al. 2000) and augmented observer methods (Gao and Ding 2007; Jiang et al. 2006; Sami and Patton 2013).

An important development of this approach is the so-called fault hiding strategy which is centered on achieving FTC loop goals such that the nominal control loop remains unchanged through the use of virtual actuators or virtual sensors (Blanke et al. 2006; Sami and Patton 2013). Fault hiding makes use of the difference between the nominal and faulty system state to changes in the system dynamics such that the required control objectives are continuously achieved even if a fault occurs. In the sensor fault case, the effect of the fault is hidden from the input of the controller. However, actuator faults are compensated by the effect of the fault (Lunze and Steffen 2006; Richter et al. 2007; Ponsart et al. 2010; Sami and Patton 2013) in which it is assumed that the FDI/FDD or FE scheme is available. The virtual actuator/sensor FTC can be good practical value if FDI/FDD/FE robustness can be demonstrated.

Traditional adaptive control methods that automatically adapt controller parameters to system changes can be used in a special application of AFTC, potentially removing the need for FDI/FDD and controller redesign steps (Tang et al. 2004; Zou and Kumar 2011) but possibly using the FE function. Adaptive control is suitable for FTC on plants that have slowly

varying parameters and can tolerate actuator and process faults. Sensor faults are not tolerated well as the controller parameters must adapt according to the faulty measurements, causing incorrect closed-loop system operation; the FDI/FDD/FE unit is required for such cases.

Summary and Future Directions

FTC is now a significant subject in control systems science with many quite significant application studies, particularly since the new millennium. Most of the applications are within the flight control field with studies such as the GARTEUR AG16 project “Fault-Tolerant Flight Control” (Edwards et al. 2010). As a very complex engineering-led and mathematically focused subject, it is important that FTC remains application-driven to keep the theoretical concepts moving in the right directions and satisfying end-user needs. The original requirement for FTC in safety-critical systems has now widened to encompass a good range of fault-tolerance requirements involving energy and economy, e.g., for greener aircraft and for FTC in renewable energy.

Faults and modelling uncertainties as well as endogenous disturbances have potentially competing effects on the control system performance and stability. This is the *robustness problem in FTC* which is beyond the scope of this article. The FTC system provides a degree of tolerance to closed-loop systems faults and it is also subject to the effects of modelling uncertainty arising from the reality that all engineering systems are nonlinear and can even have complex dynamics. For example, consider the PFTC approach relying on robustness principles – as a more complex extension to robust control. PFTC design requires the closed-loop system to be insensitive to faults as well as modelling uncertainties. This requires the use of multi-objective optimization methods e.g., using linear matrix inequalities (LMI), as well as methods of accounting for dynamical system parametric variations, e.g., linear parameter varying (LPV) system structures, Takagi-Sugeno, or sliding mode methods.

Cross-References

- [Diagnosis of Discrete Event Systems](#)
- [Estimation, Survey on](#)
- [Fault Detection and Diagnosis](#)
- [H-infinity Control](#)
- [LMI Approach to Robust Control](#)
- [Lyapunov's Stability Theory](#)
- [Model Reference Adaptive Control](#)
- [Optimization-Based Robust Control](#)
- [Robust Adaptive Control](#)
- [Robust Fault Diagnosis and Control](#)
- [Robust \$\mathcal{H}_2\$ Performance in Feedback Control](#)

Bibliography

- Ahmed-Zaid F, Ioannou P, Gousman K, Rooney R (1991) Accommodation of failures in the F-16 aircraft using adaptive control. *IEEE Control Syst* 11:73–78
- Blanke M, Kinnaert M, Lunze J, Staroswiecki M (2006) *Diagnosis and fault-tolerant control*. Springer, Berlin/New York
- Boskovic JD, Mehra RK (1999) Stable multiple model adaptive flight control for accommodation of a large class of control effector failures. In: *Proceedings of the ACC, San Diego, 2–4 June 1999*, pp 1920–1924
- Boskovic JD, Mehra RK (2002) An adaptive retrofit reconfigurable flight controller. In: *Proceedings of the 41st IEEE CDC, Las Vegas, 10–13 Dec 2002*, pp 1257–1262
- Chen J, Patton RJ (1999) *Robust model based fault diagnosis for dynamic systems*. Kluwer, Boston
- Edwards C, Spurgeon SK, Patton RJ (2000) Sliding mode observers for fault detection and isolation. *Automatica* 36:541–553
- Edwards C, Lombaerts T, Smaili H (2010) *Fault tolerant flight control a benchmark challenge*. Springer, Berlin/Heidelberg
- Efimov D, Cieslak J, Henry D (2012) Supervisory fault-tolerant control with mutual performance optimization. *Int J Adapt Control Signal Process* 27(4):251–279
- Eterno J, Weiss J, Looze D, Willsky A (1985) Design issues for fault tolerant restructurable aircraft control. In: *Proceedings of the 24th IEEE CDC, Fort-Lauderdale, Dec 1985*
- Gao Z, Antsaklis P (1992) Reconfigurable control system design via perfect model following. *Int J Control* 56:783–798
- Gao Z, Ding S (2007) Actuator fault robust estimation and fault-tolerant control for a class of nonlinear descriptor systems. *Automatica* 43:912–920
- Gertler J (1998) *Fault detection and diagnosis in engineering systems*. Marcel Dekker, New York
- Isermann R (2006) *Fault-diagnosis systems an introduction from fault detection to fault tolerance*. Springer, Berlin/New York
- Jiang BJ, Staroswiecki M, Cocquempot V (2006) Fault accommodation for nonlinear dynamic systems. *IEEE Trans Autom Control* 51:1578–1583
- Lunze J, Steffen T (2006) Control reconfiguration after actuator failures using disturbance decoupling methods. *IEEE Trans Autom Control* 51:1590–1601
- Maciejowski JM (1998) The implicit daisy-chaining property of constrained predictive control. *J Appl Math Comput Sci* 8(4):101–117
- Maybeck P, Stevens R (1991) Reconfigurable flight control via multiple model adaptive control methods. *IEEE Trans Aerosp Electron Syst* 27:470–480
- Niemann H, Stoustrup J (2005) An architecture for fault tolerant controllers. *Int J Control* 78(14):1091–1110
- Noura H, Sauter D, Hamelin F, Theilliol D (2000) Fault-tolerant control in dynamic systems: application to a winding machine. *IEEE Control Syst Mag* 20:33–49
- Ochi Y, Kanai K (1991) Design of restructurable flight control systems using feedback linearization. *J Guid Dyn Control* 14(5): 903–911
- Patton RJ, Frank PM, Clark RN (1989) *Fault diagnosis in dynamic Systems: theory and application*. Prentice Hall, New York
- Patton RJ (1993) Robustness issues in fault tolerant control. In: *Plenary paper at international conference TOOLDIAG'93, Toulouse, Apr 1993*
- Patton RJ (1997) Fault tolerant control: the 1997 situation. In: *IFAC Safeprocess '97, Hull*, pp 1033–1055
- Patton RJ, Frank PM, Clark RN (2000) *Issues of fault diagnosis for dynamic systems*. Springer, London/New York
- Ponsart J, Theilliol D, Aubrun C (2010) Virtual sensors design for active fault tolerant control system applied to a winding machine. *Control Eng Pract* 18:1037–1044
- Rauch H (1995) Autonomous control reconfiguration. *IEEE Control Syst* 15:37–48
- Richter JH, Schlage T, Lunze J (2007) Control reconfiguration of a thermofluid process by means of a virtual actuator. *IET Control Theory Appl* 1:1606–1620
- Sami M, Patton RJ (2013) Active fault tolerant control for nonlinear systems with simultaneous actuator and sensor faults. *Int J Control Autom Syst* 11(6):1149–1161
- Šiljak DD (1980) Reliable control using multiple control systems. *Int J Control* 31:303–329
- Staroswiecki M (2005) Fault tolerant control using an admissible model matching approach. In: *Joint Decision and Control Conference and European Control Conference, Seville, 12–15 Dec 2005*, pp 2421–2426
- Steinberg M (2005) Historical overview of research in reconfigurable flight control. *Proc IMechE, Part G J Aerosp Eng* 219:263–275
- Tang X, Tao G, Joshi S (2004) Adaptive output feedback actuator failure compensation for a class of non-linear systems. *Int J Adapt Control Signal Process* 19(6): 419–444

- Tornil S, Theilliol D, Ponsart JC (2010) Admissible model matching using $\mathcal{DR}$ -regions: fault accommodation and robustness against FDD inaccuracies *J Adapt Control Signal Process* 24(11): 927–943
- Veillette R, Medanic J, Perkins W (1992) Design of reliable control systems. *IEEE Trans Autom Control* 37:290–304
- Weng J, Patton RJ, Cui P (2007) Active fault-tolerant control of a double inverted pendulum. *J Syst Control Eng* 221:895
- Zhang Y, Jiang J (2008) Bibliographical review on reconfigurable fault-tolerant control systems. *Annu Rev Control* 32:229–252
- Zhang X, Parisini T, Polycarpou M (2004) Adaptive fault-tolerant control of nonlinear uncertain systems: an information-based diagnostic approach. *IEEE Trans Autom Control* 49(8):1259–1274
- Zhang K, Jiang B, Staroswiecki M (2010) Dynamic output feedback-fault tolerant controller design for Takagi-Sugeno fuzzy systems with actuator faults. *IEEE Trans Fuzzy Syst* 18:194–201
- Zhou K, Ren Z (2001) A new controller architecture for high performance, robust, and fault-tolerant control. *IEEE Trans Autom Control* 46(10):1613–1618
- Zou A, Kumar K (2011) Adaptive fuzzy fault-tolerant attitude control of spacecraft. *Control Eng Pract* 19:10–21

Feedback Linearization of Nonlinear Systems

Arthur J. Krener
Department of Applied Mathematics, Naval
Postgraduate School, Monterey, CA, USA

Abstract

Effective methods exist for the control of linear systems but this is less true for nonlinear systems. Therefore, it is very useful if a nonlinear system can be transformed into or approximated by a linear system. Linearity is not invariant under nonlinear changes of state coordinates and nonlinear state feedback. Therefore, it may be possible to convert a nonlinear system into a linear one via these transformations. This is called feedback linearization. This entry surveys feedback linearization and related topics.

Keywords

Distribution · Frobenius theorem ·
Involution distribution · Lie derivative

Introduction

A controlled linear dynamics is of the form

$$\dot{x} = Fx + Gu \quad (1)$$

where the state $x \in \mathbb{R}^n$ and the control $u \in \mathbb{R}^m$.
A controlled nonlinear dynamics is of the form

$$\dot{x} = f(x, u) \quad (2)$$

where x , u have the same dimensions but may be local coordinates on some manifolds $\mathcal{X}$, $\mathcal{U}$. Frequently, the dynamics is affine in the control, i.e.,

$$\dot{x} = f(x) + g(x)u \quad (3)$$

where $f(x) \in \mathbb{R}^{n \times 1}$ is a vector field and $g(x) = [g^1(x), \dots, g^m(x)] \in \mathbb{R}^{n \times m}$ is a matrix field.

Linear dynamics are much easier to analyze and control than nonlinear dynamics. For example, to globally stabilize the linear dynamics (1), all we need to do is to find a linear feedback law $u = Kx$ such that all the eigenvalues of $F + GK$ are in the open left half plane. Finding a feedback law $u = \kappa(x)$ to globally stabilize the nonlinear dynamics is very difficult and frequently impossible. Therefore, finding techniques to linearize nonlinear dynamics has been a goal for several centuries.

The simplest example of a linearization technique is to approximate a nonlinear dynamics around a critical point by its first-order terms. Suppose x^0 , u^0 is an operating point for the nonlinear dynamics (2), that is, $f(x^0, u^0) = 0$. Define displacement variables $z = x - x^0$ and $v = u - u^0$, and assuming $f(x, u)$ is smooth around this operating point, expand (2) to first order

$$\dot{z} = \frac{\partial f}{\partial x}(x^0, u^0)z + \frac{\partial f}{\partial u}(x^0, u^0)v + O(z, v)^2 \quad (4)$$

Ignoring the higher order terms, we get a linear dynamics (1) where

$$F = \frac{\partial f}{\partial x}(x^0, u^0), \quad G = \frac{\partial f}{\partial u}(x^0, u^0)$$

This simple technique works very well in many cases and is the basis for many engineering designs. For example, if the linear feedback $v = Kz$ puts all the eigenvalues of $F + GK$ in the left half plane, then the affine feedback $u = u^0 + K(x - x^0)$ makes the closed-loop dynamics locally asymptotically stable around x^0 . So, one way to linearize a nonlinear dynamics is to approximate it by a linear dynamics.

The other way to linearize is by a nonlinear change of state coordinates and a nonlinear state feedback because linearity is not invariant under these transformations. To see this, suppose we have a controlled linear dynamics (1) and we make the nonlinear change of state coordinates $z = \phi(x)$ and nonlinear feedback $u = \gamma(z, v)$. We assume that these transformations are invertible from some neighborhood of $x^0 = 0, u^0 = 0$ to some neighborhood of z^0, v^0 , and the inverse maps are

$$\begin{aligned} \psi(\phi(x)) &= x, & \phi(\psi(z)) &= z \\ \kappa(\psi(z), \gamma(z, v)) &= v, & \gamma(\phi(x), \kappa(x, u)) &= u \end{aligned}$$

then (1) becomes

$$\begin{aligned} \dot{z} &= \frac{\partial \phi}{\partial x}(x) (Fx + Gu) \\ &= \frac{\partial \phi}{\partial x}(\psi(z)) (F\psi(z) + G\gamma(z, v)) \end{aligned}$$

which is a controlled nonlinear dynamics (2). This raises the question asked by Brockett (1978), when is a controlled nonlinear dynamics a change of coordinate and feedback away from a controlled linear dynamics?

Linearization of a Smooth Vector Field

Let us start by addressing an apparently simpler question that was first considered by Poincaré. Given an uncontrolled nonlinear dynamics around a critical point, say $x^0 = 0$,

$$\dot{x} = f(x), \quad f(0) = 0,$$

find a smooth local change of coordinates

$$z = \phi(x), \quad \phi(0) = 0$$

which transforms it into an uncontrolled linear dynamics.

$$\dot{z} = Fz.$$

This question is apparently simpler, but as we shall see in the next section, the corresponding question for a controlled nonlinear dynamics that is affine in the control is actually easier to answer.

Without loss of generality, we can restrict our attention to changes of coordinates which carry $x^0 = 0$ to $z^0 = 0$ and whose Jacobian at this point is the identity, i.e.,

$$z = x + O(x^2)$$

then

$$F = \frac{\partial f}{\partial x}(0).$$

Poincaré's formal solution to this problem was to expand the vector field and the desired change of coordinates in a power series,

$$\begin{aligned} \dot{x} &= Fx + f^{[2]}(x) + O(x^3) \\ z &= x - \phi^{[2]}(x) \end{aligned}$$

where $f^{[2]}, \phi^{[2]}$ are n -dimensional vector fields, whose entries are homogeneous polynomials of degree 2 in x . A straightforward calculation yields

$$\dot{z} = Fz + f^{[2]}(x) - [Fx, \phi^{[2]}(x)] + O(x)^3$$

where the Lie bracket of two vector fields $f(x)$, $g(x)$ is the new vector field defined by

$$[f(x), g(x)] = \frac{\partial g}{\partial x}(x)f(x) - \frac{\partial f}{\partial x}(x)g(x).$$

Hence, $\phi^{[2]}(x)$ must satisfy the so-called homological equation (Arnol'd 1983)

$$[Fx, \phi^{[2]}(x)] = f^{[2]}(x).$$

This is a linear equation from the space of quadratic vector fields to the space of quadratic vector fields. The quadratic n -dimensional vector fields form a vector space of dimension n times $n + 1$ choose 2.

Poincaré showed that the eigenvalues of the linear map

$$\phi^{[2]}(x) \mapsto [Fx, \phi^{[2]}(x)] \quad (5)$$

are $\lambda_i + \lambda_j - \lambda_k$ where $\lambda_i, \lambda_j, \lambda_k$ are eigenvalues of F . If none of these expressions are zero, then the operator (5) is invertible. A degree two resonance occurs when $\lambda_i + \lambda_j - \lambda_k = 0$ and then the homological equation is not solvable for all $f^{[2]}(x)$.

Suppose a change of coordinates exists that linearizes the vector field up to degree r . In the new coordinates, the vector field is of the form

$$\dot{x} = Fx + f^{[r]}(x) + O(x)^{r+1}.$$

We seek a change of coordinates of the form

$$z = x - \phi^{[r]}(x)$$

to cancel the degree r terms, i.e., we seek a solution of the r th degree homological equation,

$$[Fx, \phi^{[r]}(x)] = f^{[r]}(x).$$

A degree r resonance occurs if

$$\lambda_{i_1} + \dots + \lambda_{i_r} - \lambda_k = 0.$$

If there is no resonance of degree r , then the degree r homological equation is uniquely solvable for every $f^{[r]}(x)$.

When there are no resonances of any degree, then the convergence of the formal power series solution is delicate. We refer the reader to Arnol'd (1983) for the details.

Linearization of a Controlled Dynamics by Change of State Coordinates

Given a controlled affine dynamics (3) when does there exist a smooth local change of coordinates

$$z = \phi(x), \quad 0 = \phi(0)$$

transforming it to

$$\dot{z} = Fz + Gu$$

where

$$F = \frac{\partial f}{\partial x}(0), \quad G = g(0)$$

This is an easier question to answer than that of Poincaré.

The controlled affine dynamics (3) is said to have well-defined controllability (Kronecker) indices if there exists a reordering of $g^1(x), \dots, g^m(x)$ and integers $r_1 \geq r_2 \geq \dots \geq r_m \geq 0$ such that $r_1 + \dots + r_m = n$, and the vector fields

$$\{ad^k(f)g^j : j = 1, \dots, m, k = 0, \dots, r_j - 1\}$$

are linearly independent at each x where g^i denotes the i th column of g and

$$ad^0(f)g^i = g^i, \quad ad^k(f)g^i = [f, ad^{k-1}(f)g^i].$$

If there are several sets of indices that satisfy this definition, then the controllability indices are the smallest in the lexicographic ordering.

A necessary and sufficient condition is that

$$[\text{ad}^k(f)g^i, \text{ad}^l(f)g^j] = 0$$

for $k = 0, \dots, n-1$, $l = 0, \dots, n$

The proof of this theorem is straightforward. Under a change of state coordinates, the vector fields and their Lie brackets are transformed by the Jacobian of the coordinate change. Trivially for linear systems,

$$\begin{aligned} \text{ad}^k(Fx)G^i &= (-1)^k F^k G \\ [\text{ad}^k(Fx)G^i, \text{ad}^l(Fx)G^j] &= 0. \end{aligned}$$

Feedback Linearization

We turn to a question posed and partially answered by Brockett (1978). Given a system affine in the m -dimensional control

$$\dot{x} = f(x) + g(x)u,$$

find a smooth local change of coordinates and smooth feedback

$$z = \phi(x), \quad u = \alpha(x) + \beta(x)v$$

transforming it to

$$\dot{z} = Fz + Gv$$

Brockett solved this problem under the assumptions that β is a constant and the control is a scalar, $m = 1$. The more general question for $\beta(x)$ and arbitrary m was solved in different ways by Korobov (1979), Jakubczyk and Respondek (1980), Sommer (1980), Hunt and Su (1981), Su (1982), and Hunt et al. (1983).

We describe the solution when $m = 1$. If the pair F, G is controllable, then there exist an H such that

$$\begin{aligned} HF^{k-1}G &= 0 & k &= 1, \dots, n-1 \\ HF^{n-1}G &= 1 \end{aligned}$$

If the nonlinear system is feedback linearizable, then there exists a function $h(x) = H\phi(x)$ such that

$$\begin{aligned} L_{\text{ad}^{k-1}(f)}g h &= 0 & k &= 1, \dots, n-1 \\ L_{\text{ad}^{n-1}(f)}g h &\neq 0 \end{aligned}$$

where the Lie derivative of a function h by a vector field g is given by

$$L_g h = \frac{\partial h}{\partial x} g$$

This is a system of first-order PDEs, and the solvability conditions are given by the classical Frobenius theorem, namely, that

$$\{g, \dots, \text{ad}^{n-2}(f)g\}$$

is involutive, i.e., its span is closed under Lie bracket.

For controllable systems, this is a necessary and sufficient condition. The controllability condition is that $\{g, \dots, \text{ad}^{n-1}(f)g\}$ spans x space.

Suppose $m = 2$ and the system has controllability (Kronecker) indices $r_1 \geq r_2$. Such a system is feedback linearizable iff

$$\{g^1, g^2, \dots, \text{ad}^{r_1-2}(f)g^1, \text{ad}^{r_1-2}(f)g^2\}$$

is involutive for $i = 1, 2$. Another way of putting is that the distribution spanned by the first through r th rows of the following matrix must be involutive for $r = r_i - 1$, $i = 1, 2$. This is equivalent to the distribution spanned by the first through r th rows of the following matrix being involutive for all $r = 1, \dots, r_1$.

$$\begin{bmatrix} g^1 & g^2 \\ \text{ad}(f)g & \text{ad}(f)g^2 \\ \vdots & \vdots \\ \text{ad}^{r_2-2}(f)g^1 & \text{ad}^{r_2-2}(f)g^2 \\ \text{ad}^{r_2-1}(f)g^1 & \text{ad}^{r_2-1}(f)g^2 \\ \vdots & \vdots \\ \text{ad}^{r_1-2}(f)g^1 & \vdots \\ \text{ad}^{r_1-1}(f)g^1 & \vdots \end{bmatrix}$$

One might ask if it is possible to use dynamic feedback to linearize a system that is not linearizable by static feedback. Suppose we treat one of the controls u_j as a state and let its derivative be a new control,

$$\dot{u}_j = \bar{u}_j$$

can the resulting system be linearized by state feedback and change of state coordinates? Loosely speaking, the effect of adding such an integrator to the j th control is to shift the j th column of the above matrix down by one row. This changes the distribution spanned by the first through r th rows of the above matrix and might make it involutive. A scalar input system $m = 1$ that is linearizable by dynamic state feedback is also linearizable by static state feedback. There are multi-input systems $m > 1$ that are dynamically linearizable but not statically linearizable (Charlet et al. 1989, 1991).

The generic system is not feedback linearizable, but mechanical systems with one actuator for each degree of freedom typically are feedback linearizable. This fact had been used in many applications, e.g., robotics, before the concept of feedback linearization.

One should not lose sight of the fact that stabilization, model-matching, or some other performance criterion is typically the goal of controller design. Linearization is a means to the goal. We linearize because we know how to meet the performance goal for linear systems.

Even when the system is linearizable, finding the linearizing coordinates and feedback can be a nontrivial task. Mechanical systems are the exception as the linearizing coordinates are usually the generalized positions. Since the $ad^{k-1}(f)g$ for $k = 1, \dots, n-1$ are characteristic directions of the PDE for h , the general solutions of the ODE's

$$\dot{x} = ad^{k-1}(f)g(x)$$

can be used to construct the solution (Blankenship and Quadrat 1984). The Gardner-Shadwick (GS) algorithm (1992) is the most efficient method that is known.

Linearization of discrete time systems was treated by Lee et al. (1986). Linearization of discrete time systems around an equilibrium manifold was treated by Barbot et al. (1995) and Jakubczyk (1987). Banaszuk and Hauser have also considered the feedback linearization of the transverse dynamics along a periodic orbit, (Banaszuk and Hauser 1995a, b).

Input–Output Linearization

Feedback linearization as presented above ignores the output of the system but typically one uses the input to control the output. Therefore, one wants to linearize the input–output response of the system rather than the dynamics. This was first treated in Isidori and Krener (1982) and Isidori and Ruberti (1984).

Consider a scalar input, scalar output system of the form

$$\dot{x} = f(x) + g(x)u, \quad y = h(x)$$

The relative degree of the system is the number of integrators between the input and the output. To be more precise, the system is of relative degree $r \geq 1$ if for all x of interest,

$$L_{ad^j(f)g}h(x) = 0 \quad j = 0, \dots, r-2$$

$$L_{ad^{r-1}(f)g}h(x) \neq 0$$

In other words, the control appears first in the r th time derivative of the output. Of course, a system might not have a well-defined relative degree as the r might vary with x .

Rephrasing the result of the previous section, a scalar input nonlinear system is feedback linearizable if there exist an pseudo-output map $h(x)$ such the resulting scalar input, scalar output system has a relative degree equal to the state dimension n .

Assume we have a scalar input, scalar output system with a well-defined relative degree $1 \leq r \leq n$. We can define r partial coordinate functions

$$\zeta_i(x) = (L_f)^{i-1}h(x) \quad i = 1, \dots, r$$

and choose $n - r$ functions $\xi_i(x)$, $i = 1, \dots, n - r$ so that (ζ, ξ) are a full set of coordinates on the state space. Furthermore, it is always possible (Isidori 1995) to choose $\xi_i(x)$ so that

$$L_g \xi_i(x) = 0 \quad i = 1, \dots, n - r$$

In these coordinates, the system is in the normal form

$$\begin{aligned} y &= \zeta_1 \\ \dot{\zeta}_1 &= \zeta_2 \\ &\vdots \\ \dot{\zeta}_{r-1} &= \zeta_r \\ \dot{\zeta}_r &= f_r(\zeta, \xi) + g_r(\zeta, \xi)u \\ \dot{\xi} &= \phi(\zeta, \xi) \end{aligned}$$

The feedback $u = u(\zeta, \xi, v)$ defined by

$$u = \frac{(v - f_r(\zeta, \xi))}{g_r(\zeta, \xi)}$$

transforms the system to

$$\begin{aligned} y &= H\zeta \\ \dot{\zeta} &= F\zeta + Gv \\ \dot{\xi} &= \phi(\zeta, \xi) \end{aligned}$$

where F , G , H are the $r \times r$, $r \times 1$, $1 \times r$ matrices

$$F = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$H = [1 \ 0 \ 0 \ \dots \ 0].$$

The system has been transformed into a string of integrators plus additional dynamics that is unobservable from the output.

By suitable choice of additional feedback $v = K\zeta$, one can insure that the poles of $F + GK$ are stable. The stability of the overall system then depends on the stability of the zero dynamics (Byrnes and Isidori 1984, 1988),

$$\dot{\xi} = \phi(0, \xi)$$

If this is stable, then the overall system will be stable. The zero dynamics is so-called because it is the dynamics that results from imposing the constraint $y(t) = 0$ on the system. For this to be satisfied, the initial value must satisfy $\zeta(0) = 0$ and the control must satisfy

$$u(t) = -\frac{f_r(0, \xi(t))}{g_r(0, \xi(t))}$$

Similar results hold in the multiple input, multiple output case, see Isidori (1995) for the details.

Approximate Feedback Linearization

Since so few controlled dynamics are exactly feedback linearizable, Krener (1984) introduced the concept of approximate feedback linearization. The goal is to find a smooth local change of coordinates and a smooth feedback

$$z = \phi(x), \quad u = \alpha(x) + \beta(x)v$$

transforming the affinity controlled dynamics (3) to

$$\dot{z} = Fz + Gv + N(x, u)$$

where the nonlinearity $N(x, u)$ is small in some sense. In the design process, the nonlinearity is ignored, and the controller design is done on the linear model and then transformed back into a controller for the original system.

The power series approach of Poincaré was taken by Krener et al. (1987, 1988, 1991), Krener (1990), and Krener and Maag (1991). See also Kang (1994). It is applicable to dynamics which may not be affine in the control. The controlled nonlinear dynamics (2), the change of coordinates, and the feedback are expanded in

a power series

$$\dot{x} = Fx + Gu + f^{[2]}(x, u) + O(x, u)^3$$

$$z = x - \phi^{[2]}(x)$$

$$v = u - \alpha^{[2]}(x, u)$$

The transformed system is

$$\dot{z} = Fz + Gv + f^{[2]}(x, u)$$

$$- \left[Fx + Gu, \phi^{[2]}(x) \right] + G\alpha^{[2]}(x, u) \\ + O(x, u)^3$$

To eliminate the quadratic terms, one seeks a solution of the degree two homological equations for $\phi^{[2]}$, $\alpha^{[2]}$

$$\left[Fx + Gu, \phi^{[2]}(x) \right] - G\alpha^{[2]}(x, u) = f^{[2]}(x, u)$$

Unlike before, the degree two homological equations are not square. Almost always, the number of unknowns is less than the number of equations. Furthermore, the the mapping

$$\left(\phi^{[2]}(x), \alpha^{[2]}(x, u) \right) \mapsto \left[Fx + Gu, \phi^{[2]}(x) \right] \\ - G\alpha^{[2]}(x, u)$$

is less than full rank. Hence, only an approximate, e.g., a least squares solution, is possible. Krener has written a MATLAB toolbox (<http://www.math.ucdavis.edu/~krener> 1995) to compute term by term solutions to the homological equations. The routine `fh2f_h.m` sequentially computes the least square solutions of the homological equations to arbitrary degree.

Observers with Linearizable Error Dynamics

The dual of linear state feedback is linear input–output injection. Linear input–output injection is

the transformation carrying

$$\dot{x} = Fx + Gu$$

$$y = Hx$$

into

$$\dot{x} = Fx + Bu + Ly + Mu$$

$$y = Hx$$

Linear input–output injection and linear change of state coordinates

$$\dot{x} = Fx + Bu + Ly + Mu$$

$$y = Hx$$

$$z = Tx$$

$$\dot{z} = TFT^{-1}x + TLy + TMu$$

define a group action on the class of linear systems. Of course, output injection is not physically realizable on the original system, but it is realizable on the observer error dynamics.

Nonlinear input–output injection is not well defined independent of the coordinates; input–output injection in one coordinate system does not look like input–output injection in another coordinate system.

If a system

$$\dot{x} = f(x, u), \quad y = h(x)$$

can be transformed by nonlinear changes of state and output coordinates

$$z = \phi(x), \quad w = \gamma(y)$$

to a linear system with nonlinear input–output injection

$$\dot{z} = Fz + Gu + \alpha(y, u)$$

$$w = Hz$$

then the observer

$$\dot{\hat{z}} = (F + LH)\hat{z} + Gu + \alpha(y, u) - Lw$$

has linear error dynamics

$$\begin{aligned}\tilde{z} &= z - \hat{z} \\ \dot{\tilde{z}} &= (F + LH)\tilde{z}\end{aligned}$$

If H, F is detectable, then $F + LH$ can be made Hurwitz, i.e., all its eigenvalues are in the open left half plane.

The case when $\gamma = \text{identity}$, there are no inputs $m = 0$ and one output $p = 1$,

$$\begin{aligned}\dot{x} &= f(x) \\ y &= h(x)\end{aligned}$$

was solved by Krener and Isidori (1983) and Bestle and Zeitz (1983) when the pair H, F defined by

$$\begin{aligned}F &= \frac{\partial f}{\partial x}(0) \\ H &= \frac{\partial h}{\partial x}(0)\end{aligned}$$

is observable.

One seeks a change of coordinates $z = \phi(x)$ so that the system is linear up to output injection

$$\begin{aligned}\dot{z} &= Fz + \alpha(y) \\ y &= Hz\end{aligned}$$

If they exist, the z coordinates satisfy the PDE's

$$L_{ad^{n-k}(f)g}(z_j) = \delta_{k,j}$$

where the vector field $g(x)$ is defined by

$$L_g L_f^{k-1} h = \begin{cases} 0 & 1 \leq k < n \\ 1 & k = n \end{cases}$$

The solvability conditions for these PDE's are that for $1 \leq k < l \leq n - 1$

$$[ad^{k-1}(f)g, ad^{l-1}(f)g] = 0.$$

The general case with γ, m, p arbitrary was solved by Krener and Respondek (1985). The solution is a three-step process. First, one must set up and solve a linear PDE for $\gamma(y)$. The integrability conditions for this PDE involve the vanishing of a pseudo-curvature (Krener 1986). The next two steps are similar to the above. One defines a vector field g^j , $1 \leq j \leq p$ for each output, these define a PDE for the change of coordinates, for which certain integrability conditions must be satisfied. The process is more complicated than feedback linearization and even less likely to be successful so approximate solutions must be sought which we will discuss later in this section. We refer the reader Krener and Respondek (1985) and related work Zeitz (1987) and Xia and Gao (1988a, b, 1989).

Very few systems can be linearized by change of state coordinates and input–output injection, so Krener et al. (1987, 1988, 1991), Krener (1990), and Krener and Maag (1991) sought approximate solutions by the power series approach. Again, the system, the changes of coordinates, and the output injection are expanded in a power series. See the above references for details.

Conclusion

We have surveyed the various ways a nonlinear system can be approximated by a linear system.

Cross-References

- [Differential Geometric Methods in Nonlinear Control](#)
- [Lie Algebraic Methods in Nonlinear Control](#)
- [Nonlinear Zero Dynamics](#)

Bibliography

- Arnol'd VI (1983) Geometrical methods in the theory of ordinary differential equations. Springer, Berlin
- Banaszuk A, Hauser J (1995a) Feedback linearization of transverse dynamics for periodic orbits. Syst Control Lett 26:185–193

- Banaszuk A, Hauser J (1995b) Feedback linearization of transverse dynamics for periodic orbits in $\mathbf{R}^3$ with points of transverse controllability loss. *Syst Control Lett* 26:95–105
- Barbot JP, Monaco S, Normand-Cyrot D (1995) Linearization about an equilibrium manifold in discrete time. In: *Proceedings of IFAC NOLCOS 95*, Tahoe City. Pergamon
- Barbot JP, Monaco S, Normand-Cyrot D (1997) Quadratic forms and approximate feedback linearization in discrete time. *Int J Control* 67:567–586
- Bestle D, Zeitz M (1983) Canonical form observer design for non-linear time-variable systems. *Int J Control* 38:419–431
- Blankenship GL, Quadrat JP (1984) An expert system for stochastic control and signal processing. In: *Proceedings of IEEE CDC, Las Vegas*. IEEE, pp 716–723
- Brockett RW (1978) Feedback invariants for nonlinear systems. In: *Proceedings of the international congress of mathematicians, Helsinki*, pp 1357–1368
- Brockett RW (1981) Control theory and singular Riemannian geometry. In: Hilton P, Young G (eds) *New directions in applied mathematics*. Springer, New York, pp 11–27
- Brockett RW (1983) Nonlinear control theory and differential geometry. In: *Proceedings of the international congress of mathematicians, Warsaw*, pp 1357–1368
- Brockett RW (1996) Characteristic phenomena and model problems in nonlinear control. In: *Proceedings of the IFAC congress, Sidney*. IFAC
- Byrnes CI, Isidori A (1984) A frequency domain philosophy for nonlinear systems. In: *Proceedings, IEEE conference on decision and control, Las Vegas*. IEEE, pp 1569–1573
- Byrnes CI, Isidori A (1988) Local stabilization of minimum-phase nonlinear systems. *Syst Control Lett* 11:9–17
- Charlet R, Levine J, Marino R (1989) On dynamic feedback linearization. *Syst Control Lett* 13:143–151
- Charlet R, Levine J, Marino R (1991) Sufficient conditions for dynamic state feedback linearization. *SIAM J Control Optim* 29:38–57
- Gardner RB, Shadwick WF (1992) The GS-algorithm for exact linearization to Brunovsky normal form. *IEEE Trans Autom Control* 37:224–230
- Hunt LR, Su R (1981) Linear equivalents of nonlinear time varying systems. In: *Proceedings of the symposium on the mathematical theory of networks and systems, Santa Monica*, pp 119–123
- Hunt LR, Su R, Meyer G (1983) Design for multi-input nonlinear systems. In: Millman RS, Brockett RW, Sussmann HJ (eds) *Differential geometric control theory*. Birkhauser, Boston, pp 268–298
- Isidori A (1995) *Nonlinear control systems*. Springer, Berlin
- Isidori A, Krener AJ (1982) On the feedback equivalence of nonlinear systems. *Syst Control Lett* 2:118–121
- Isidori A, Ruberti A (1984) On the synthesis of linear input–output responses for nonlinear systems. *Syst Control Lett* 4:17–22
- Jakubczyk B (1987) Feedback linearization of discrete-time systems. *Syst Control Lett* 9:17–22
- Jakubczyk B, Respondek W (1980) On linearization of control systems. *Bull Acad Polonaise Sci Ser Sci Math* 28:517–522
- Kang W (1994) Approximate linearization of nonlinear control systems. *Syst Control Lett* 23: 43–52
- Korobov VI (1979) A general approach to the solution of the problem of synthesizing bounded controls in a control problem. *Math USSR-Sb* 37:535
- Krener AJ (1973) On the equivalence of control systems and the linearization of nonlinear systems. *SIAM J Control* 11:670–676
- Krener AJ (1984) Approximate linearization by state feedback and coordinate change. *Syst Control Lett* 5: 181–185
- Krener AJ (1986) The intrinsic geometry of dynamic observations. In: Fliess M, Hazewinkel M (eds) *Algebraic and geometric methods in nonlinear control theory*. Reidel, Amsterdam, pp 77–87
- Krener AJ (1990) Nonlinear controller design via approximate normal forms. In: Helton JW, Grunbaum A, Khargonekar P (eds) *Signal processing, Part II: control theory and its applications*. Springer, New York, pp 139–154
- Krener AJ, Isidori A (1983) Linearization by output injection and nonlinear observers. *Syst Control Lett* 3: 47–52
- Krener AJ, Maag B (1991) Controller and observer design for cubic systems. In: Gombani A, DiMasi GB, Kurzhansky AB (eds) *Modeling, estimation and control of systems with uncertainty*. Birkhauser, Boston, pp 224–239
- Krener AJ, Respondek W (1985) Nonlinear observers with linearizable error dynamics. *SIAM J Control Optim* 23:197–216
- Krener AJ, Karahan S, Hubbard M, Frezza R (1987) Higher order linear approximations to nonlinear control systems. In: *Proceedings, IEEE conference on decision and control, Los Angeles*. IEEE, pp 519–523
- Krener AJ, Karahan S, Hubbard M (1988) Approximate normal forms of nonlinear systems. In: *Proceedings, IEEE conference on decision and control, San Antonio*. IEEE, pp 1223–1229
- Krener AJ, Hubbard M, Karahan S, Phelps A, Maag B (1991) Poincaré's linearization method applied to the design of nonlinear compensators. In: Jacob G, Lamnabhi-Lagarigue F (eds) *Algebraic computing in control*. Springer, Berlin, pp 76–114
- Lee HG, Arapostathis A, Marcus SI (1986) Linearization of discrete-time systems. *Int J Control* 45:1803–1822
- Murray RM (1995) Nonlinear control of mechanical systems: a Lagrangian perspective. In: Krener AJ, Mayne DQ (eds) *Nonlinear control systems design*. Pergamon, Oxford

- Nonlinear Systems Toolbox available for down load at <http://www.math.ucdavis.edu/~krener/1995>
- Sommer R (1980) Control design for multivariable nonlinear time varying systems. *Int J Control* 31:883–891
- Su R (1982) On the linear equivalents of control systems. *Syst Control Lett* 2:48–52
- Xia XH, Gao WB (1988a) Nonlinear observer design by canonical form. *Int J Control* 47:1081–1100
- Xia XH, Gao WB (1988b) On exponential observers for nonlinear systems. *Syst Control Lett* 11:319–325
- Xia XH, Gao WB (1989) Nonlinear observer design by observer error linearization. *SIAM J Control Optim* 27:199–216
- Zeitz M (1987) The extended Luenberger observer for nonlinear systems. *Syst Control Lett* 9:149–156

Feedback Stabilization of Nonlinear Systems

Alessandro Astolfi

Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma Tor Vergata, Rome, Italy

Abstract

We consider the simplest design problem for nonlinear systems: the problem of rendering asymptotically stable a given equilibrium by means of feedback. For such a problem, we provide a necessary condition, known as Brockett condition, and a sufficient condition, which relies upon the definition of a class of functions, known as control Lyapunov functions. The theory is illustrated by means of a few examples. In addition we discuss a nonlinear enhancement of the so-called separation principle for stabilization by means of partial state information.

Keywords

Stability · Stabilization · Feedback · Nonlinear systems

Introduction

The problem of feedback stabilization, namely, the problem of designing a feedback control law locally, or globally, asymptotically stabilizing a given equilibrium point, is the simplest design problem for nonlinear systems. If the state of the system is available for feedback, then the problem is referred to as the state feedback stabilization problem, whereas if only part of the state, for example, an output signal, is available for feedback, the problem is referred to as the partial state feedback (or output feedback) stabilization problem. We initially focus on the state feedback stabilization problem which can be formulated as follows.

Consider a nonlinear system described by the equation

$$\dot{x} = F(x, u), \quad (1)$$

where $x(t) \in \mathbb{R}^n$ denotes the state of the system, $u(t) \in \mathbb{R}^m$ denotes the input of the system, and $F : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a smooth mapping.

Let $x_0 \in \mathbb{R}^n$ be an *achievable* equilibrium, i.e., x_0 is such that there exists a constant $u_0 \in \mathbb{R}^m$ such that $F(x_0, u_0) = 0$. The state feedback stabilization problem consists in finding, if possible, a state feedback control law, described by the equation

$$u = \alpha(x), \quad (2)$$

with $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}^m$, such that the equilibrium x_0 is a locally asymptotically stable equilibrium for the closed-loop system

$$\dot{x} = F(x, \alpha(x)). \quad (3)$$

Alternatively, one could require that the equilibrium be globally asymptotically stable. Note that it is not always possible to *extend* local properties to global properties. For example, for the system described by the equations

$$\dot{x}_1 = x_2(1 - x_1^2), \quad \dot{x}_2 = u,$$

with $x_1(t) \in \mathbb{R}$, $x_2(t) \in \mathbb{R}$, and $u(t) \in \mathbb{R}$, it is not possible to design a feedback law which renders the zero equilibrium globally asymptotically stable.

If only partial information on the state is available, then one has to resort to a dynamic *output* feedback controller, namely, a controller described by equations of the form

$$\dot{\xi} = \beta(\xi, y), \quad u = \alpha(\xi), \quad (4)$$

where $\xi(t) \in \mathbb{R}^v$ describes the state of the controller, $y(t) \in \mathbb{R}^p$ is given by $y = h(x)$, for some mapping $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$, and describes the available information on the state x , and $\beta : \mathbb{R}^v \times \mathbb{R}^p \rightarrow \mathbb{R}^v$ and $\alpha : \mathbb{R}^v \rightarrow \mathbb{R}^m$ are smooth mappings. Within this scenario the stabilization problem boils down to selecting the (nonnegative) integer v (i.e., the order of the controller), a constant $\xi_0 \in \mathbb{R}^v$ and the mappings α and β such that the closed-loop system

$$\dot{x} = F(x, \alpha(x)), \quad \dot{\xi} = \beta(\xi, h(x)), \quad (5)$$

has a locally (or globally) asymptotically stable equilibrium at (x_0, ξ_0) . Alternatively, one may require that the equilibrium (x_0, ξ_0) of the closed-loop system (5) be locally asymptotically stable with a region of attraction that contains a given, user-specified, set.

The rest of the entry is organized as follows. We begin discussing two key results. The first is a necessary condition, due to R.W. Brockett, for continuous stabilizability. This provides an obstruction to the solvability of the problem and can be used to show that, for nonlinear systems, controllability does not imply stabilizability by continuous feedback. The second one is the extension of the Lyapunov direct method to systems with control. The main idea is the introduction of a control version of Lyapunov functions, the *control Lyapunov functions*, which can be used to design stabilizing control laws by means of a *universal formula*. We then describe two classes of systems for which it is possible to construct, with systematic procedures, smooth control laws yielding global asymptotic stability of a given equilibrium: systems in feedback and in feedforward form. There are several other constructive and systematic stabilization methods which have been developed in the last few

decades. Worth mentioning are passivity-based methods and center manifold-based methods.

We conclude the entry describing a nonlinear version of the separation principle for the asymptotic stabilization, by output feedback, of a general class of nonlinear systems.

Preliminary Results

To highlight the difficulties and peculiarities of the nonlinear stabilization problem, we recall some basic facts from linear systems theory and exploit such facts to derive a sufficient condition and a necessary condition. In the case of linear systems, i.e., systems described by the equation $\dot{x} = Ax + Bu$, with $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, and linear state feedback, i.e., feedback described by the equation $u = Kx$, with $K \in \mathbb{R}^{m \times n}$, the stabilization problem boils down to the problem of placing, in the complex plane, the eigenvalues of the matrix $A + BK$ to the left of the imaginary axis. This problem is solvable if and only if the uncontrollable modes of the system are located, in the complex plane, to the left of the imaginary axis.

The linear theory may be used to provide a simple obstruction to feedback stabilizability and a simple sufficient condition. Let x_0 be an achievable equilibrium with $u_0 = 0$ and note that the linear approximation of the system (1) around x_0 is described by an equation of the form $\dot{x} = Ax + Bu$.

If for any $K \in \mathbb{R}^{m \times n}$ the condition (the notation $\sigma(A)$ denotes the spectrum of the matrix A , i.e., the set of the eigenvalues of A)

$$\sigma(A + BK) \cap C^+ \neq \emptyset \quad (6)$$

holds, then the equilibrium of the nonlinear system cannot be stabilized by any continuously differentiable feedback such that $\alpha(x_0) = 0$. Note however that, if the condition $\alpha(x_0) = 0$ is dropped, the obstruction does not hold: the zero equilibrium of $\dot{x} = x + xu$ is not stabilizable by any continuous feedback such that $\alpha(0) = 0$, yet the feedback $u = -2$ is a (global) stabilizer.

On the contrary, if there exists a K such that

$$\sigma(A + BK) \subset C^-$$

then the feedback $\alpha(x) = K(x - x_0)$ locally asymptotically stabilizes the equilibrium x_0 of the closed-loop system. This fact is often referred to as *the linearization approach*.

The above linear arguments are often inadequate to design feedback stabilizers: a theory for nonlinear feedback has to be developed. However, this theory is much more involved. In particular, it is important to observe that the solvability of the stabilization problem may depend upon the regularity properties of the feedback i.e., of the mapping α . In fact, a given equilibrium of a nonlinear system may be rendered locally asymptotically stable by a continuous feedback, whereas there may be no continuously differentiable feedback achieving the same goal. If the feedback is required to be continuously differentiable, then the problem is often referred to as the *smooth stabilization problem*.

Example To illustrate the role of the regularity properties of the feedback consider the system described by the equations

$$\dot{x}_1 = x_1 - x_2^3, \quad \dot{x}_2 = u,$$

with $x_1(t) \in \mathbb{R}$, $x_2(t) \in \mathbb{R}$, and $u(t) \in \mathbb{R}$, and the equilibrium $x_0 = (0, 0)$. The equilibrium is globally asymptotically stabilized by the continuous feedback

$$\alpha(x) = -x_2 + x_1 + \frac{4}{3}x_1^{\frac{1}{3}} - x_2^3,$$

but it is not stabilizable by any continuously differentiable feedback. Note, in fact, that condition (6) holds.

Brockett Theorem

Brockett's necessary condition, which is far from being sufficient, provides a simple test to rule out the existence of a continuous stabilizer.

Theorem 1 Consider the system (1) and assume $x_0 = 0$ is an achievable equilibrium with $u_0 = 0$.

Assume there exists a continuous stabilizing feedback $u = \alpha(x)$. Then, for each $\varepsilon > 0$ there exists $\delta > 0$ such that, for all y with $\|y\| < \delta$, the equation $y = F(x, u)$ has at least one solution in the set $\|x\| < \varepsilon$, $\|u\| < \varepsilon$.

Theorem 1 can be reformulated as follows. The existence of a continuous stabilizer implies that the image of the mapping $F : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ covers a neighborhood of the origin. The obstruction expressed by Theorem 1 is of topological nature: it requires continuity of F and α and time-invariance, that is, the theorem is not applicable if $u = \alpha(x, t)$, i.e., a time-varying feedback is considered.

In the linear case, Brockett condition reduces to the condition

$$\text{rank}[A - \lambda I, B] = n$$

for $\lambda = 0$. This is a necessary, but clearly not sufficient, condition for the stabilizability of $\dot{x} = Ax + Bu$.

Example Consider the kinematic model of a mobile robot given by the equations

$$\begin{aligned} \dot{x} &= \cos \theta v, \\ \dot{y} &= \sin \theta v, \\ \dot{\theta} &= \omega, \end{aligned}$$

where $(x(t), y(t)) \in \mathbb{R}^2$ denotes the Cartesian position of the robot, $\theta(t) \in (-\pi, \pi]$ denotes the robot orientation (with respect to the x -axis), $v(t) \in \mathbb{R}$ is the forward velocity of the robot, and $\omega(t) \in \mathbb{R}$ is its angular velocity. Simple intuitive considerations suggest that the system is controllable, i.e., it is possible to select the forward and angular velocities to drive the robot from any initial position/orientation to any final position/orientation in any given positive time. Nevertheless, the zero equilibrium (and any other equilibrium of the system) is not continuously stabilizable. In fact, the equations

$$y_1 = \cos \theta v, \quad y_2 = \sin \theta v, \quad y_3 = \omega,$$

with $\|(y_1, y_2, y_3)\| < \delta$ and $\|(x, y, \theta)\| < \varepsilon$, $\|(v, \omega)\| < \varepsilon$ are in general not solvable. For

example, if $\varepsilon < \pi/2$ and $y_1 = 0$, $y_2 \neq 0$, $y_3 = 0$, then the unique solution of the first and third equations is $v = 0$ and $\omega = 0$, implying $\sin \theta v = 0$; hence the second equation does not have a solution.

Control Lyapunov Functions

The Lyapunov theory states that the equilibrium x_0 of the system

$$\dot{x} = f(x),$$

with $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, is locally asymptotically stable if there exists a continuously differentiable function $V : \mathbb{R}^n \rightarrow \mathbb{R}$, called Lyapunov function, and a neighborhood U of x_0 such that $V(x_0) = 0$, $V(x) > 0$, for all $x \in U$ and $x \neq x_0$, and $\frac{\partial V}{\partial x} f(x) < 0$, for all $x \in U$ and $x \neq x_0$.

To apply this idea to the stabilization problem, consider the system (1). If the equilibrium x_0 of $\dot{x} = F(x, u)$ is continuously stabilizable then there must exist a continuously differentiable function V and a neighborhood U of x_0 such that

$$\inf_u \frac{\partial V}{\partial x} F(x, u) < 0,$$

for all $x \in U$ and $x \neq x_0$. This motivates the following definition.

Definition 1 A continuously differentiable function V such that

- $V(x_0) = 0$ and $V(x) > 0$, for all $x \in U$ and $x \neq x_0$, where U is a neighborhood of x_0 ;
- $\inf_u \frac{\partial V}{\partial x} F(x, u) < 0$, for all $x \in U$ and $x \neq x_0$;

is called a *control Lyapunov function*.

By Lyapunov theory, the existence of a continuous stabilizer implies the existence of a con-

trol Lyapunov function. On the other hand, the existence of a control Lyapunov function does not guarantee the existence of a stabilizer. However, in the case of systems affine in the control, i.e., systems described by the equation

$$\dot{x} = f(x) + g(x)u, \quad (7)$$

with $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ smooth mappings, very general results can be proven. These have been proven by Z. Artstein, who gave a non-constructive statement, and have been given a constructive form by E.D. Sontag. In particular, for single-input nonlinear systems, the following statement holds.

Theorem 2 Consider the system (7), with $m = 1$, and assume $f(0) = 0$. There exists an almost smooth feedback which globally asymptotically stabilizes the equilibrium $x = 0$ if and only if there exists a positive definite, radially unbounded and smooth function $V(x)$ such that

1. the condition $\frac{\partial V}{\partial x} g(x) = 0$ implies $\frac{\partial V}{\partial x} f(x) < 0$, for all $x \neq 0$;
2. for each $\varepsilon > 0$ there is a $\delta > 0$ such that $\|x\| < \delta$ implies that there is a $|u| < \varepsilon$ such that

$$\frac{\partial V}{\partial x} f(x) + \frac{\partial V}{\partial x} g(x)u < 0.$$

Recall that *almost smooth* means that the feedback $\alpha(x)$ is continuously differentiable, for all $x \in \mathbb{R}^n$ and $x \neq 0$, and it is continuous at $x = 0$. A function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ is radially unbounded if $\lim_{\|x\| \rightarrow \infty} V(x) = \infty$.

Condition 2 is known as the *small control property*, and it is necessary to guarantee continuity of the feedback at $x = 0$. If Conditions 1 and 2 hold then an almost smooth feedback is given by the so-called Sontag's universal formula:

$$\alpha(x) = \begin{cases} 0, & \text{if } \frac{\partial V}{\partial x} g(x) = 0, \\ -\frac{\frac{\partial V}{\partial x} f(x) + \sqrt{(\frac{\partial V}{\partial x} f(x))^2 + (\frac{\partial V}{\partial x} g(x))^4}}{\frac{\partial V}{\partial x} g(x)}, & \text{elsewhere.} \end{cases}$$

Constructive Stabilization

We now introduce two classes of nonlinear systems for which systematic design methods to solve the state feedback stabilization problem are available.

Feedback Systems

Consider a nonlinear system described by equations of the form

$$\dot{x}_1 = f_1(x_1, x_2), \quad \dot{x}_2 = u, \quad (8)$$

with $x_1(t) \in \mathbb{R}^n$, $x_2(t) \in \mathbb{R}$, $u(t) \in \mathbb{R}^n$ and $f_1(0, 0) = 0$. This system belongs to the so-called class of feedback systems for which a sort of reduction principle holds: the zero equilibrium of the system is smoothly stabilizable if the same holds for the *reduced* system $\dot{x}_1 = f_1(x_1, v)$, which is obtained from the first of equations (8) replacing the state variable x_2 with a *virtual control* input v . To show this property, suppose there exist a continuously differentiable function $\alpha_1 : \mathbb{R}^n \rightarrow \mathbb{R}$ and a continuously differentiable and radially unbounded function $V_1 : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $V_1(0) = 0$, $V_1(x_1) > 0$, for all $x_1 \neq 0$, and

$$\frac{\partial V_1}{\partial x_1} f_1(x_1, \alpha_1(x_1)) < 0,$$

for all $x_1 \neq 0$, i.e., the zero equilibrium of the system $\dot{x}_1 = f_1(x_1, v)$ is globally asymptotically stabilizable.

Consider now the function

$$V(x_1, x_2) = V_1(x_1) + \frac{1}{2} (x_2 - \alpha_1(x_1))^2,$$

which is radially unbounded and such that $V(0, 0) = 0$ and $V(x_1, x_2) > 0$ for all non-zero (x_1, x_2) , and note that

$$\begin{aligned} \dot{V} &= \frac{\partial V_1}{\partial x_1} f_1(x_1, x_2) \\ &\quad + (x_2 - \alpha_1(x_1))(u + \Delta_1(x_1, x_2)) \\ &= \frac{\partial V_1}{\partial x_1} f_1(x_1, \alpha_1(x_1)) \\ &\quad + (x_2 - \alpha_1(x_1))(u + \Delta_2(x_1, x_2)), \end{aligned}$$

for some continuously differentiable mappings Δ_1 and Δ_2 , which exist by Hadamard's Lemma. As a result, the feedback

$$\alpha(x_1, x_2) = -\Delta_2(x_1, x_2) - k(x_2 - \alpha_1(x_1)),$$

with $k > 0$, yields $\dot{V} < 0$ for all non-zero (x_1, x_2) ; hence the feedback is a continuously differentiable stabilizer for the zero equilibrium of the system (8). Note, finally, that the function V is a control Lyapunov for the system (8); hence Sontag's formula can be also used to construct a stabilizer.

The result discussed above is at the basis of the so-called *backstepping* technique for recursive stabilization of systems described, for example, by equations of the form

$$\begin{aligned} \dot{x}_1 &= x_2 + \varphi_1(x_1), \\ \dot{x}_2 &= x_3 + \varphi_2(x_1, x_2), \\ \dot{x}_3 &= x_4 + \varphi_3(x_1, x_2, x_3), \\ &\vdots \\ \dot{x}_n &= u + \varphi_n(x_1, \dots, x_n), \end{aligned}$$

with $x_i(t) \in \mathbb{R}$, for all $i \in [1, \dots, n]$, and φ_i smooth mappings such that $\varphi_i(0) = 0$, for all $i \in [1, \dots, n]$.

Feedforward Systems

Consider a nonlinear system described by equations of the form

$$\dot{x}_1 = f_1(x_2), \quad \dot{x}_2 = f_2(x_2) + g_2(x_2)u, \quad (9)$$

with $x_1(t) \in \mathbb{R}$, $x_2(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}$, $f_1(0) = 0$ and $f_2(0) = 0$. This system belongs to the so-called class of feedforward systems for which, similarly to feedback systems, a sort of reduction principle holds: the zero equilibrium of the system is smoothly stabilizable if the zero equilibrium of the *reduced* system $\dot{x}_2 = f_2(x_2)$ is globally asymptotically stable and some additional structural assumption holds. To show this property, suppose that there exists a continuously differentiable and radially unbounded function $V_2 : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $V_2(0) = 0$, $V_2(x_2) > 0$, for all $x_2 \neq 0$, and

$$\frac{\partial V_2}{\partial x_2} f_2(x_2) < 0,$$

for all $x_2 \neq 0$. Suppose, in addition, that there exists a continuously differentiable mapping $M(x_2)$ such that

$$f_1(x_2) - \frac{\partial M}{\partial x_2} f_2(x_2) = 0,$$

$M(0) = 0$ and $\left. \frac{\partial M}{\partial x_2} \right|_{x_2=0} \neq 0$. Existence of such a mapping is guaranteed, for example, by asymptotic stability of the linearization of the system $\dot{x}_2 = f_2(x_2)$ around the origin and controllability of the linearization of the system (9) around the origin.

Consider now the function

$$V(x_1, x_2) = \frac{1}{2} (x_1 - M(x_2))^2 + V_2(x_2),$$

which is radially unbounded and such that $V(0, 0) = 0$ and $V(x_1, x_2) > 0$ for all non-zero (x_1, x_2) , and note that

$$\begin{aligned} \dot{V} &= -(x_1 - M(x_2)) \frac{\partial M}{\partial x_2} g_2(x_2) u \\ &\quad + \frac{\partial V_2}{\partial x_2} f_2(x_2) + \frac{\partial V_2}{\partial x_2} g_2(x_2) u \\ &= \frac{\partial V_2}{\partial x_2} f_2(x_2) + \\ &\quad \left(\frac{\partial V_2}{\partial x_2} - (x_1 - M(x_2)) \frac{\partial M}{\partial x_2} \right) g_2(x_2) u. \end{aligned}$$

As a result, the feedback

$$\begin{aligned} \alpha(x_1, x_2) \\ = -k \left(\frac{\partial V_2}{\partial x_2} - (x_1 - M(x_2)) \frac{\partial M}{\partial x_2} \right) g_2(x_2), \end{aligned}$$

with $k > 0$, yields $\dot{V} < 0$ for all non-zero (x_1, x_2) ; hence the feedback is a continuously differentiable stabilizer for the zero equilibrium of the system (9). Note, finally, that the function V is a control Lyapunov for the system (9); hence

Sontag's formula can be also used to construct a stabilizer.

The result discussed above is at the basis of the so-called forwarding technique for recursive stabilization of systems described, for example, by equations of the form

$$\begin{aligned} \dot{x}_1 &= \varphi_1(x_2, \dots, x_n), \\ \dot{x}_2 &= \varphi_2(x_3, \dots, x_n), \\ &\vdots \\ \dot{x}_{n-1} &= \varphi_{n-1}(x_n), \\ \dot{x}_n &= u, \end{aligned}$$

with $x_i(t) \in \mathbb{R}$ for all $i \in [1, \dots, n]$, and φ_i smooth mappings such that $\varphi_i(0) = 0$, for all $i \in [1, \dots, n]$.

Stabilization via Output Feedback

In the previous sections, we have studied the stabilization problem for nonlinear systems under the assumption that the whole state is available for feedback. This requires the on-line measurement of the state vector x , which may pose a severe constraint in applications. This observation motivates the study of the much more challenging, but more realistic, problem of stabilization with partial state information. This problem requires the introduction of a specific notion of observability. Note that for nonlinear systems, it is possible to define several, nonequivalent, observability notions.

Similarly to section “Control Lyapunov Functions,” we focus on the class of systems affine in the control, i.e., systems described by equations of the form

$$\begin{aligned} \dot{x} &= f(x) + g(x)u, \\ y &= h(x), \end{aligned} \tag{10}$$

with $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$, and $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ smooth mappings. This is precisely the class of systems in equation (7) with the addition of the *output map* h , i.e., a map which describes the information that is available for feedback. In addition we assume, to simplify the notation, that

$m = 1$ and $p = 1$: the system is single-input, single-output.

To define the observability notion of interest, consider the sequence of mappings

$$\begin{aligned}\phi_0(x) &= h(x), \\ \phi_1(x, v_0) &= \frac{\partial \phi_0}{\partial x} [f(x) + g(x)v_0], \\ \phi_2(x, v_0, v_1) &= \frac{\partial \phi_1}{\partial x} [f(x) + g(x)v_0] \\ &\quad + \frac{\partial \phi_1}{\partial v_0} v_1, \\ &\dots \\ \phi_k(x, v_0, v_1, \dots, v_{k-1}) &= \frac{\partial \phi_{k-1}}{\partial x} [f(x) + g(x)v_0] \\ &\quad + \sum_{i=0}^{k-2} \frac{\partial \phi_{k-1}}{\partial v_i} v_{i+1},\end{aligned}\quad (11)$$

with $k \leq n$. Note that, if $u(t)$ is of class C^{k-1} then

$$y^{(k)}(t) = \phi_k(x(t), u(t), \dots, u^{(k-1)}(t)),$$

where the notation $y^{(k)}(t)$, with k positive integer, is used to denote the k -th order derivative of the function $y(t)$, provided it exists. The mappings ϕ_0 to ϕ_{n-1} can be collected into a unique mapping $\Phi : \mathbb{R}^n \times \mathbb{R}^{n-1} \rightarrow \mathbb{R}^n$ defined as

$$\Phi(x, v_0, v_1, \dots, v_{n-2}) = \begin{bmatrix} \phi_0(x) \\ \phi_1(x, v_0) \\ \vdots \\ \phi_{n-1}(x, v_0, v_1, \dots, v_{n-2}) \end{bmatrix}.$$

The mapping Φ is, by construction, such that

$$\begin{aligned}\Phi(x(t), u(t), \dot{u}(t), \dots, u^{(n-2)}(t)) \\ = [y(t) \ \dot{y}(t) \ \dots \ y^{(n-1)}(t)]^T,\end{aligned}$$

for any t in which the indicated signals exist. As a consequence, if the mapping Φ is such that, as

some point $(\bar{x}, \bar{v})$, where $\bar{v} = [\bar{v}_0 \ \bar{v}_1 \ \dots \ \bar{v}_{n-2}]^T$,

$$\text{rank} \frac{\partial \Phi}{\partial x}(\bar{x}, \bar{v}) = n, \quad (12)$$

then, by the Implicit Function Theorem, there exists locally around $(\bar{x}, \bar{v})$ a smooth mapping $\Psi : \mathbb{R}^n \times \mathbb{R}^{n-1} \rightarrow \mathbb{R}^n$ such that

$$\omega = \Phi(\Psi(\omega, v), v),$$

i.e., the mapping Ψ is the *inverse* of Φ , parameterized by v . We conclude the discussion noting that if, at a certain time $\bar{t}$, $\bar{x} = x(\bar{t})$ and $\bar{v} = [u(\bar{t}) \ \dot{u}(\bar{t}) \ \dots \ u^{(n-2)}(\bar{t})]^T$ are such that the rank condition (12) holds then the mapping Ψ can be used to reconstruct the state of the system from measurements of the input, and its derivatives, and the output and its derivatives, for all t in a neighborhood of $\bar{t}$: $x(t) = \Psi(\omega(t), v(t))$, for all t in a neighborhood of $\bar{t}$, where

$$\begin{aligned}v(t) &= [u(t) \ \dot{u}(t) \ \dots \ u^{(n-2)}(t)]^T, \\ \omega(t) &= [y(t) \ \dot{y}(t) \ \dots \ y^{(n-1)}(t)]^T.\end{aligned}\quad (13)$$

This property is a local property: to derive a property which allows a global reconstruction of the state, we need to impose additional conditions.

Definition 2 Consider the system (10) with $m = p = 1$. The system is said to be *uniformly observable* if the following conditions hold.

- (i) The mapping $H : \mathbb{R}^n \rightarrow \mathbb{R}^n$ defined as (the functions $L_f^i h$, with i nonnegative integer, are defined as $L_f h(x) = L_f^1 h(x) = \frac{\partial h}{\partial x} f(x)$ and, recursively, as $L_f^{i+1} h(x) = L_f(L_f^i h(x))$)

$$H(x) = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{bmatrix}$$

is a global diffeomorphism.

- (ii) The rank condition (12) holds for all $(x, v) \in \mathbb{R}^n \times \mathbb{R}^{n-1}$.

The notion of uniform observability allows *performing* a global reconstruction of the state, i.e., it makes sure that the identities

$$\omega = \Phi(\Psi(\omega, v), v), \quad x = \Psi(\Phi(x, v), v),$$

hold for all x, v , and ω . In principle this property may be used in an output feedback control architecture obtained implementing a stabilizing state feedback $u = \alpha(x)$ as $u = \alpha(\Psi(\omega, v))$, with v and ω as given in (13). This implementation is however not possible, since it gives an implicit definition of u and requires the exact differentiation of the input and output signals.

To circumvent these difficulties one needs to follow a somewhat longer path, as described hereafter. In addition, the global asymptotic stability requirement should be replaced by a less ambitious, yet practically meaningful, requirement: semi-global asymptotic stability. This requirement can be formalized as follows.

Definition 3 The equilibrium x_0 of the system (1), or (7), is said to be semi-globally asymptotically stabilizable if, for each compact set $\mathcal{K} \subset \mathbb{R}^n$ such that $x_0 \in \text{int } \mathcal{K}$, where $\text{int } \mathcal{K}$ is the set of all interior points of $\mathcal{K}$, there exists a feedback control law, possibly depending on $\mathcal{K}$, such that the equilibrium x_0 is a locally asymptotically stable equilibrium of the closed-loop system, and for any $x(0) \in \mathcal{K}$ one has $\lim_{t \rightarrow \infty} x(t) = x_0$.

To bypass the need for the derivatives of the input signal, consider the extended system

$$\begin{aligned} \dot{x} &= f(x) + g(x)v_0, & \dot{v}_0 &= v_1, \\ \dot{v}_1 &= v_2, & \dots & \dot{v}_{n-1} = \tilde{u}. \end{aligned} \quad (14)$$

Note that, as described in section the “[Feedback Systems](#),” if the equilibrium $x_0 = 0$ of the system $\dot{x} = f(x) + g(x)u$ is globally asymptotically stabilizable by a (smooth) feedback $u = \alpha(x)$, then there exists a smooth state feedback $\tilde{u} = \tilde{\alpha}(x, v_0, v_1, \dots, v_{n-1})$ which globally asymptotically stabilizes the zero equilibrium of the system (14). In the feedback $\tilde{\alpha}$ one can replace x

with $\psi(\omega, v)$, thus yielding a feedback of the measurable part of the state of the system (14) and of the output y and its derivatives. Note that if $\omega(t) = [y(t) \ \dot{y}(t) \ \dots \ y^{(n-1)}(t)]^T$ then the feedback $\tilde{u} = \tilde{\alpha}(\psi(\omega, v), v_0, v_1, \dots, v_{n-1})$ globally asymptotically stabilizes the zero equilibrium of the system (14).

To avoid the computation of the time derivatives of y we exploit the uniform observability property, which implies that the auxiliary system

$$\begin{aligned} \dot{\eta}_0 &= \eta_1, \\ \dot{\eta}_1 &= \eta_2, \\ &\vdots \\ \dot{\eta}_{n-1} &= \phi_n(\psi(\eta, v), v_0, v_1, \dots, v_{n-1}), \end{aligned} \quad (15)$$

with $\eta = [\eta_0 \ \eta_1 \ \dots \ \eta_{n-1}]^T$, has the property of reproducing $y(t)$ and its derivatives up to $y^{(n-1)}(t)$ if properly initialized. This initialization is not feasible, since it requires the knowledge of the derivative of the output at $t = 0$. Nevertheless, the auxiliary system (15) can be modified to provide an *estimate* of y and its derivatives. The modification is obtained adding a linear correction term yielding the system

$$\begin{aligned} \dot{\eta}_0 &= \eta_1 + L \ c_{n-1}(y - \eta_0), \\ \dot{\eta}_1 &= \eta_2 + L^2 \ c_{n-2}(y - \eta_0), \\ &\vdots \\ \dot{\eta}_{n-1} &= \phi_n(\psi(\eta, v), v_0, v_1, \dots, v_{n-1}) \\ &\quad + L^n \ c_0(y - \eta_0), \end{aligned} \quad (16)$$

with $L > 0$ and the coefficients $c_0, \dots, c_{n-1}$ such that all roots of the polynomial $\lambda^n + c_{n-1}\lambda^{n-1} + \dots + c_1\lambda + c_0$ are in C^- . The system (16) has the ability to provide asymptotic estimates of y and its derivatives up to $y^{(n-1)}$ provided these are bounded and the gain L is selected sufficiently large, i.e., the system (16) is a *semi-global* observer of y and its derivatives up to $y^{(n-1)}$.

The closed-loop system obtained using the feedback law $\tilde{u} = \tilde{\alpha}(\psi(\eta, v), v_0, v_1, \dots, v_{n-1})$ has a locally asymptotically stable equilibrium at

the origin. To achieve semi-global stability one has to select L sufficient large and replace ψ with

$$\tilde{\psi}(\eta, v) = \begin{cases} \psi(\eta, v), & \text{if } \|\psi(\eta, v)\| < M, \\ M \frac{\psi(\eta, v)}{\|\psi(\eta, v)\|}, & \text{if } \|\psi(\eta, v)\| \geq M, \end{cases}$$

with $M > 0$ to be selected, as detailed in the following statement.

Theorem 3 Consider the system (10) with $m = p = 1$. Let $x_0 = 0$ be an achievable equilibrium. Assume $h(0) = 0$. Suppose the system is uniformly observable and there exists a smooth state feedback control law $u = \alpha(x)$ which globally asymptotically stabilizes the zero equilibrium and it is such that $\alpha(0) = 0$.

Then for each $R > 0$ there exist $\tilde{R} > 0$ and $M^* > 0$, and for each $M > M^*$ there exists L^* such that, for each $M > M^*$ and $L > L^*$ the dynamic output feedback control law

$$\begin{aligned} \dot{v}_0 &= v_1, \\ \dot{v}_1 &= v_2, \\ &\vdots \\ \dot{v}_{n-1} &= \tilde{\alpha}(\tilde{\psi}(\eta, v), v_0, v_1, \dots, v_{n-1}) \\ \dot{\eta}_0 &= \eta_1 + L c_{n-1}(y - \eta_0), \\ \dot{\eta}_1 &= \eta_2 + L^2 c_{n-2}(y - \eta_0), \\ &\vdots \\ \dot{\eta}_{n-1} &= \phi_n(\psi(\eta, v), v_0, v_1, \dots, v_{n-1}) \\ &\quad + L^n c_0(y - \eta_0), \\ u &= v_0, \end{aligned}$$

yields a closed-loop system with the following properties.

- The zero equilibrium of the system is locally asymptotically stable.
- For any $x(0)$, $v(0)$, and $\eta(0)$ such that $\|x(0)\| < R$ and $\|(v(0), \eta(0))\| < \tilde{R}$ the

trajectories of the closed-loop system are such that

$$\begin{aligned} \lim_{t \rightarrow \infty} x(t) &= 0, \\ \lim_{t \rightarrow \infty} v(t) &= 0, \\ \lim_{t \rightarrow \infty} \eta(t) &= 0. \end{aligned}$$

The foregoing result can be informally formulated as follows: global state feedback stabilizability and uniform observability imply semi-global stabilizability by output feedback. This can be regarded as a nonlinear enhancement of the so-called separation principle for the stabilization, by output feedback, of linear systems. Note, finally, that a global version of the separation principle can be derived under one additional assumption: the existence of an estimator of the norm of the state x .

Summary and Future Directions

A necessary condition and a sufficient condition for stabilizability of an equilibrium of a nonlinear system have been given, together with two systematic design methods. The necessary condition allows to rule out, using a simple algebraic test, existence of continuous stabilizers, whereas the sufficient condition provides a link with classical Lyapunov theory. In addition, the problem of semi-global stability by dynamic output feedback has been discussed in detail. Several issues have not been discussed, including the use of discontinuous, hybrid and time-varying feedbacks; stabilization by static output feedback and dynamic state feedback; and robust stabilization. Note finally that similar considerations can be carried out for nonlinear discrete-time systems.

Cross-References

- [Feedback Linearization of Nonlinear Systems](#)
- [Regulation and Tracking of Nonlinear Systems](#)

Recommended Reading

Classical references on stabilization for nonlinear systems and on recent research directions are given below.

Bibliography

- Artstein Z (1983) Stabilization with relaxed controls. *Nonlinear Anal Theory Methods Appl* 7:1163–1173
- Astolfi A, Karagiannis D, Ortega R (2008) *Nonlinear and adaptive control with applications*. London: Springer
- Astolfi A, Praly L (2006) Global complete observability and output-to-state stability imply the existence of a globally convergent observer. *Math Control Signals Syst* 18:32–65
- Bacciotti A (1992) *Local stabilizability of nonlinear control systems*. World Scientific, Singapore
- Brockett RW (1983) Asymptotic stability and feedback stabilization. In: Brockett RW, Millman RS, Sussmann HJ (eds) *Differential geometry control theory*. Birkhauser, Boston, pp 181–191
- Gauthier J-P, Kupka I (2001) *Deterministic observation theory and applications*. Cambridge University Press, Cambridge
- Isidori A (1995) *Nonlinear control systems*, 3rd edn. Springer, Berlin
- Isidori A (1999) *Nonlinear control systems II*. Springer, London
- Jurdjevic V, Quinn JP (1978) Controllability and stability. *J Differ Equ* 28:381–389
- Khalil HK (2002) *Nonlinear systems*, 3rd edn. Prentice-Hall, Upper Saddle River
- Krstić M, Kanellakopoulos I, Kokotović P (1995) *Nonlinear and adaptive control design*. John Wiley and Sons, New York
- Marino R, Tomei P (1995) *Nonlinear control design: geometric, adaptive and robust*. Prentice-Hall, London
- Mazenc F, Praly L (1996) Adding integrations, saturated controls, and stabilization for feedforward systems. *IEEE Trans Autom Control* 41:1559–1578
- Mazenc F, Praly L, Dayawansa WP (1994) Global stabilization by output feedback: examples and counterexamples. *Syst Control Lett* 23:119–125
- Ortega R, Loria A, Nicklasson PJ, Sira-Ramírez H (1998) *Passivity-based control of Euler-Lagrange systems*. Springer, London
- Ryan EP (1994) On Brockett's condition for smooth stabilizability and its necessity in a context of nonsmooth feedback. *SIAM J Control Optim* 32:1597–1604
- Sepulchre R, Janković M, and Kokotović P (1996) *Constructive nonlinear control*. Springer, Berlin
- Sontag ED (1989) A “universal” construction of Artstein's theorem on nonlinear stabilization. *Syst Control Lett* 13:117–123

- Teel A, Praly L (1995) Tools for semiglobal stabilization by partial state and output feedback. *SIAM J Control Optim* 33(5):1443–1488
- van der Schaft A (2000) *L_2 -gain and passivity techniques in nonlinear control*, 2nd edn. Springer, London

Financial Markets Modeling

Monique Jeanblanc
Laboratoire de Mathématiques et Modélisation,
LaMME, IBGBI, Université Paris Saclay, Evry,
France

Abstract

Mathematical finance is an important part of applied mathematics since the 1980s. At the beginning, the main goal was to price derivative products and to provide hedging strategies. Nowadays, the goal is to provide models for prices and interest rates, such that better calibration of parameters can be done. In these pages, we present some basic models. Details can be found in Musiela M, Rutkowski M (*Martingale methods in financial modelling*. Springer, Berlin, 2005).

Keywords

Affine process · Brownian process · Default times · Lévy process · Wishart distribution

Models for Prices of Stocks

The first model of prices was elaborated by Louis Bachelier, in his thesis (1900). The idea was that the dynamic of prices has a trend, perturbed by a noise. For this noise, Bachelier set the fundamental properties of the Brownian motion. Bachelier's prices were of the form

$$S_t^B = S_0^B + \nu t + \sigma W_t$$

where W is a Brownian motion.

This model, where prices can take negative values, was changed by taking the exponential as in the celebrated Black-Scholes-Merton model (BSM) where

$$S_t = e^{S_t^B} = S_0 \exp(vt + \sigma W_t).$$

Only two constant parameters were needed; the coefficient σ is called the volatility. From Itô's formula, the dynamics of the BSM's price are

$$dS_t = S_t(\mu dt + \sigma dW_t)$$

where $\mu = v + \frac{1}{2}\sigma^2$. The interest rate is assumed to be a constant r .

The price of a derivative product of payoff $H \in \mathcal{F}_T = \sigma(S_s, s \leq T)$ is obtained using a hedging procedure. One proves that there exists a (self-financing) portfolio with value V , investing in the savings account and in the stock S , with terminal value H , i.e., $V_T = H$. The price of H at time t is V_t . Using that methodology, the price of a European call with strike K (i.e., for $H = (S_T - K)^+$) is

$$V_t = BS(\sigma, K)_t := S_t \mathcal{N}(d_1) - K e^{-r(T-t)} \mathcal{N}(d_2)$$

where $\mathcal{N}$ is the cumulative distribution function of a standard Gaussian law and

$$d_1 = \frac{1}{\sigma \sqrt{T-t}} \left(\ln \left(\frac{S_t}{K e^{-r(T-t)}} \right) \right) + \frac{1}{2} \sigma \sqrt{T-t}, \quad d_2 = d_1 - \sigma \sqrt{T-t}$$

Note that the coefficient μ plays no role in this pricing methodology. This formula opened the door to the notion of risk neutral probability measure: the price of the option (or of any derivative product) is the expectation under the unique probability measure $\mathbb{Q}$, equivalent to $\mathbb{P}$ such that the discounted price $S_t e^{-rt}$, $t \geq 0$ is a $\mathbb{Q}$ martingale.

During one decade, the financial market was quite smooth, and this simple model was efficient to price derivative products and to calibrate the coefficients from the data. After the Black Monday of October 1987, the model was recognized

to suffer some weakness, and the door was fully open to more sophisticated models, with more parameters. In particular, the smile effect was seen on the data: the BSM formula being true, the price of the option would be a deterministic function of the fixed parameter σ and of K (the other parameter as maturity, underlying price, interest rate being fixed), and one would obtain, using $\psi(\cdot, K)$, the inverse of $BS(\cdot, K)$, the constant $\sigma = \psi(C^o(K), K)$, where $C^o(K)$ is the observed prices associated with the strike K . This is not the case, the curve $K \rightarrow \psi(C^o(K), K)$ having a smile shape (or a smirk shape). The BSM model is still used as a benchmark in the concept of implied volatility: for a given observed option price $C^o(K, T)$ (with strike K and maturity T), one can find the value of σ^* such that $BS(\sigma^*, K, T) = C^o(K, T)$. The surface $\sigma^*(K, T)$ is called the implied volatility surface and plays an important role in calibration issues.

Due to the need of more accurate formula, many models were presented, the only (mathematical) restriction being that prices have to be semi-martingales (to avoid arbitrage opportunities).

A first class is the stochastic volatility models. Assuming that the diffusion coefficient (called the local volatility) is a deterministic function of time and underlying, i.e.,

$$dS_t = S_t(\mu dt + \sigma(t, S_t) dW_t)$$

Dupire proved, using Kolmogorov backward equation, that the function σ is determined by the observed prices of call options by

$$\frac{1}{2} K^2 \sigma^2(T, K) \tag{1}$$

$$= \frac{\partial_T C^o(K, T) + r K \partial_K C^o(K, T)}{\partial_{KK}^2 C^o(K, T)} \tag{2}$$

where ∂_T (resp. ∂_K) is the partial derivative operator with respect to the maturity (resp., the strike). However, this important model (which allows hedging for derivative products) does not allow a full calibration of the volatility surface.

A second class consists of assuming that the volatility is a stochastic process. The first example is the Heston model which assumes that

$$\begin{aligned} dS_t &= S_t(\mu dt + \sqrt{v_t} dW_t) \\ dv_t &= -\lambda(v_t - \bar{v})dt + \eta\sqrt{v_t}dB_t \end{aligned}$$

where B and W are two Brownian motions with correlation factor ρ . This model is generalized by Gouriéroux and Sufana (2003) as Wishart model where the risky asset S is a d dimensional process, with matrix of quadratic variation Σ satisfy

$$\begin{aligned} dS_t &= \text{Diag}(S_t)(\mu dt + \sqrt{\Sigma_t} dW_t) \\ d\Sigma_t &= (\Lambda\Lambda^T + M\Sigma_t + \Sigma_t M^T)dt \\ &\quad + \sqrt{\Sigma_t} dB_t Q + Q^T (dB_t)^T \sqrt{\Sigma_t} \end{aligned}$$

where W is a d dimensional Brownian motion and B a $(d \times d)$ matrix Brownian; Λ , M , Q are $(d \times d)$ matrices; M is semidefinite negative; and $\Lambda\Lambda^T = \beta QQ^T$ with $\beta \geq d - 1$ to ensure strict positivity. The efficiency of these models was checked using calibration methodology; however, the main idea of hedging for pricing issues is forgotten: there is, in general, no hedging strategy, even for common derivative products, and the validation of the model (the form of the volatility, since the drift term plays no role in pricing) is done by calibration. The risk neutral probability is no more unique.

A new generation of stochastic volatility models where the instantaneous volatility is driven by a (rough) fractional Brownian motion is now studied (see Gatheral et al. 2018).

Interest Rate Models

In the beginning of the 1970s, a specific attention was paid for the interest rate modeling, a constant interest rate being far from the real world.

A first class consists of a dynamic for the instantaneous interest rate r .

Vasisek suggested an Ornstein-Uhlenbeck diffusion, i.e., $dr_t = a(b - r_t)dt + \sigma dW_t$, the solution being a Gaussian process of the form

$$r_t = (r_0 - b)e^{-at} + b + \sigma \int_0^t e^{-a(t-u)} dW_u.$$

This model is fully tractable; one of its weakness is that the interest rate can take negative values.

Cox, Ingersoll, and Rubinstein (CIR) studied the square root process

$$dr_t = a(b - r_t)dt + \sigma\sqrt{r_t}dB_t,$$

where $ab > 0$ (so that r is nonnegative). No closed form for r is known; however, closed form for the price of the associated zero coupons is known (see below in Affine Processes).

A second class is the celebrated Heath-Jarrow-Morton (HJM) model: the starting point being to model the price of a zero coupon with maturity T (i.e., an asset which pays one monetary unit at time T , these prices being observable) in terms of the instantaneous forward rate $f(t, T)$, as

$$B(t, T) = \exp\left(-\int_t^T f(t, u)du\right).$$

Assuming that

$$df(t, T) = \alpha(t, T)dt + \sigma(t, T)dW_t$$

one finds

$$dB(t, T) = B(t, T)(a(t, T)dt + b(t, T)dW_t)$$

where the relationship between a , b and α , β is known. The instantaneous interest rate is $r_t = f(t, t)$. This model is quite efficient; however, no conditions are known in order that the interest rate is positive.

The new generation of term structure models is devoted to modeling multiple yield curves which have emerged after the last financial crisis. In particular, an HJM approach to model the term structure of multiplicative spreads between FRA rates and simply compounded OIS risk-free

forward rates is provided in Grbac and Runggaldier (2016).

Models with Jumps

Lévy Processes

Following the idea to produce tractable models to fit the data, many models are now based on Lévy's processes. These models are used for modeling prices as $S_t = e^{X_t}$ where X is a Lévy process. They present a nice feature: even if closed forms for pricing are not known, numerical methods are efficient. One of the most popular is the Carr-Geman-Madan-Yor (CGMY) model which is a Lévy process without Gaussian component and with Lévy density

$$\frac{C}{x^{Y+1}} e^{-Mx} \mathbb{1}_{\{x>0\}} + \frac{C}{|x|^{Y+1}} e^{Gx} \mathbb{1}_{\{x<0\}}$$

with $C > 0$, $M \geq 0$, $G \geq 0$, and $Y < 2$.

These models are often presented as a special case of a change of time: roughly speaking, any semi-martingale is a time-changed Brownian motion, and many examples are constructed as $S_t = B_{A_t}$, where B is a Brownian motion and A an increasing process (the change of time), chosen independent of B (for simplicity) and A a Lévy process (for computational issues).

Affine Processes

Affine models were introduced by Duffie et al. (2003) and are now a standard tool for modeling stock prices or interest rates. An affine process enjoys the property that, for any affine function g ,

$$\begin{aligned} \mathbb{E}(\exp(uX_T + \int_t^T g(X_s)ds) | \mathcal{F}_t) \\ = \exp(\alpha(t, T)X_t + \beta(t, T)) \end{aligned}$$

where α and β are deterministic solutions of PDEs.

A class of affine processes X ($\mathbb{R}^n$ -valued) is the one where

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t + dZ_t$$

where the drift vector b is an affine function of x , the covariance matrix $\sigma(x)\sigma^T(x)$ is an affine function of x , W is an n -dimensional Brownian motion, and Z is a pure jump process whose jumps have a given law ν on $\mathbb{R}^n$ and arrive with intensity $f(X_t)$ where f is an affine function. An example is the CIR model (without jumps), where one can find the price of a zero coupon as

$$\begin{aligned} \mathbb{E}(\exp(-\int_t^T r_s ds) | \mathcal{F}_t) \\ = \Phi(T-t) \exp[-r_t \Psi(T-t)] \end{aligned}$$

where

$$\begin{aligned} \Psi(s) &= \frac{2(e^{\gamma s} - 1)}{(\gamma + a)(e^{\gamma s} - 1) + 2\gamma}, \\ \Phi(s) &= \left(\frac{2\gamma e^{(\gamma+a)\frac{s}{2}}}{(\gamma + a)(e^{\gamma s} - 1) + 2\gamma} \right)^{\frac{2ab}{\sigma^2}}, \\ \gamma^2 &= a^2 + 2\sigma^2. \end{aligned}$$

Models for Defaults

At the end of the 1990s, new kinds of financial products appear on the market: defaultable options, defaultable zero coupons, and credit derivatives, as the CDOs. Then, particular attention was paid to the modeling of default times; see Bielecki and Rutkowski (2001). The most popular model for a single default is the reduced form approach, which is based on the knowledge of the intensity rate process. Given a nonnegative process λ , adapted with respect to a reference filtration $\mathbb{F}$, a random time τ is defined so that

$$\mathbb{P}(\tau > t | \mathcal{F}_\infty) = \exp(-\int_0^t \lambda_s ds)$$

This implies that $\mathbb{1}_{\tau \leq t} - \int_0^{t \wedge \tau} \lambda_u du$ is a martingale (in the smallest filtration $\mathbb{G}$ which contains $\mathbb{F}$

and makes τ a random time). This intensity rate λ plays the role of the interest spread due to the equality, for any $Y \in \mathcal{F}_T$ and $\mathbb{F}$ adapted interest rate r

$$\begin{aligned} \mathbb{E}(Y \mathbb{1}_{T < \tau} \exp(-\int_t^T r_s ds) | \mathcal{G}_t) \mathbb{1}_{t < \tau} \\ = \mathbb{1}_{t < \tau} \mathbb{E}(Y \exp(-\int_t^T (r_s + \lambda_s) ds) | \mathcal{F}_t) \end{aligned}$$

The real challenge is to model multi-defaults. Defaults are assumed to occur at times τ_i , and one has to describe the (conditional) joint law of the vector $\tau = (\tau_1, \tau_2, \dots, \tau_n)$, the number n of defaults being quite large (100). A first step is to study the law of the defaults, i.e.,

$$\mathbb{P}(\tau_1 > t_1, \dots, \tau_n > t_n)$$

which is performed using the classical copula approach, based on the knowledge of the marginal laws of the τ_i , i.e., on the knowledge of $\mathbb{P}(\tau_i \leq s) =: F_i(s)$ where the cumulative distribution functions F_i are assumed invertible. The simplest and most popular copula being the Gaussian one

$$\begin{aligned} \mathbb{P}(F_i^{-1}(\tau_i) \leq u_i, i = 1, \dots, n) \\ = \mathcal{N}_{\Sigma}^n \left(\mathcal{N}^{-1}(u_1), \dots, \mathcal{N}^{-1}(u_n) \right), \end{aligned}$$

where $\mathcal{N}_{\Sigma}^n$ is the c.d.f. for the n -variate central normal distribution with the linear correlation matrix Σ and $\mathcal{N}^{-1}$ is the inverse of the c.d.f. for the univariate standard normal distribution. However, a dynamical model was needed, mainly to study the contagion effect, i.e., how the occurrence of a default affect the probability of occurrence of the next defaults.

A first class of models is based on the intensity: starting from a given family of intensities $(\lambda_t^i, i = 1, \dots, n)$ which satisfies

$$d\lambda_t^i = f(t, \lambda_t^i)dt + g(t, \lambda_t^i)dW_t + dL_t^{(-i)}$$

where $L_t^{(-i)} = \sum_{k \neq i} \beta_k \mathbb{1}_{\tau_k \leq t}$, one constructs random times having the given intensities.

Another class of models, which allows for common jumps, is based on Markov Chains with absorbing state, the default time of the i -th entity being the time where the i -th component of the Markov Chain enters in the absorbing state. These models are efficient due to the introduction of common factors.

Many studies are done to find a form of a dynamic copula, i.e., a family of processes $G_t(\theta), t \geq 0$ such that

$$G_t(\theta) = \mathbb{P}(\tau_1 > \theta_1, \dots, \tau_n > \theta_n | \mathcal{F}_t)$$

Some examples where, for any fixed t , the quantity $G_t(\cdot)$ is a (stochastic) Gaussian law are known; however, there are few concrete examples for other cases.

Cross-References

- [Credit Risk Modeling](#)
- [Option Games: The Interface Between Optimal Stopping and Game Theory](#)
- [Stock Trading via Feedback Control Methods](#)

Bibliography

- Bachelier L (1900) Théorie de la Spéculation, Thèse, Annales Scientifiques de l'Ecole Normale Supérieure, pp 21–86, III-17
- Bielecki TR, Rutkowski M (2001) Credit risk: modelling valuation and hedging. Springer, Berlin
- Duffie D, Filipović D, Schachermayer W (2003) Affine processes and applications in finance. Ann Appl Probab 13:984–1053 (2003)
- Gatheral J, Jaisson T, Rosenbaum M (2018) Volatility is rough. Quant Finan 18(6):933–949
- Gourieroux C, Sufana R (2003) Wishart quadratic term structure, CREF 03–10, HEC Montreal
- Grbac Z, Runggaldier WJ (2016) Interest rate modeling: post-crisis challenges and approaches. Springer briefs in quantitative finance. Springer
- Musiela M, Rutkowski M (2005) Martingale methods in financial modelling. Springer, Berlin

Finite-Horizon Linear-Quadratic Optimal Control with General Boundary Conditions

Augusto Ferrante¹ and Lorenzo

Ntogramatzidis²

¹Dipartimento di Ingegneria dell'Informazione,
Università di Padova, Padova, Italy

²Department of Mathematics and Statistics,
Curtin University, Perth, WA, Australia

Abstract

The linear-quadratic (LQ) problem is the prototype of a large number of optimal control problems, including the fixed endpoint, the point-to-point, and several H_2/H_∞ control problems, as well as the dual counterparts. In the past 50 years, these problems have been addressed using different techniques, each tailored to their specific structure. It is only in the last 10 years that it was recognized that a unifying framework is available. This framework hinges on formulae that parameterize the solutions of the Hamiltonian differential equation in the continuous-time case and the solutions of the extended symplectic system in the discrete-time case. Whereas traditional techniques involve the solutions of Riccati differential or difference equations, the formulae used here to solve the finite-horizon LQ control problem only rely on solutions of the algebraic Riccati equations. In this entry, aspects of the framework are described within a discrete-time context.

Keywords

Cyclic boundary conditions · Discrete-time linear systems · Fixed end-point · Initial value · Point-to-point boundary conditions · Quadratic cost · Riccati equations

Introduction

Ever since the *linear-quadratic* (LQ) optimal control problem was introduced in the 1960s by Kalman in his pioneering paper (1960), it has found countless applications in areas such as chemical process control, aeronautics, robotics, servomechanisms, and motor control, to name but a few.

For details on the *raisons d'être* of LQ problems, readers are referred to the classical textbooks on this topic (Kwakernaak and Sivan 1972; Anderson and Moore 1971) and to the Special Issue on LQ optimal control problems in *IEEE Trans. Aut. Contr.*, vol. AC-16, no. 6, 1971. The LQ regulator is not only important per se. It is also the prototype of a variety of fundamental optimization problems. Indeed, several optimal control problems that are extremely relevant in practice can be recast into composite LQ, dual LQ, or generalized LQ problems. Examples include LQG, H_2 and H_∞ problems, and Kalman filtering problems. Moreover, LQ optimal control is intimately related, via matrix Riccati equations, to absolute stability, dissipative networks, and optimal filtering. The importance of LQ problems is not restricted to linear systems. For example, LQ control techniques can be used to modify an optimal control law in response to perturbations in the dynamics of a nonlinear plant. For these reasons, the LQ problem is universally regarded as a cornerstone of modern control theory.

In its simplest and most classical version, the finite-horizon discrete LQ optimal control can be stated as follows:

Problem 1 Let $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, and consider the linear system

$$x_{t+1} = Ax_t + Bu_t, \quad y_t = Cx_t + Du_t, \quad (1)$$

where the initial state $x_0 \in \mathbb{R}^n$ is given. Let $W = W^\top \in \mathbb{R}^{n \times n}$ be positive semidefinite. Find a sequence of inputs u_t , with $t = 0, 1, \dots, N-1$, minimizing the cost function

$$J_{N,x_0}(u) \stackrel{\text{def}}{=} \sum_{t=0}^{N-1} \|y_t\|^2 + x_N^\top W x_N. \quad (2)$$

Historically, LQ problems were first introduced and solved by Kalman in (1960). In this paper, Kalman showed that the LQ problem can be solved for *any* initial state x_0 , and the optimal control can be written as a state feedback $u(t) = K(t)x(t)$, where $K(t)$ can be found by solving a famous quadratic matrix difference equation known as the *Riccati equation*. When W is no longer assumed to be positive semidefinite, the optimal solution may or may not exist. A complete analysis of this case has been worked out in Bilardi and Ferrante (2007). In the infinite-horizon case (i.e., when N is infinite), the optimal control (when it exists) is stationary and may be computed by solving an *algebraic* Riccati equation (Kwakernaak and Sivan 1972; Anderson and Moore 1971).

Since its introduction, the original formulation of the classic LQ optimal control problem has been generalized in several different directions, to accommodate for the need of considering more general scenarios than the one represented by Problem 1. Examples include the so-called fixed endpoint LQ, in which the extreme states are sharply assigned, and the *point-to-point* case, in which the initial and terminal values of an output of the system are constrained to be equal to specified values. This led to a number of contributions in the area where different adaptations of the Riccati theory were tailored to these diversified contexts of LQ optimal control. These variations of the classic LQ problem are becoming increasingly important due to their use in several applications of interest. Indeed, many applications including spacecraft, aircraft, and chemical processes involve maneuvering between two states during some phases of a typical mission. Another interesting example is the H_2 -optimization of transients in switching plants, where the problem can be divided into a set of finite-horizon LQ problems with

welding conditions on the optimal arcs for each switch instant. This problem has been the object of a large number of contributions in the recent literature, under different names: Parameter varying systems, jump linear systems, switching systems, and bumpless systems are definitions extensively used to denote different classes of systems affected by sensible changes in their parameters or structures (Balas and Bokor 2004). In recent years, a new unified approach emerged in Ferrante et al. (2005), Ferrante and Ntogramatzidis (2005, 2007a, b) that solves the finite-horizon LQ optimal control problem via a formula which parameterizes the set of trajectories generated by the corresponding Hamiltonian differential equation in the continuous time and the extended symplectic difference equation in the discrete case. Loosely, we can say that the expressions parameterizing the trajectories of the Hamiltonian differential equation and the extended symplectic difference equation using this approach hinge on a pair of “opposite” solutions of the associated algebraic Riccati equations. This active stream of research considerably enlarged the range of optimal control problems that can be successfully addressed. This point of view always requires some controllability-type assumption and the extended symplectic pencil (Ferrante and Ntogramatzidis 2005, 2007b) to be regular and devoid of generalized eigenvalues on the unit circle. More recently, a new point of view has emerged which yields a more direct solution to this problem, without requiring system-theoretic assumptions (Ntogramatzidis and Ferrante 2013; Ferrante and Ntogramatzidis 2013a, b).

The discussion here is restricted to the discrete-time case; for the corresponding continuous-time counterpart, we will only make some comments and refer to the literature.

Notation For the reader’s convenience, we briefly review some, mostly standard, matrix notation used throughout the paper. Given a matrix $B \in \mathbb{R}^{n \times m}$, we denote by $B^\top$ its *transpose*

and by $B^\dagger$ its *Moore-Penrose pseudo-inverse*, the unique matrix $B^\dagger$ that satisfies $BB^\dagger B = B$, $B^\dagger BB^\dagger = B^\dagger$, $(BB^\dagger)^\top = BB^\dagger$, and $(B^\dagger B)^\top = B^\dagger B$. The *kernel* of B is the subspace $\{x \in \mathbb{R}^n \mid Bx = 0\}$ and is denoted $\ker B$. The *image* of B is the subspace $\{y \in \mathbb{R}^m \mid \exists x \in \mathbb{R}^n : y = Ax\}$ and is denoted by $\text{im } B$. Given a square matrix A , we denote by $\sigma(A)$ its *spectrum*, i.e., the set of its eigenvalues. We write $A_1 > A_2$ (resp. $A_1 \geq A_2$) when $A_1 - A_2$ is *positive definite* (resp. *positive semidefinite*).

Classical Finite-Horizon Linear-Quadratic Optimal Control

The simplest classical version of the finite-horizon LQ optimal control is Problem 1. By employing some standard linear algebra, this problem may be solved by the classical technique known as “completion of squares”: First of all, the cost can be rewritten as

$$J_{N,x_0}(u) = \sum_{t=0}^{N-1} [x_t^\top \ u_t^\top] \begin{bmatrix} Q + A^\top X_{t+1} A - X_t & S + A^\top X_{t+1} B \\ S^\top + B^\top X_{t+1} A & R + B^\top X_{t+1} B \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + x_N^\top (W - X_N) x_N + x_0^\top X_0 x_0, \quad (6)$$

which holds for any sequence of matrices X_t . With $X_N \stackrel{\text{def}}{=} W$ fixed, for $t = N-1, N-2, \dots, 0$, let

$$X_t \stackrel{\text{def}}{=} Q + A^\top X_{t+1} A - (S + A^\top X_{t+1} B) (R + B^\top X_{t+1} B)^\dagger (S^\top + B^\top X_{t+1} A). \quad (7)$$

It is now easy to see that all the matrices of the sequence X_t defined above are positive semidefinite. Indeed, $X_N = W \geq 0$. Assume by induction that $X_{t+1} \geq 0$. Then,

$$M_{t+1} \stackrel{\text{def}}{=} \begin{bmatrix} Q + A^\top X_{t+1} A & S + A^\top X_{t+1} B \\ S^\top + B^\top X_{t+1} A & R + B^\top X_{t+1} B \end{bmatrix} = \Pi + \begin{bmatrix} A^\top \\ B^\top \end{bmatrix} X_{t+1} \begin{bmatrix} A & B \end{bmatrix} \geq 0.$$

$$J_{N,x_0}(u) = \sum_{t=0}^{N-1} [x_t^\top \ u_t^\top] \Pi \begin{bmatrix} x_t \\ u_t \end{bmatrix} + x_N^\top W x_N, \quad (3)$$

with

$$\Pi \stackrel{\text{def}}{=} \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} C^\top \\ D^\top \end{bmatrix} [C \ D] = \Pi^\top \geq 0. \quad (4)$$

Now, let $X_0, X_1, \dots, X_N$ be an arbitrary sequence of $n \times n$ symmetric matrices. We have the identity

$$\left(\sum_{t=0}^{N-1} [x_{t+1}^\top X_{t+1} x_{t+1} - x_t^\top X_t x_t] \right) + x_0^\top X_0 x_0 - x_N^\top X_N x_N = 0. \quad (5)$$

Adding (5)–(3) and using the expression (1) for x_{t+1} , we get

Since X_t is the generalized Schur complement of the right upper block of M_{t+1} in M_{t+1} , it follows that $X_t \geq 0$, which in turns implies

$$R + B^\top X_t B \geq 0.$$

Moreover, by employing Eq.(7) and recalling that, given a positive semidefinite matrix $\Pi_0 = \begin{bmatrix} Q_0 & S_0 \\ S_0^\top & R_0 \end{bmatrix} = \Pi_0^\top \geq 0$, we have $\begin{bmatrix} S_0 & R_0^\dagger S_0^\top & S_0 \\ & S_0^\top & R_0 \end{bmatrix} = \begin{bmatrix} S_0 \\ R_0 \end{bmatrix} R_0^\dagger \begin{bmatrix} S_0^\top & R_0 \end{bmatrix} \geq 0$, we easily see that

$$M_{t+1} - \begin{bmatrix} X_t & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} S + A^\top X_{t+1} B \\ R + B^\top X_{t+1} B \end{bmatrix} (R + B^\top X_t B)^\dagger \begin{bmatrix} S^\top + A^\top X_{t+1} B & R + B^\top X_{t+1} B \end{bmatrix} \geq 0.$$

Hence, (6) takes the form

$$J_{N,x_0}(u) = \sum_{t=0}^{N-1} \left\| [(R + B^\top X_{t+1} B)^\dagger]^{1/2} \right. \\ \times [(S^\top + B^\top X_{t+1} A)x_t \\ \left. + (R + B^\top X_{t+1} B)u_t \right\|_2^2 \\ + x_0^\top X_0 x_0. \quad (8)$$

Now it is clear that u_t is optimal if and only if

$$[(R + B^\top X_{t+1} B)^\dagger]^{1/2} [(S^\top + B^\top X_{t+1} A)x_t \\ + (R + B^\top X_{t+1} B)u_t] = 0,$$

whose solutions are parameterized by the feedback control

$$u_t = -K_t x_t + G_t v_t, \quad (9)$$

where $K_t \stackrel{\text{def}}{=} (R + B^\top X_{t+1} B)^\dagger (S^\top + B^\top X_{t+1} A)$ and $G_t \stackrel{\text{def}}{=} [I - (R + B^\top X_{t+1} B)^\dagger (R + B^\top X_{t+1} B)]$ is the orthogonal projector onto the linear space of vectors that can be added to the optimal control u_t without affecting optimality and v_t is a free parameter. The optimal state trajectory is now given by the closed-loop dynamics

$$x_{t+1} = (A - B K_t)x_t + B G_t v_t. \quad (10)$$

The optimal cost is clearly

$$J^* = x_0^\top X_0 x_0. \quad (11)$$

The corresponding results in continuous time can be obtained along the same lines as the discrete-time case; see Ferrante and Ntogramatzidis (2013b) and references therein.

More General Linear-Quadratic Problems

The problem discussed in the previous section presents some limitations that prevent its applicability in several important situations. In

particular, three relevant generalizations of the classical problem are:

1. *The fixed endpoint case*, where the states at the endpoints x_0 and x_N are both assigned.
2. *The point-to-point case*, where the initial and terminal values z_0 and z_N of linear combination $z_t = V x_t$ of the state of the dynamical system described by (1) are constrained to be equal to two assigned vectors.
3. *The cyclic case*, where the states at the endpoints x_0 and x_N are not sharply assigned, but they are constrained to be equal (clearly, we can have combinations of (2) and (3)).

All these problems are special cases of a general LQ problem that can be stated as follows:

Problem 2 Consider the dynamical setting (1) of Problem 1. Find a sequence of inputs u_t , with $t = 0, 1, \dots, N-1$ and an initial state x_0 minimizing the cost function

$$J_{N,x_0}(u) \stackrel{\text{def}}{=} \sum_{t=0}^{N-1} \|y_t\|^2 + [x_0^\top - \bar{x}_0^\top \quad x_N^\top - \bar{x}_N^\top] \\ \times \begin{bmatrix} W_{11} & W_{12} \\ W_{12}^\top & W_{22} \end{bmatrix} \begin{bmatrix} x_0 - \bar{x}_0 \\ x_N - \bar{x}_N \end{bmatrix} \quad (12)$$

under the dynamic constraints (1) and the endpoints constraints

$$V \begin{bmatrix} x_0 \\ x_N \end{bmatrix} = v. \quad (13)$$

Here $W \stackrel{\text{def}}{=} \begin{bmatrix} W_{11} & W_{12} \\ W_{12}^\top & W_{22} \end{bmatrix}$ is a positive semidefinite matrix (partitioned in four blocks) that quadratically penalizes the differences between the initial state x_0 and a desired initial state $\bar{x}_0$ and between the final state x_N and a desired final state $\bar{x}_N$ (this general problem formulation can also encompass problems where the difference $x_0 - x_N$ is not fixed but has to be quadratically penalized with a matrix $\Delta = \Delta^\top \geq 0$ in the performance index $J'_{N,x_0}(u) = \sum_{t=0}^{N-1} [x_t^\top \quad u_t^\top] \Pi \begin{bmatrix} x_t \\ u_t \end{bmatrix} + (x_0 - x_N)^\top \Delta (x_0 - x_N)$). It is simple to see that this

performance index can be brought back to (12) by setting $W = \begin{bmatrix} \Delta & -\Delta \\ -\Delta & \Delta \end{bmatrix}$ and $\bar{x}_0 = \bar{x}_N = 0$. Equation (13) permits to impose also a hard constraint on an arbitrary linear combination of initial and final states.

The solution of Problem 2 can be obtained by parameterizing the solutions of the so-called extended symplectic system (see Ferrante and Levy (1998) and references therein for a discussion on symplectic matrices and pencils). This solution can be convenient also for the classical case of Problem 1. In fact it does not require to iterate the difference Riccati equation (which can be undesirable if the time horizon is large) but only to solve an algebraic Riccati equation and a related discrete Lyapunov equation.

This solution requires some definitions, preliminary results, and standing assumptions (see Ntogramatzidis and Ferrante (2013) and Ferrante and Ntogramatzidis (2013a) for a more general approach which does not require such assumptions). A detailed proof of the main result can be found in Ferrante and Ntogramatzidis (2007b); see also Zattoni (2008) and Ferrante and Ntogramatzidis (2012). The extended symplectic pencil is defined by

$$\begin{aligned} zF - G, \quad F &\stackrel{\text{def}}{=} \begin{bmatrix} I_n & 0 & 0 \\ 0 & -A^\top & 0 \\ 0 & -B^\top & 0 \end{bmatrix}, \\ G &\stackrel{\text{def}}{=} \begin{bmatrix} A & 0 & B \\ Q & -I_n & S \\ S^\top & 0 & R \end{bmatrix}, \end{aligned} \quad (14)$$

where Q, S, R are defined as in (3). We make the following assumptions:

- (A1) The pair (A, B) is *modulus controllable*, i.e., $\forall \lambda \in \mathbb{C} \setminus \{0\}$ at least one of the two matrices $[\lambda I - A \mid B]$ and $[\lambda^{-1} I - A \mid B]$ has full row rank.
- (A2) The pencil $zF - G$ is regular (i.e., $\det(zF - G)$ is not the zero polynomial) and has no generalized eigenvalues on the unit circle.

Consider the discrete algebraic Riccati equation

$$\begin{aligned} X &= A^\top X A - (A^\top X B + S)(R + B^\top X B)^{-1} \\ &\quad (S^\top + B^\top X A) + Q. \end{aligned} \quad (15)$$

Under assumptions (A1)–(A2), (15) admits a strongly unmixed solution $X = X^\top$, i.e., a solution X for which the corresponding closed-loop matrix

$$\begin{aligned} A_X &\stackrel{\text{def}}{=} A - B K_X, \quad \text{with} \\ K_X &\stackrel{\text{def}}{=} (R + B^\top X B)^{-1}(S^\top + B^\top X A), \end{aligned} \quad (16)$$

has spectrum that does not contain reciprocal values, i.e., $\lambda \in \sigma(A_X)$ implies $\lambda^{-1} \notin \sigma(A_X)$. It can now be proven that the following closed-loop Lyapunov equation admits a unique solution $Y = Y^\top \in \mathbb{R}^{n \times n}$:

$$A_X Y A_X^\top - Y + B(R + B^\top X B)^{-1} B^\top = 0. \quad (17)$$

The following theorem provides an explicit formula parameterizing all the optimal state and control trajectories for Problem 2. Notice that this formula can be readily implemented starting from the problem data.

Theorem 1 *With reference to Problem 2, assume that (A1) and (A2) are satisfied. Let $X = X^\top$ be any strongly unmixed solution of (15) and $Y = Y^\top$ be the corresponding solution of (17). Let N_V be a basis matrix (in the case when $\ker V = \{0\}$, we consider N_V to be void) of the null space of V . Moreover, let*

$$\begin{aligned} F &\stackrel{\text{def}}{=} A_X^N, \\ K_\star &\stackrel{\text{def}}{=} K_X Y A_X^\top - (R + B^\top X B)^{-1} B^\top, \\ \hat{X} &\stackrel{\text{def}}{=} \text{diag}(-X, X), \quad \bar{x} \stackrel{\text{def}}{=} \begin{bmatrix} \bar{x}_0 \\ \bar{x}_N \end{bmatrix}, \\ w &\stackrel{\text{def}}{=} \begin{bmatrix} v \\ -N_V^\top W \bar{x} \end{bmatrix}, \quad L \stackrel{\text{def}}{=} \begin{bmatrix} I_n & Y F^\top \\ F & Y \end{bmatrix}, \\ U &\stackrel{\text{def}}{=} \begin{bmatrix} 0 & -F^\top \\ 0 & I_n \end{bmatrix}, \end{aligned}$$

$$M \stackrel{\text{def}}{=} \begin{bmatrix} V L \\ N_V^\top [(\hat{X} - W) L - U] \end{bmatrix}.$$

$$\mathcal{P} \stackrel{\text{def}}{=} \{\pi = M^\dagger w + N_M \zeta \mid \zeta \text{ arbitrary}\}. \quad (18)$$

Problem 2 admits solutions if and only if $w \in \text{im } M$. In this case, let N_M be a basis matrix of the null space of M , and define

Then, the set of optimal state and control trajectories of Problem 2 is parameterized in terms of $\pi \in \mathcal{P}$, by

$$\begin{bmatrix} x(t) \\ u(t) \end{bmatrix} = \begin{cases} \begin{bmatrix} A_X^t & Y (A_X^\top)^{N-t} \\ -K_X A_X^t & -K_\star (A_X^\top)^{N-t-1} \end{bmatrix} \pi, & 0 \leq t \leq N-1, \\ \begin{bmatrix} A_X^N & Y \\ 0 & 0 \end{bmatrix} \pi, & t = N. \end{cases} \quad (19)$$

F

The interpretation of the above result is the following. As π varies, (19) describes the trajectories of the extended symplectic system. The set $\mathcal{P}$ defined in (18) is the set of π for which these trajectories satisfy the boundary conditions. All the details of this construction can be found in Ferrante and Ntogramatzidis (2007b). If the pair (A, B) is stabilizable, we can choose $X = X^\top$ to be the stabilizing solution of (15). In such case, the matrices A_X^t , $(A_X^\top)^{N-t}$ and $(A_X^\top)^{N-t-1}$ appearing in (19) are asymptotically stable for all $t = 0, \dots, N$. Thus, in this case, the optimal state trajectory and control are expressed in terms of powers of strictly stable matrices in the overall time interval, thus ensuring the robustness of the obtained solution even for very large time horizons. Indeed, the stabilizing solution of an algebraic Riccati equation and the solution of a Lyapunov equation may be computed by standard and robust algorithms available in any control package (see the MATLAB[®] routines `dare.m` and `dlyap.m`). We refer to Ferrante et al. (2005) and Ferrante and Ntogramatzidis (2013b) for the continuous-time counterpart of the above results.

Summary

With the technique discussed in this entry, a large number of LQ problems can be tackled in a unified framework. Moreover, several finite-horizon LQ problems that can be interesting and

useful in practice can be recovered as particular cases of the control problem considered here. The generality of the optimal control problem herein considered is crucial in the solution of several $H_2 - H_\infty$ optimization problems whose optimal trajectory is composed of a set of arches, each one solving a parametric LQ subproblem in a specific time horizon and all joined together at the end-points of each subinterval. In these cases, in fact, a very general form of constraint on the extreme states is essential in order to express the condition of conjunction of each pair of subsequent arches at the endpoints.

Cross-References

- [H₂ Optimal Control](#)
- [H-Infinity Control](#)
- [Linear Quadratic Optimal Control](#)

Bibliography

- Anderson BDO, Moore JB (1971) Linear optimal control. Prentice Hall International, Englewood Cliffs
- Balas G, Bokor J (2004) Detection filter design for LPV systems – a geometric approach. *Automatica* 40: 511–518
- Bilardi G, Ferrante A (2007) The role of terminal cost/reward in finite-horizon discrete-time LQ optimal control. *Linear Algebra Appl (Spec Issue honor Paul Fuhrmann)* 425:323–344
- Ferrante A (2004) On the structure of the solution of discrete-time algebraic Riccati equation with singular

- closed-loop matrix. *IEEE Trans Autom Control* AC-49(11):2049–2054
- Ferrante A, Levy B (1998) Canonical form for symplectic matrix pencils. *Linear Algebra Appl* 274:259–300
- Ferrante A, Ntogramatzidis L (2005) Employing the algebraic Riccati equation for a parametrization of the solutions of the finite-horizon LQ problem: the discrete-time case. *Syst Control Lett* 54(7):693–703
- Ferrante A, Ntogramatzidis L (2007a) A unified approach to the finite-horizon linear quadratic optimal control problem. *Eur J Control* 13(5):473–488
- Ferrante A, Ntogramatzidis L (2007b) A unified approach to finite-horizon generalized LQ optimal control problems for discrete-time systems. *Linear Algebra Appl (Spec Issue honor Paul Fuhrmann)* 425(2–3):242–260
- Ferrante A, Ntogramatzidis L (2012) Comments on “structural invariant subspaces of singular Hamiltonian systems and nonrecursive solutions of finite-horizon optimal control problems”. *IEEE Trans Autom Control* 57(1):270–272
- Ferrante A, Ntogramatzidis L (2013a) The extended symplectic pencil and the finite-horizon LQ problem with two-sided boundary conditions. *IEEE Trans Autom Control* 58(8):2102–2107
- Ferrante A, Ntogramatzidis L (2013b) The role of the generalised continuous algebraic Riccati equation in impulse-free continuous-time singular LQ optimal control. In: *Proceedings of the 52nd conference on decision and control (CDC 13)*, Florence, 10–13 Dec 2013b
- Ferrante A, Marro G, Ntogramatzidis L (2005) A parametrization of the solutions of the finite-horizon LQ problem with general cost and boundary conditions. *Automatica* 41:1359–1366
- Kalman RE (1960) Contributions to the theory of optimal control. *Bulletin de la Sociedad Matematica Mexicana* 5:102–119
- Kwakernaak H, Sivan R (1972) *Linear optimal control systems*. Wiley, New York
- Ntogramatzidis L, Ferrante A (2010) On the solution of the Riccati differential equation arising from the LQ optimal control problem. *Syst Control Lett* 59(2):114–121
- Ntogramatzidis L, Ferrante A (2013) The generalised discrete algebraic Riccati equation in linear-quadratic optimal control. *Automatica* 49:471–478. <https://doi.org/10.1016/j.automatica.2012.11.006>
- Ntogramatzidis L, Marro G (2005) A parametrization of the solutions of the Hamiltonian system for stabilizable pairs. *Int J Control* 78(7):530–533
- Prattichizzo D, Ntogramatzidis L, Marro G (2008) A new approach to the cheap LQ regulator exploiting the geometric properties of the Hamiltonian system. *Automatica* 44:2834–2839
- Zattoni E (2008) Structural invariant subspaces of singular Hamiltonian systems and nonrecursive solutions of finite-horizon optimal control problems. *IEEE Trans Autom Control* AC-53(5):1279–1284

Flexible Robots

Alessandro De Luca

Sapienza Università di Roma, Roma, Italy

Abstract

Mechanical flexibility in robot manipulators is due to compliance at the joints and/or distributed deflection of the links. Dynamic models of the two classes of robots with flexible joints or flexible links are presented, together with control laws addressing the motion tasks of regulation to constant equilibrium states and of asymptotic tracking of output trajectories. Control design for robots with flexible joints takes advantage of the passivity and feedback linearization properties. In robots with flexible links, basic differences arise when controlling the motion at the joint level or at the tip level.

Keywords

Feedback linearization · Gravity compensation · Joint elasticity · Link flexibility · Noncausal and stable inversion · Regulation by motor feedback · Singular perturbation · Vibration damping

Introduction

Robot manipulators are usually considered as rigid multi-body mechanical systems. This ideal assumption simplifies dynamic analysis and control design but may lead to performance degradation and even unstable behavior, due to the excitation of vibrational phenomena.

Flexibility is mainly due to the limited stiffness of transmissions at the joints (Sweet and Good 1985) and to the deflection of slender and lightweight links (Cannon and Schmitz 1984). Joint flexibility is common when motion transmission/reduction elements such as belts, long shafts, cables, harmonic drives, or cycloidal gears

are used. Link flexibility is present in large articulated structures, such as very long arms needed for accessing hostile environments (deep sea or space) or automated crane devices for building construction. In both situations, static displacements and dynamic oscillations are introduced between the driving actuators and the actual position of the robot end effector. Undesired vibrations are typically confined beyond the closed-loop control bandwidth, but flexibility cannot be neglected when large speed/acceleration and high accuracy are requested by the task.

In the dynamic modeling, flexibility is assumed concentrated at the robot joints or distributed along the robot links (most of the times with some finite-dimensional approximation). In both cases, additional generalized coordinates are introduced beside those used to describe the rigid motion of the arm in a Lagrangian formulation. As a result, the number of available control inputs is strictly less than the number of degrees of freedom of the mechanical system. This type of under-actuation, though counterbalanced by the presence of additional potential energy helping to achieve system controllability, suggests that the design of satisfactory motion control laws is harder than in the rigid case.

From a control point of view, different design approaches are needed because of structural differences arising between flexible-joint and flexible-link robots. These differences hold for single- or multiple-link robots, in the linear or nonlinear domain, and depend on the physical co-location or not of mechanical flexibility versus control actuation, as well as on the choice of controlled outputs.

In order to measure the state of flexible robots for trajectory tracking control or feedback stabilization purposes, a large variety of sensing devices can be used, including encoders, joint torque sensors, strain gauges, accelerometers, and high-speed cameras. In particular, measuring the full state of the system would require twice the number of sensors than in the rigid case for robots with flexible joints and possibly more for robots with flexible links. The design of

controllers that work provably good with a reduced set of measurements is thus particularly attractive.

Robots with Flexible Joints

Dynamic Modeling

A robot with flexible joints is modeled as an open kinematic chain of $n + 1$ rigid bodies, interconnected by n joints undergoing deflection, and actuated by n electrical motors. Let θ be the n -vector of motor (i.e., rotor) positions, as reflected through the reduction gears, and q the n -vector of link positions. The joint deflection is $\delta = \theta - q \neq 0$. The standard assumptions are:

- A1** Joint deflections δ are small, limited to the domain of linear elasticity. The elastic torques due to joint deformations are $\tau_J = K(\theta - q)$, where K is the positive definite, diagonal joint stiffness matrix.
- A2** The rotors of the electrical motors are modeled as uniform bodies having their center of mass on the rotation axis.
- A3** The angular velocity of the rotors is due only to their own spinning.

The last assumption, introduced by Spong (1987), is very reasonable for large reduction ratios and also crucial for simplifying the dynamic model.

From the gravity and elastic potential energy, $\mathcal{U} = \mathcal{U}_g + \mathcal{U}_\delta$, and the kinetic energy $\mathcal{T}$ of the robot, applying the Euler-Lagrange equations to the Lagrangian $\mathcal{L} = \mathcal{T} - \mathcal{U}$ and neglecting all dissipative effects leads to the dynamic model

$$M(q)\ddot{q} + n(q, \dot{q}) + K(q - \theta) = 0 \quad (1)$$

$$B\ddot{\theta} + K(\theta - q) = \tau, \quad (2)$$

where $M(q)$ is the positive definite, symmetric inertia matrix of the robot links (including the motor masses); $n(q, \dot{q})$ is the sum of Coriolis and centrifugal terms $c(q, \dot{q})$ (quadratic in $\dot{q}$) and gravitational terms $g(q) = (\partial \mathcal{U}_g / \partial q)^T$; B is

the positive definite, diagonal matrix of motor inertias (reflected through the gear ratios); and τ are the motor torques (performing work on θ). The inertia matrix of the *complete* system is then $\mathcal{M}(q) = \text{block diag}\{M(q), B\}$. The two n -dimensional second-order differential equations (1) and (2) are referred to as the *link* and the *motor* equations, respectively. When the joint stiffness $K \rightarrow \infty$, it is $\theta \rightarrow q$ and $\tau_J \rightarrow \tau$, so that the two equations collapse in the limit into the standard dynamic model of rigid robots with total inertia $\mathcal{M}(q) = M(q) + B$. On the other hand, when the joint stiffness K is relatively large but still finite, robots with elastic joints show a *two-time-scale* dynamic behavior. A common large scalar factor $1/\epsilon^2 \gg 1$ can be extracted from the diagonal stiffness matrix as $K = \hat{K}/\epsilon^2$. The *slow* subsystem is associated to the link dynamics

$$M(q)\ddot{q} + n(q, \dot{q}) = \tau_J, \quad (3)$$

while the *fast* subsystem takes the form

$$\epsilon^2 \ddot{\tau}_J = \hat{K} \left(B^{-1} (\tau - \tau_J) + M^{-1}(q) (n(q, \dot{q}) - \tau_J) \right) \quad (4)$$

For small ϵ , Eqs. (3) and (4) represent a singularly perturbed system. The two separate time scales governing the slow and fast dynamics are t and $\sigma = t/\epsilon$.

Regulation

The basic robotic task of moving between two arbitrary equilibrium configurations is realized by a feedback control law that asymptotically stabilizes the desired robot state.

In the absence of gravity ($g \equiv 0$), the equilibrium states are parameterized by the desired reference position q_d of the links and take the form $q = q_d$, $\theta = \theta_d = q_d$ (with no joint deflection at steady state) and $\dot{q} = \dot{\theta} = 0$. As a result of passivity of the mapping from τ to $\dot{\theta}$, global regulation is achieved by a *decentralized PD* law using only feedback from the motor variables,

$$\tau = K_P(\theta_d - \theta) - K_D\dot{\theta}, \quad (5)$$

with diagonal $K_P > 0$ and $K_D > 0$.

In the presence of gravity, the (unique) equilibrium position of the motor associated with a desired link position q_d becomes $\theta_d = q_d + K^{-1}g(q_d)$. Global regulation is obtained by adding an extra gravity-dependent term τ_g to the PD control law (5),

$$\tau = K_P(\theta_d - \theta) - K_D\dot{\theta} + \tau_g, \quad (6)$$

with diagonal matrices $K_P > 0$ (at least) and $K_D > 0$. The term τ_g needs to match the gravity load $g(q_d)$ at steady state. The following choices are of slight increasing control complexity, with progressively better transient performance.

- *Constant* gravity compensation: $\tau_g = g(q_d)$. Global regulation is achieved when the smallest positive gain in the diagonal matrix K_P is large enough (Tomei 1991). This sufficient condition can be enforced only if the joint stiffness K dominates the gradient of gravity terms.
- *Online* gravity compensation: $\tau_g = g(\tilde{\theta})$, $\tilde{\theta} = \theta - K^{-1}g(q_d)$. Gravity effects on the links are approximately compensated during robot motion. Global regulation is proven under the same conditions above (De Luca et al. 2005).
- *Quasi-static* gravity compensation: $\tau_g = g(\tilde{q}(\theta))$. At any measured motor position θ , the link position $\tilde{q}(\theta)$ is computed by solving numerically $g(q) + K(q - \theta) = 0$. This removes the need of a strictly positive lower bound on K_P (Kugi et al. 2008), but the joint stiffness should still dominate the gradient of gravity terms.
- *Exact* gravity cancelation: $\tau_g = g(q) + BK^{-1}\dot{g}(q)$. This law requires full-state measurements (through the presence of $\ddot{q}$ in the term $\dot{g}$) and achieves perfect cancelation of the gravity term in the link dynamics, through a non-collocated torque at the motor side (De Luca and Flacco 2011). The robot links behave as if there were no gravity acting on the system, whereas the motor dynamics

is indeed still influenced. When completing the design with (5), no strictly positive lower bounds are needed for global asymptotic stability, i.e., $K_P > 0$ and $K > 0$ are now sufficient.

Additional feedback from the full robot state $(q, \dot{q}, \theta, \dot{\theta})$, measured or reconstructed through dynamic observers, can provide faster and damped transient responses. This solution is particularly convenient when a joint torque sensor measuring τ_J is available (*torque-controlled* robots). Using

$$\tau = K_P(\theta_d - \theta) - K_D\dot{\theta} + K_T(g(q_d) - \tau_J) - K_S\dot{\tau}_J + g(q_d), \quad (7)$$

the four diagonal gain matrices can be given a special structure so that asymptotic stability is automatically guaranteed (Albu-Schäffer and Hirzinger 2001).

Trajectory Tracking

Let a desired sufficiently smooth trajectory $q_d(t)$ be specified for the robot links over a finite or infinite time interval. The control objective is to asymptotically stabilize the trajectory tracking error $e = q_d(t) - q(t)$ to zero, starting from a generic initial robot state. Assuming that $q_d(t)$ is four times continuously differentiable, a torque input profile $\tau_d(t) = \tau_d(q_d, \dot{q}_d, \ddot{q}_d, \dddot{q}_d, \ddot{q}_d)$ can be derived from the dynamic model (1) and (2) so as to reproduce exactly the desired trajectory, when starting from matched initial conditions. A local solution to the trajectory tracking problem is provided by the combination of such feedforward term $\tau_d(t)$ with a stabilizing linear feedback from the partial or full robot state; see Eqs. (6) or (7).

When the joint stiffness is large enough, one can take advantage of the system being *singularly perturbed*. A control law τ_s designed for the rigid robot will deal with the slow dynamics, while a relatively simple action τ_f is used to stabilize the fast vibratory dynamics around an invariant manifold associated to the rigid robot control (Spong et al. 1987). This class of composite control laws

has the general form

$$\tau = \tau_s(q, \dot{q}, t) + \epsilon \tau_f(q, \dot{q}, \tau_J, \dot{\tau}_J). \quad (8)$$

When setting $\epsilon = 0$ in Eqs. (3), (4), and (8), the control setup of the equivalent rigid robot is recovered as

$$(M(q) + B)\ddot{q} + n(q, \dot{q}) = \tau_s. \quad (9)$$

Though more complex, the best performing trajectory tracking controller for the general case is based on *feedback linearization*. Spong (1987) has shown that the nonlinear state feedback

$$\tau = \alpha(q, \dot{q}, \ddot{q}, \dddot{q}) + \beta(q)v, \quad (10)$$

with

$$\begin{aligned} \alpha &= M(q)\ddot{q} + n(q, \dot{q}) \\ &\quad + BK^{-1}((\ddot{M}(q) + K)\ddot{q} + 2\dot{M}(q)\ddot{q} \\ &\quad + \ddot{n}(q, \dot{q})) \\ \beta &= BK^{-1}M(q), \end{aligned}$$

leads globally to the closed-loop linear system

$$q^{[4]} = v, \quad (11)$$

i.e., to decoupled chains of four input-output integrators from each auxiliary input v_i to each link position output q_i , for $i = 1, \dots, n$. The control design is then completed on the linear SISO side, by forcing the trajectory tracking error to be *exponentially stable* with an arbitrary decaying rate. The control law (10) is expressed as a function of the *linearizing* coordinates $(q, \dot{q}, \ddot{q}, \ddot{q})$ (up to the link jerk), which can be however rewritten in terms of the original state $(q, \dot{q}, \theta, \dot{\theta})$ using the dynamic model equations. This fundamental result is the direct extension of the so-called “computed torque” method for rigid robots.

Robots with Flexible Links

Dynamic Modeling

For the dynamic modeling of a single flexible link, the distributed nature of structural flexibility can be captured, under suitable assumptions, by partial differential equations (PDE) with associated boundary conditions. A common model is the *Euler-Bernoulli* beam. The link is assumed to be a slender beam, with uniform geometric characteristics and homogeneous mass distribution, clamped at the base to the rigid hub of an actuator producing a torque τ and rotating on a horizontal plane. The beam is flexible in the lateral direction only, being stiff with respect to axial forces, torsion, and bending due to gravity. Deformations are small and are in the elastic domain. The physical parameters of interest are the linear density ρ of the beam, its flexural rigidity EI , the beam length ℓ , and the hub inertia I_h (with $I_t = I_h + \rho\ell^3/3$). The equations of motion combine lumped and distributed parameter parts, with the hub rotation $\theta(t)$ and the link deformation $w(x, t)$, being $x \in [0, \ell]$ the position along the link. From Hamilton principle, we obtain

$$I_t \ddot{\theta}(t) + \rho \int_0^\ell x \ddot{w}(x, t) dx = \tau(t) \quad (12)$$

$$EI w''''(x, t) + \rho \ddot{w}(x, t) + \rho x \ddot{\theta}(t) = 0 \quad (13)$$

$$\begin{aligned} w(0, t) &= w'(0, t) = 0, \\ w''(\ell, t) &= w'''(\ell, t) = 0, \end{aligned} \quad (14)$$

where a prime denotes partial derivative w.r.t. to space. Equation (14) are the clamped-free boundary conditions at the two ends of the beam (no payload is present at the tip).

For the analysis of this self-adjoint PDE problem, one proceeds by separation of variables in space and time, defining

$$w(x, t) = \phi(x)\delta(t) \quad \theta(t) = \alpha(t) + k\delta(t), \quad (15)$$

where $\phi(x)$ is the link spatial deformation, $\delta(t)$ is its time behavior, $\alpha(t)$ describes the angular motion of the instantaneous center of mass of the beam, and k is chosen so as to satisfy (12) for $\tau = 0$. Being system (12), (13), and (14) linear, nonrational transfer functions can be derived in the Laplace transform domain between the input torque and some relevant system output, e.g., the angular position of the hub or of the tip of the beam (Kano 1990). The PDE formalism provides also a convenient basis for analyzing distributed sensing, feedback from strain sensors (Luo 1993), or even distributed actuation with piezo-electric devices placed along the link.

The transcendental characteristic equation associated to the spatial part of the solution to Eqs. (12), (13), and (14) is

$$\begin{aligned} I_h \gamma^3 (1 + \cos(\gamma\ell) \cosh(\gamma\ell)) \\ + \rho (\sin(\gamma\ell) \cosh(\gamma\ell) - \cos(\gamma\ell) \sinh(\gamma\ell)) = 0. \end{aligned} \quad (16)$$

When the hub inertia $I_h \rightarrow \infty$, the second term can be neglected and the characteristic equation collapses into the so-called *clamped* condition. Equation (16) has an infinite but countable number of positive real roots γ_i , with associated eigenvalues of resonant frequencies $\omega_i = \gamma_i^2 \sqrt{EI/\rho}$ and orthonormal eigenvectors $\phi_i(x)$, which are the natural deformation shapes of the beam (Barbieri and Özgüner 1988). A finite-dimensional dynamic model is obtained by truncation to a finite number m_e of eigenvalues/shapes. From

$$w(x, t) = \sum_{i=1}^{m_e} \phi_i(x) \delta_i(t) \quad (17)$$

we get

$$\begin{aligned} I_t \ddot{\alpha}(t) &= \tau(t) \\ \ddot{\delta}_i(t) + \omega_i^2 \delta_i(t) &= \phi_i'(0) \tau(t), \\ i &= 1, \dots, m_e, \end{aligned} \quad (18)$$

where the rigid body motion (top equation) appears as decoupled from the flexible dynamics,

thanks to the choice of variable α rather than θ . Modal damping can be added on the left-hand sides of the lower equations through terms $2\zeta_i\omega_i\dot{\delta}_i$ with $\zeta_i \in [0, 1]$. The angular position of the motor hub at the joint is given by

$$\theta(t) = \alpha(t) + \sum_{i=1}^{m_e} \phi'_i(0)\delta_i(t), \quad (19)$$

while the tip angular position is

$$y(t) = \alpha(t) + \sum_{i=1}^{m_e} \frac{\phi_i(\ell)}{\ell} \delta_i(t). \quad (20)$$

The *joint-level* transfer function $p_{\text{joint}}(s) = \theta(s)/\tau(s)$ will always have relative degree two and only minimum phase zeros. On the other hand, the *tip-level* transfer function $p_{\text{tip}}(s) = y(s)/\tau(s)$ will contain non-minimum phase zeros. This basic difference in the pattern of the transmission zeros is crucial for motion control design.

In a simpler modeling technique, a specified class of spatial functions $\phi_i(x)$ is assumed for describing link deformation. The functions need to satisfy only a reduced set of geometric boundary conditions (e.g., *clamped* modes at the link base), but otherwise no dynamic equations of motion such as (13). The use of finite-dimensional expansions like (17) limits the validity of the resulting model to a maximum frequency. This truncation must be accompanied by suitable filtering of measurements and of control commands, so as to avoid or limit spillover effects (Balas 1978).

In the dynamic modeling of robots with n flexible links, the resort to *assumed modes* of link deformation becomes unavoidable. In practice, some form of approximation and a finite-dimensional treatment is necessary. Let θ be the n -vector of joint variables describing the rigid motion and δ be the m -vector collecting the deformation variables of all flexible links. Following a Lagrangian formulation, the dynamic model with clamped modes takes the general form (Book 1984)

$$\begin{pmatrix} \mathbf{M}_{\theta\theta}(\theta, \delta) & \mathbf{M}_{\theta\delta}(\theta, \delta) \\ \mathbf{M}_{\theta\delta}^T(\theta, \delta) & \mathbf{M}_{\delta\delta}(\theta, \delta) \end{pmatrix} \begin{pmatrix} \ddot{\theta} \\ \ddot{\delta} \end{pmatrix} + \begin{pmatrix} \mathbf{n}_\theta(\theta, \delta, \dot{\theta}, \dot{\delta}) \\ \mathbf{n}_\delta(\theta, \delta, \dot{\theta}, \dot{\delta}) \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{D}\dot{\delta} + \mathbf{K}\delta \end{pmatrix} = \begin{pmatrix} \boldsymbol{\tau} \\ \mathbf{0} \end{pmatrix}, \quad (21)$$

where the positive definite, symmetric inertia matrix $\mathcal{M}$ of the complete robot and the Coriolis, centrifugal, and gravitational terms $\mathbf{n}$ have been partitioned in blocks of suitable dimensions, $\mathbf{K} > 0$ and $\mathbf{D} \geq 0$ are the robot link stiffness and damping matrices, and $\boldsymbol{\tau}$ is the n -vector of actuating torques.

The dynamic model (21) shows the general couplings existing between nonlinear rigid body motion and linear flexible dynamics. In this respect, the linear model (18) of a single flexible link is a remarkable exception.

The choice of specific assumed modes may simplify the blocks of the robot inertia matrix, e.g., orthonormal modes used for each link induce a decoupled structure of the diagonal inertia subblocks of $\mathbf{M}_{\delta\delta}$. Quite often the total kinetic energy of the flexible robot is evaluated only in the undeformed configuration $\delta = \mathbf{0}$. With this approximation, the inertia matrix becomes independent of δ , and so the velocity terms in the model. Furthermore, due to the hypothesis of small deformation of each link, the dependence of the gravity term in the lower component $\mathbf{n}_\delta$ is only a function of θ .

The validation of (21) goes through the experimental identification of the relevant dynamic parameters. Besides those inherited from the rigid case (mass, inertia, etc.), also the set of structural resonant frequencies and associated deformation profiles should be identified.

Control of Joint-Level Motion

When the target variables to be controlled are defined at the joint level, the control problem for robots with flexible links is similar to that of robots with flexible joints. As a matter of fact, the models (1), (2), and (21) are both *passive* systems with respect to the output θ ; see (19)

in the scalar case. For instance, regulation is achieved by a PD action with constant gravity compensation, using a control law of the form (6) without the need of feeding back link deformation variables (De Luca and Siciliano 1993a). Similarly, stable tracking of a joint trajectory $\theta_d(t)$ is obtained by a singular perturbation control approach, with flexible modes dynamics acting at multiple time scales with respect to rigid body motion (Siciliano and Book 1988), or by an inversion-based control (De Luca and Siciliano 1993b), where input-output (rather than full state) exact linearization is realized and the effects of link flexibility are canceled on the motion of the robot joints. While vibrational behavior will still affect the robot at the level of end-effector motion, the closed-loop dynamics of the δ variables is stable, and link deformations converge to a steady-state constant value (zero in the absence of gravity), thanks to the intrinsic damping of the mechanical structure. Improved transients are indeed obtained by active modal damping control (Cannon and Schmitz 1984).

A control approach specifically developed for the rest-to-rest motion of flexible mechanical systems is *command shaping* (Singer and Seering 1990). The original command designed to achieve a desired motion for a rigid robot is convolved with suitable signals delayed in time, so as to cancel (or reduce to a minimum) the effects of the excited vibration modes at the time of motion completion. For a single slewing link with linear dynamics, as in (18), the rest-to-rest input command is computed in closed form by using impulsive signals and can be made robust via an over-parameterization.

Control of Tip-Level Motion

The design of a control law that allows asymptotic tracking of a desired trajectory for the end effector of a robot with flexible links needs to face the unstable zero dynamics associated to the problem. In the linear case of a single flexible link, this is equivalent to the presence of non-minimum phase zeros in the transfer function to the tip output (20). Direct inversion of the input-output map leads to instability, due to cancelation of non-minimum phase zeros by unstable poles,

with link deformation growing unbounded and control saturations.

The solution requires instead to determine the unique reference state trajectory of the flexible structure that is associated to the desired tip trajectory and has *bounded* deformation. Based on regulation theory, the control law will be the superposition of a nominal feedforward action, which keeps the system along the reference state trajectory (and thus the output on the desired trajectory), and of a stabilizing feedback that reduces the error with respect to this state trajectory to zero without resorting to dangerous cancelations.

In general, computing such a control law requires the solution of a set of nonlinear partial differential equations. However, in the case of a single flexible link with linear dynamics, the feedforward profile is simply derived by an inversion defined in the *frequency domain* (Bayo 1987). The desired tip acceleration $\ddot{y}_d(t)$, $t \in [0, T]$, is considered as part of a rest-to-rest periodic signal, with zero mean value and zero integral. The procedure, implemented efficiently using fast Fourier transform on discrete-time samples, will automatically generate bounded time signals only. The resulting unique torque profile $\tau_d(t)$ will be a *noncausal* command, anticipating the actual start of the output trajectory at $t = 0$ (so as to *precharge* the link to the correct initial deformation) and ending after $t = T$ (to *discharge* the residual link deformation and recover the final rest configuration).

The same result was recovered by Kwon and Book (1994) in the time domain, by forward integrating in time the stable part of the inverse system dynamics and backward integrating the unstable part. An extension to the multi-link nonlinear case uses an iterative approach on repeated linear approximations of the system along the nominal trajectory (Bayo et al. 1989).

Summary and Future Directions

The presence of mechanical flexibility in the joints and the links of multi-dof robots poses challenging control problems. Control designs

take advantage or are limited by some system-level properties. Robots with flexible joints are passive systems at the level of motor outputs, have no zero dynamics associated to the link position outputs, and are always feedback linearizable systems. Robots with flexible links are still passive for joint-level outputs, but cannot be feedback linearized in general, and have unstable zero dynamics (non-minimum phase zeros in the linear case) when considering the end-effector position as controlled output.

State-of-the-art control laws address regulation and trajectory tracking tasks in a satisfactory way, at least in nominal conditions and under full-state feedback. Current research directions are aimed at achieving robustness to model uncertainties and external disturbances (with adaptive, learning, or iterative schemes) and further exploit the design of control laws under limited measurements and noisy sensing. Beyond free motion tasks, an accurate treatment of interaction tasks with the environment, requiring force or impedance controllers, is still missing for flexible robots. In this respect, passivity-based control approaches that do not necessarily operate dynamic cancelations may take advantage of the existing compliance, trading off between improved energy efficiency and some reduction in nominal performance.

Another avenue of research is to exploit the natural behavior of flexible robots, modifying by feedback the minimum amount of system dynamics needed in order to achieve the control target. In a sense, this approach is placed at the crossroad between model-based feedback linearization and passivity-based control. A first example is the gravity cancelation method of De Luca and Flacco (2010), which makes a robot having joints with (constant or variable) flexibility that moves under gravity feedback equivalent to the same robot without gravity. Such an *elastic structure preserving* concept has been expanded by introducing a number of similar control schemes, e.g., for link damping injection or trajectory tracking (Keppler et al. 2018). Recent results along this line have been obtained also for a class of soft continuous robot with distributed flexibility. Model-based controllers for tracking trajectories

in free space and for following contact surfaces can be designed so as to preserve the natural softness of the robot body and adapt to interactions with an unstructured environment (Della Santina et al. 2018).

Often seen as a limiting factor for performance, the presence of elasticity is now becoming an explicit advantage for safe physical human-robot interaction and for locomotion. The next generation of lightweight robots and humanoids will make use of flexible joints and also of a compact actuation with online controlled variable joint stiffness, an area of active research.

Cross-References

- ▶ [Feedback Linearization of Nonlinear Systems](#)
- ▶ [Modeling of Dynamic Systems from First Principles](#)
- ▶ [Nonlinear Zero Dynamics](#)
- ▶ [PID Control](#)
- ▶ [Regulation and Tracking of Nonlinear Systems](#)

Recommended Reading

In addition to the works cited in the body of this entry, a detailed treatment of dynamic modeling and control issues for flexible robots can be found in De Luca and Book (2016). This includes also the use of dynamic feedback linearization for a more general model of robots with elastic joints. For the same class of robots, Brogliato et al. (1995) provided a comparison of passivity-based and inversion-based tracking controllers.

Bibliography

- Albu-Schäffer A, Hirzinger G (2001) A globally stable state feedback controller for flexible joint robots. *Adv Robot* 15(8):799–814
- Balas MJ (1978) Feedback control of flexible systems. *IEEE Trans Autom Control* 23(4):673–679
- Barbieri E, Özgüner Ü (1988) Unconstrained and constrained mode expansions for a flexible slewing link. *ASME J Dyn Syst Meas Control* 110(4):416–421

- Bayo E (1987) A finite-element approach to control the end-point motion of a single-link flexible robot. *J Robot Syst* 4(1):63–75
- Bayo E, Papadopoulos P, Stubbe J, Serna MA (1989) Inverse dynamics and kinematics of multi-link elastic robots: an iterative frequency domain approach. *Int J Robot Res* 8(6):49–62
- Book WJ (1984) Recursive Lagrangian dynamics of flexible manipulators. *Int J Robot Res* 3(3):87–106
- Brogliato B, Ortega R, Lozano R (1995) Global tracking controllers for flexible-joint manipulators: a comparative study. *Automatica* 31(7):941–956
- Cannon RH, Schmitz E (1984) Initial experiments on the end-point control of a flexible one-link robot. *Int J Robot Res* 3(3):62–75
- De Luca A, Book W (2016) Robots with flexible elements. In: Siciliano B, Khatib O (eds) *Springer handbook of robotics*, 2nd edn. Springer, Berlin, pp 243–282
- De Luca A, Flacco F (2010) Dynamic gravity cancellation in robots with flexible transmissions. In: *Proceedings of 49th IEEE Conference on Decision and Control*, pp 288–295
- De Luca A, Flacco F (2011) A PD-type regulator with exact gravity cancellation for robots with flexible joints. In: *Proceedings of IEEE International Conference on Robotics and Automation*, pp 317–323
- De Luca A, Siciliano B (1993a) Regulation of flexible arms under gravity. *IEEE Trans Robot Autom* 9(4):463–467
- De Luca A, Siciliano B (1993b) Inversion-based nonlinear control of robot arms with flexible links. *AIAA J Guid Control Dyn* 16(6):1169–1176
- De Luca A, Siciliano B, Zollo L (2005) PD control with on-line gravity compensation for robots with elastic joints: theory and experiments. *Automatica* 41(10):1809–1819
- Della Santina C, Katzschmann R, Bicchi A, Rus D (2018) Dynamic control of a soft robots interacting with the environment. In: *Proceedings of IEEE International Conference on Soft Robotics*, pp 46–53
- Kanoh H (1990) Distributed parameter models of flexible robot arms. *Adv Robot* 5(1):87–99
- Keppeler M, Lakatos D, Ott C, Albu-Schäffer A (2018) Elastic structure preserving (ESP) control for compliantly actuated robots. *IEEE Trans Robot* 34(2):317–335
- Kugi A, Ott C, Albu-Schäffer A, Hirzinger G (2008) On the passivity-based impedance control of flexible joint robots. *IEEE Trans Robot* 24(2):416–429
- Kwon D-S, Book WJ (1994) A time-domain inverse dynamic tracking control of a single-link flexible manipulator. *ASME J Dyn Syst Meas Control* 116(2):193–200
- Luo ZH (1993) Direct strain feedback control of flexible robot arms: new theoretical and experimental results. *IEEE Trans Autom Control* 38(11):1610–1622
- Siciliano B, Book WJ (1988) A singular perturbation approach to control of lightweight flexible manipulators. *Int J Robot Res* 7(4):79–90
- Singer N, Seering WP (1990) Preshaping command inputs to reduce system vibration. *ASME J Dyn Syst Meas Control* 112(1):76–82
- Spong MW (1987) Modeling and control of elastic joint robots. *ASME J Dyn Syst Meas Control* 109(4):310–319
- Spong MW, Khorasani K, Kokotovic PV (1987) An integral manifold approach to the feedback control of flexible joint robots. *IEEE J Robot Autom* 3(4):291–300
- Sweet LM, Good MC (1985) Redefinition of the robot motion control problem. *IEEE Control Syst Mag* 5(3):18–24
- Tomei P (1991) A simple PD controller for robots with elastic joints. *IEEE Trans Autom Control* 36(10):1208–1213

Flocking in Networked Systems

Ali Jadbabaie

University of Pennsylvania, Philadelphia, PA, USA

Abstract

Flocking is a collective behavior exhibited by many animal species such as birds, insects, and fish. Such behavior is generated by distributed motion coordination through nearest-neighbor interactions. Empirical study of such behavior has been an active research in ecology and evolutionary biology. Mathematical study of such behaviors has become an active research area in a diverse set of disciplines, ranging from statistical physics and computer graphics to control theory, robotics, opinion dynamics in social networks, and general theory of multiagent systems. While models vary in detail, they are all based on local diffusive dynamics that results in emergence of consensus in direction of motion. Flocking is closely related to the notion of consensus and synchronization in multiagent systems, as examples of collective phenomena that emerge in multiagent systems as result of local nearest-neighbor interactions.

Keywords

Consensus · Dynamics · Flocking · Graph theory · Markov chains · Switched dynamical systems · Synchronization

Flocking or social aggregation is a group behavior observed in many animal species, ranging from various types of birds to insects and fish. The phenomena can be loosely defined as any aggregate collective behavior in parallel rectilinear formation or (in case of fish) in collective circular motion. The mechanisms leading to such behavior have been (and continues to be) an active area of research among ecologists and evolutionary biologists, dating back to the 1950s if not earlier. The engineering interest in the topic is much more recent.

In 1986, Craig Reynolds (1987), a computer graphics researcher, developed a computer model of collective behavior for animated artificial objects called boids. The flocking model for boids was used to realistically duplicate aggregation phenomena in fish flocks and bird schools for computer animation. Reynolds developed a simple, intuitive physics-based model: each boid was a point mass subject to three simple steering forces: alignment (to steer each boid towards the average heading of its *local* flockmates), cohesion (steering to move towards the average position of *local* flockmates), and separation (to avoid crowding local flockmates). The term *local* should be understood as those flockmates who are within each other's influence zone, which could be a disk (or a wide-angle sector of a disk) centered at each boid with a prespecified radius. This simple zone-based model created very realistic flocking behaviors and was used in many animations (Fig. 1). Reynolds' 3 rules of flocking.

Nine years later, in 1995, Vicsek et al. (1995) and coauthors independently developed a model for velocity alignment of self-propelled particles (SPPs) in a square with periodic boundary conditions. SPPs are essentially kinematic particles moving with constant speed and the steering law determines the angle of (what control

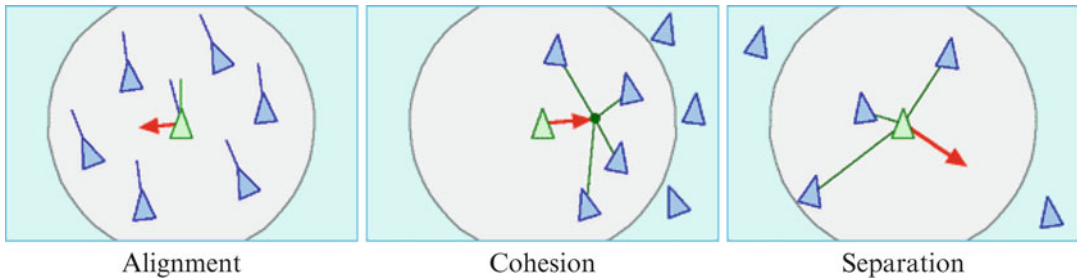
theorists call a kinematic, nonholonomic vehicle model). Vicsek et al.'s steering law was very intuitive and simple and was essentially Reynolds' zone-based alignment rule (Vicsek was not aware of Reynolds' result): each particle averages the *angle* of its velocity vector with that of its neighbors (those within a disk of a prespecified distance), plus a noise term used to model inaccuracies in averaging. Once the velocity vector is determined at each time, each particle takes a unit step along that direction, then determining its neighbors again and repeating the protocol.

The simulations were done in a square of unit length with periodic boundary conditions, to simulate infinite space. Vicsek and coauthors simulated this behavior and found that as the density of the particles increased, a preferred direction spontaneously emerged, resulting in a global group behavior with nearly aligned velocity vectors for all particles, despite the fact that the update protocol is entirely local.

With the interest in control theory shifting towards multiagent systems and networked coordination and control, it became clear that the mathematics of how birds flock and fish school and how individuals in a social network reach agreement (even though they are often only influenced by other like-minded individuals) are quite related to the question of how can one engineer a swarm of robots to behave like bird flocks.

These questions have occupied the minds of many researchers in diverse areas ranging from control theory to robotics, mathematics, and computer science. As discussed above, most of the early research, which happened in computer graphics and statistical physics, was on modeling and simulation of collective behavior. Over the past 13 years, however, the focus has shifted to rigorous systems theoretic foundations, leading to what one might call a theory of collective phenomena in multiagent systems. This theory blends dynamical systems, graph theory, Markov chains, and algorithms.

This type of collective phenomena are often modeled as many-degrees-of-freedom (discrete-time or continuous-time) dynamical systems with an additional twist that the interconnection



Flocking in Networked Systems, Fig. 1 Photo from <http://red3d.com/cwr/boids/>

structure between individual dynamical systems changes, since the motion of each node in a flock (or opinion of an individual) is affected primarily by those in each node's *local neighborhood*. The twist here is that the local neighborhood is not fixed: neighbors are defined based on the actual state of the system, for example, in case of Vicsek's alignment rule, as each particle averages its velocity direction with that of its neighbors and then takes a step, the set of its nearest neighbors can change.

Interestingly, very similar models were developed in statistics and mathematical sociology literature to describe how individuals in a social network update their opinions as a function of the opinion of their friends. The first such model goes back to the seminal work of DeGroot (1974) in 1974. DeGroot's model simply described the evolution of a group's scalar opinion as a function of the opinion of their neighbors by an iterative averaging scheme that can be conveniently modeled as a Markov chain. Individuals are represented by nodes of a graph, start from an opinion at time zero, and then are influenced by the people in their social clique. In DeGroot's model though, the network is given exogenously and does not change as a function of the opinions. The model therefore can be analyzed using the celebrated Perron-Frobenius theorem. The evolution of opinions is a discrete dynamic system that corresponds to an averaging map. When the network is fixed and connected (i.e., there is a path from every node to every other node) and agents also include their own opinions in the averaging, the update results in computation of a global weighted average of initial opinions,

where the weight of each initial opinion in the final aggregate is proportional to the "importance" of each node in the network.

The flocking models of Reynolds (1987) and Vicsek et al. (1995), however, have an extra twist: the network changes as opinions are updated. Similarly, more refined sociological models developed over the past decade also capture this endogeneity (Hegselmann and Krause 2002): each individual agent is influenced by others only when their opinion is close to her own. In other words, as opinions evolve, neighborhood structures change as the function of the evolving opinion, resulting in a switched dynamical system in which switching is state dependent.

In a paper in 2003, Jadbabaie and coauthors (2003) studied the Reynolds' alignment rule in the context of Vicsek's model when there is no exogenous noise. To model the endogeneity of the change in neighborhood structure, they developed a model based on repeated local averaging in which the neighborhood structure changes over time and therefore instead of a simple discrete-time linear dynamical system, the model is a discrete linear inclusion or a switched linear system. The question of interest was to determine what regimes of network changes could result in flocking. Clearly, as also DeGroot's model suggests, when the local neighbor structures do not change, connectivity is the key factor for flocking. This is a direct consequence of Perron-Frobenius theory. The result can also be described in terms of directed graphs. What is the equivalent condition in changing networks? Jadbabaie and coauthors show in their paper that indeed connectivity is important, but it need not hold every time: rather,

there needs to be time periods over which the graphs are *connected in time*. More formally, the process of neighborhood changes due to motion in Vicsek's model can be simply abstracted as a graph in which the links "blink" on and off. For flocking, one needs to ensure that there are time periods over which the union of edges (that occur as a result of proximity of particles) needs to correspond to a connected graph and such intervals need to occur infinitely often. It turns out that many of these ideas were developed much earlier in a thesis and following paper by Tsitsiklis (1984) and Tsitsiklis et al. (1986), in the context of distributed and asynchronous computation of global averages in changing graphs, and in a paper by Chatterjee and Seneta (1977), in the context of nonhomogeneous Markov chains. The machinery for proving such a results, however, is classical and has a long history in the theory of inhomogeneous Markov chains, a subject studied since the time of Markov himself, followed by Birkhoff and other mathematicians such as Hajnal, Dobrushin, Seneta, Hatfield, Daubachies, and Lagarias, to name a few.

The interesting twist in analysis of Vicsek's model is that the Euclidean norm of the distance to the globally converged "consensus angle" (or consensus opinion in the case of opinion models) can actually *grow* in a single step of the process; therefore, standard quadratic Lyapunov function arguments which serve as the main tool for analysis of switched linear systems are not suitable for the analysis of such models. However, it is fairly easy to see that under the process of local averaging, the largest value cannot increase and the smallest value cannot decrease. In fact, one can show that if enough connected graphs occur as the result of the switching process, the maximum value will be strictly decreasing and the minimum value will be strictly increasing. The paper by Jadbabaie and coauthors (2003) has lead to a flurry of results in this area over the past decade. One important generalization to the results of Jadbabaie et al. (2003), Tsitsiklis (1984), and Tsitsiklis et al. (1986) came 2 years later in a paper by Moreau (2005), who showed that these results can be generalized to nonlinear

updates and directed graphs. Moreau showed that any dynamic process that assigns a point in the interior of the convex hull of the value of each node and its neighbors will eventually result in agreement and consensus, if and only if the union of graphs from every time step till infinity contains a directed spanning tree (a node who has direct links to every other node).

Some of these results were also extended to the analysis of the Reynolds' model of flocking including the other two behaviors. First, Tanner and coauthors (2003, 2007) showed in a series of papers in 2003 and 2007 that a zone-based model similar to Reynolds can result in flocking for dynamic agents, provided that the graph representing interagent communications stays connected. Olfati-Saber and coauthors (2007) developed similar results with a slightly different model.

Many generalizations and extension for these results exist in a diverse set of disciplines, resulting in a rich theory which has had applications from robotics (such as rendezvous in mobile robots) (Cortés et al. 2006) to mathematical sociology (Hegselmann and Krause 2002) and from economics (Golub and Jackson 2010) to distributed optimization theory (Nedic and Ozdaglar 2009). However, some of the fundamental mathematical questions related to flocking still remain open.

First, most results focus on endogenous models of network change. A notable extension is a paper by Cucker and Smale (2007), in which the authors develop and analyze an endogenous model of flocking that cleverly *smoothens out* the discontinuous change in network structure by allowing each node's influence to decay smoothly as a function of distance.

Recently, in a series of papers, Chazelle has made progress in this arena by using tools from computational geometry and algorithms for analysis of endogenous models of flocking (Chazelle 2012). Chazelle has introduced the notion of the *s-energy* of a flock, which can be thought of as a parameterized family of Lyapunov functions that represent the evolution of global misalignment between flockmates. Via

tools from dynamical systems, computational geometry, combinatorics, complexity theory, and algorithms, Chazelle creates an “algorithmic calculus,” for *diffusive influence systems*: surprisingly, he shows that the orbit or flow of such systems is attracted to a fixed point in the case of undirected graphs and a limit cycle for almost all arbitrarily small random perturbations. Furthermore, the convergence time can also be bounded in both cases and the bounds are essentially optimal. The setup of the diffusive influence system developed by Chazelle creates a near-universal setup for analyzing various problems involving collective behavior in networked multiagent systems, from flocking, opinion dynamics, and information aggregation to synchronization problems.

To make further progress on analysis of what one might call networked dynamical systems (which Chazelle calls influence systems), one needs to combine mathematics of algorithms, complexity, combinatorics, and graphs with systems theory and dynamical systems.

Summary and Future Directions

This article presented a brief summary of the literature on flocking and distributed motion coordination. Flocking is the process by which various species exhibit synchronous collective motion from simple local interaction rules. Motivated by social aggregation in various species, various algorithms have been developed in the literature to design distributed control laws for group behavior in collective robotics and analysis of opinion dynamics in social networks. The models describe each agent as a kinematic or point mass particle that aligns each agent’s direction with that of its neighbors using repeated local averaging of directions. Since the neighborhood structures change due to motion, this results in a distributed switched dynamical system. If a weak notion of connectivity among agents is preserved over time, then agents reach consensus in their direction of motion. Despite the flurry of results in this area, the analysis of this phenomenon that

accounts for endogenous change in dynamics is for the most part open.

Cross-References

- [Averaging Algorithms and Consensus](#)
- [Oscillator Synchronization](#)

Bibliography

- Chatterjee S, Seneta E (1977) Towards consensus: some convergence theorems on repeated averaging. *J Appl Probab* 14:89–97
- Chazelle B (2012) Natural algorithms and influence systems. *Commun ACM* 55(12):101–110
- Cortés J, Martínez S, Bullo F (2006) Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Trans Autom Control* 51(8):1289–1298
- Cucker F, Smale S (2007) Emergent behavior in flocks. *IEEE Trans Autom Control* 52(5):852–862
- DeGroot MH (1974) Reaching a consensus. *J Am Stat Assoc* 69(345):118–121
- Golub B, Jackson MO (2010) Naive learning in social networks and the wisdom of crowds. *Am Econ J Microecon* 2(1):112–149
- Hegselmann R, Krause U (2002) Opinion dynamics and bounded confidence models, analysis, and simulation. *J Artif Soc Soc Simul* 5(3):2
- Jadbabaie A, Lin J, Morse AS (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans Autom Control* 48(6):988–1001
- Moreau L (2005) Stability of multiagent systems with time-dependent communication links. *IEEE Trans Autom Control* 50(2):169–182
- Nedic A, Ozdaglar A (2009) Distributed subgradient methods for multi-agent optimization. *IEEE Trans Autom Control* 54(1):48–61
- Olfati-Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. *Proc IEEE* 95(1):215–233
- Reynolds CW (1987) Flocks, herds, and schools: a distributed behavioral model. *Comput Graph* 21(4):25–34. (SIGGRAPH ’87 Conference Proceedings)
- Tanner HG, Jadbabaie A, Pappas GJ (2003) Stable flocking of mobile agents, Part I: fixed Topology, Part II: switching topology. In: *Proceedings of the 42nd IEEE conference on decision and control, Maui, vol 2*. IEEE, pp 2010–2015
- Tanner HG, Jadbabaie A, Pappas GJ (2007) Flocking in fixed and switching networks. *IEEE Trans Autom Control* 52(5):863–868

- Tsitsiklis JN (1984) Problems in decentralized decision making and computation (No. LIDS-TH-1424). Laboratory for Information and Decision Systems, Massachusetts Institute of Technology
- Tsitsiklis J, Bertsekas D, Athans M (1986) Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans Autom Control* 31(9):803–812
- Vicsek T, Czirók A, Ben-Jacob E, Cohen I, Shochet O (1995) Novel type of phase transition in a system of self-driven particles. *Phys Rev Lett* 75(6):1226

Force Control in Robotics

Luigi Villani

Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, Università degli Studi di Napoli Federico II, Napoli, Italy

Abstract

Force control is used to handle the physical interaction between a robot and the environment and also to ensure safe and dependable operation in the presence of humans. The control goal may be that to keep the interaction forces limited or that to guarantee a desired force along the directions where interaction occurs while a desired motion is ensured in the other directions. This entry presents the basic control schemes, focusing on robot manipulators.

Keywords

Compliance control · Constrained motion · Force control · Force/torque sensor · Hybrid force/motion control · Impedance control · Stiffness control

Introduction

Control of the physical interaction between a robot manipulator and the environment is crucial for the successful execution of a number of practical tasks where the robot end effector

has to manipulate an object or perform some operation on a surface. Typical examples in industrial settings include polishing, deburring, machining, or assembly.

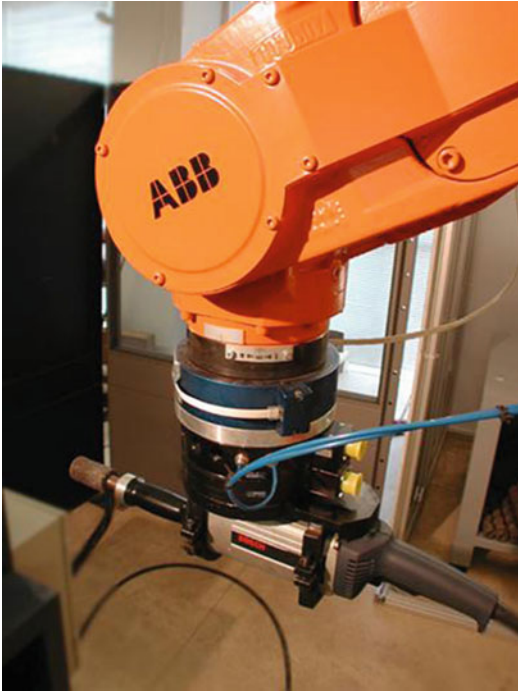
During contact, the environment may set constraints on the geometric paths that can be followed by the robot's end effector (kinematic constraints) as in the case of sliding on a rigid surface. In other situations, the interaction occurs with a dynamic environment as in the case of collaboration with a human. In all cases, a pure motion control strategy is not recommended, especially if the environment is stiff.

The higher the environment stiffness and position control accuracy are, the more easily the contact forces may rise and reach unsafe values. This drawback can be overcome by introducing compliance, either in a passive or in an active fashion, to accommodate the robot motion in response to interaction forces.

Passive compliance may be due to the structural compliance of the links, joints, and end effector or to the compliance of the position servo. Soft robot arms with elastic joints or links are purposely designed for intrinsically safe interaction with humans. In contrast, active compliance is entrusted to the control system, denoted *interaction control* or *force control*. In some cases, the measurement of the contact force and moment is required, which is fed back to the controller and used to modify or even generate online the desired motion of the robot (Whitney 1977).

The passive solution is faster than active reaction commanded by a computer control algorithm. However, the use of passive compliance alone lacks of flexibility and cannot guarantee that high contact forces will never occur. Hence, the most effective solution is that of using active force control (with or without force feedback) in combination with some degree of passive compliance.

In general, six force components are required to provide complete contact force information: three translational force components and three torques. Often, a *force/torque sensor* is mounted at the robot wrist (see an example in Fig. 1),



Force Control in Robotics, Fig. 1 Industrial robot with wrist force/torque sensor and deburring tool

but other possibilities exist, for example, force sensors can be placed on the fingertips of robotic hands; also, external forces and moments can be estimated via shaft torque measurements of joint torque sensors.

The force control strategies can be grouped into two categories (Siciliano and Villani 1999): those performing indirect force control and those performing direct force control. The main difference between the two categories is that the former achieve force control via motion control, without explicit closure of a force feedback loop; the latter instead offer the possibility of controlling the contact force and moment to a desired value, thanks to the closure of a force feedback loop.

Modeling

The case of interaction of the end effector of a robot manipulator with the environment is con-

sidered, which is the most common situation in industrial applications.

The end-effector pose can be represented by the position vector p_e and the rotation matrix $\mathbf{R}_e$, corresponding to the position and orientation of a frame attached to the end effector with respect to a fixed-base frame.

The end-effector velocity is denoted by the 6×1 twist vector $\mathbf{v}_e = (\dot{p}_e^T \ \omega_e^T)^T$ where $\dot{p}_e$ is the translational velocity and ω_e is the angular velocity and can be computed from the joint velocity vector $\dot{q}$ using the linear mapping

$$\mathbf{v}_e = \mathbf{J}(q)\dot{q}.$$

The matrix $\mathbf{J}$ is the end-effector Jacobian. For simplicity, the case of nonredundant nonsingular manipulators is considered; therefore, the Jacobian is a square nonsingular matrix.

The force f_e and moment m_e applied by the end effector to the environment are the components of the wrench $h_e = (f_e^T \ m_e^T)^T$. The joint torques τ corresponding to h_e can be computed as

$$\tau = \mathbf{J}^T(q)h_e.$$

It is useful to consider the operational space formulation of the dynamic model of a rigid robot manipulator in contact with the environment (Khatib 1987):

$$\Lambda(q)\dot{\mathbf{v}}_e + \Gamma(q, \dot{q})\mathbf{v}_e + \eta(q) = h_c - h_e, \quad (1)$$

where $\Lambda(q)$ is the 6×6 operational space inertia matrix, $\Gamma(q, \dot{q})$ is the wrench including centrifugal and Coriolis effects, and $\eta(q)$ is the wrench of the gravitational effects. The vector $h_c = \mathbf{J}^{-T}\tau$ is the equivalent end-effector wrench corresponding to the input joint torques τ_c .

Equation (1) can be seen as a representation of the Newton's Second Law of Motion where all the generalized forces acting on the joints of the robot are reported at the end effector.

The full specification of the system dynamics would require also the analytic description of the interaction force and moment h_e . This is a very demanding task from a modeling viewpoint.

The design of the interaction control and the performance analysis are usually carried out under simplifying assumptions. The following two cases are considered:

1. The robot is perfectly rigid, all the compliance in the system is localized in the environment, and the contact wrench is approximated by a linear elastic model.
2. The robot and the environment are perfectly rigid and purely kinematics constraints are imposed by the environment.

It is obvious that these situations are only ideal. However, the robustness of the control should be able to cope with situations where some of the ideal assumptions are relaxed. In that case the control laws may be adapted to deal with nonideal characteristics.

Indirect Force Control

The aim of indirect force control is that of achieving a desired compliant dynamic behavior of the robot's end effector in the presence of interaction with the environment.

Stiffness Control

The simpler approach is that of imposing a suitable static relationship between the deviation of the end-effector position and orientation from a desired pose and the force exerted on the environment, by using the control law

$$h_c = \mathbf{K}_P \Delta x_{de} - \mathbf{K}_D \mathbf{v}_e + \boldsymbol{\eta}(q), \quad (2)$$

where $\mathbf{K}_P$ and $\mathbf{K}_D$ are suitable matrix gains and Δx_{de} is a suitable error between a desired and the actual end-effector position and orientation. The position error component of Δx_{de} can be simply chosen as $p_d - p_e$. Concerning the orientation error component, different choices are possible (Caccavale et al. 1999), which are not all equivalent, but this issue is outside the scope of this entry.

The control input (2) corresponds to a wrench (force and moment) applied to the end effector,

which includes a gravity compensation term $\boldsymbol{\eta}(q)$, a viscous damping term $\mathbf{K}_D \mathbf{v}_e$, and an elastic wrench provided by a virtual spring with stiffness matrix $\mathbf{K}_P$ (or, equivalently, compliance matrix $\mathbf{K}_P^{-1}$) connecting the end-effector frame with a frame of desired position and orientation. This control law is known as *stiffness control* or *compliance control* (Salisbury 1980).

Using the Lyapunov method, it is possible to prove the asymptotic stability of the equilibrium solution of equation

$$\mathbf{K}_P \Delta x_{de} = h_e,$$

meaning that, at steady state, the robot's end effector has a desired elastic behavior under the action of the external wrench h_e . It is clear that, if $h_e \neq \mathbf{0}$, then the end effector deviates from the desired pose, which is usually denoted as *virtual pose*.

Physically, the closed-loop system (1) with (2) can be seen as a 6-DOF nonlinear and configuration-dependent mass-spring-damper system with inertia (mass) matrix $\boldsymbol{\Lambda}(q)$ and adjustable damping $\mathbf{K}_D$ and stiffness $\mathbf{K}_P$, under the action of the external wrench h_e .

Impedance Control

A configuration-independent dynamic behavior can be achieved if the measure of the end-effector force and moment h_e is available, by using the control law, known as *impedance control* (Hogan 1985):

$$h_c = \boldsymbol{\Lambda}(q)\boldsymbol{\alpha} + \boldsymbol{\Gamma}(q, \dot{q})\dot{q} + \boldsymbol{\eta}(q) + h_e,$$

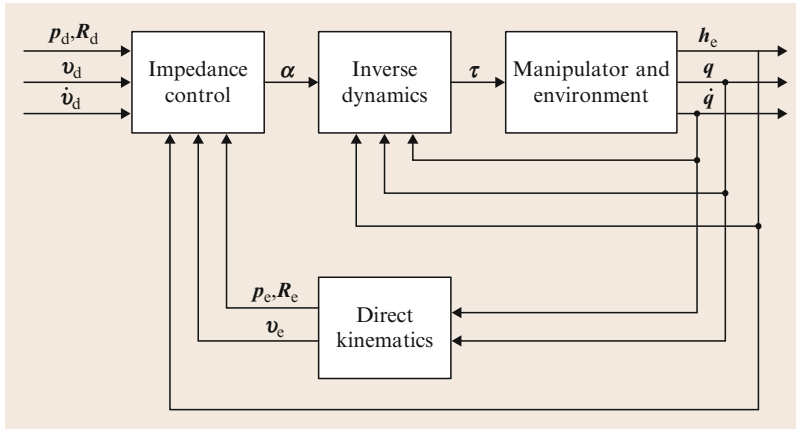
where $\boldsymbol{\alpha}$ is chosen as:

$$\boldsymbol{\alpha} = \dot{\mathbf{v}}_d + \mathbf{K}_M^{-1}(\mathbf{K}_D \Delta \mathbf{v}_{de} + \mathbf{K}_P \Delta x_{de} - h_e).$$

The following expression can be found for the closed-loop system

$$\mathbf{K}_M \Delta \dot{\mathbf{v}}_{de} + \mathbf{K}_D \Delta \mathbf{v}_{de} + \mathbf{K}_P \Delta x_{de} = h_e, \quad (3)$$

representing the equation of a 6-DOF configuration-independent mass-spring-damper system with adjustable inertia (mass) matrix



Force Control in Robotics, Fig. 2 Impedance control

$\mathbf{K}_M$, damping $\mathbf{K}_D$, and stiffness $\mathbf{K}_P$, known as mechanical *impedance*.

A block diagram of the resulting impedance control is sketched in Fig. 2.

The selection of good impedance parameters ensuring a satisfactory behavior is not an easy task and can be simplified under the hypothesis that all the matrices are diagonal, resulting in a decoupled behavior for the end-effector coordinates.

Moreover, the dynamics of the controlled system during the interaction depends on the dynamics of the environment that, for simplicity, can be approximated as a simple elastic law for each coordinate, of the form

$$h_e = k \Delta x_{e0},$$

where $\Delta x_{e0} = x_e - x_0$, while x_0 and k are the undeformed position and the stiffness coefficient of the spring, respectively.

In the above hypotheses, the transient behavior of each component of Eq. (3) can be set by assigning the natural frequency and damping ratio with the relations

$$\omega_n = \sqrt{\frac{k_P + k}{k_M}}, \quad \zeta = \frac{1}{2} \frac{k_D}{\sqrt{k_M(k_P + k)}}.$$

Hence, if the gains are chosen so that a given natural frequency and damping ratio are ensured during the interaction (i.e., for $k \neq 0$), a smaller natural frequency with a higher damping ratio will be obtained when the end effector moves in free space (i.e., for $k = 0$). As for the steady-state performance, the end-effector error and the interaction force for the generic component are

$$\Delta x_{de} = \frac{k}{(k_P + k)} \Delta x_{do}, \quad h = \frac{k_P k}{k_P + k} \Delta x_{do},$$

showing that, during interaction, the contact force can be made small at the expense of a large position error in steady state, as long as the robot stiffness k_P is set low with respect to the stiffness of the environment k and vice versa.

Direct Force Control

Indirect force control does not require explicit knowledge of the environment, although to achieve a satisfactory dynamic behavior, the control parameters have to be tuned for a particular task. On the other hand, a model of the interaction task is usually required for the synthesis of direct force control algorithms.

In the following, it is assumed that the environment is rigid and frictionless and imposes kinematic constraints to the robot's end-effector motion (Mason 1981). These constraints reduce the dimension of the space of the feasible end-effector velocities and of the contact forces and moments. In detail, in the presence of m independent constraints ($m < 6$), the end-effector velocity belongs to a subspace of dimension $6 - m$, while the end-effector wrench belongs to a subspace of dimension m and can be expressed in the form

$$\mathbf{v}_e = \mathbf{S}_v(q)\mathbf{v}, \quad \mathbf{h}_e = \mathbf{S}_f(q)\boldsymbol{\lambda}$$

where $\mathbf{v}$ is a suitable $(6 - m) \times 1$ vector and $\boldsymbol{\lambda}$ is a suitable $m \times 1$ vector. Moreover, the subspaces of forces and velocity are *reciprocal*, i.e.:

$$\mathbf{h}_e^T \mathbf{v}_e = 0, \quad \mathbf{S}_f^T(q)\mathbf{S}_v(q) = \mathbf{0}.$$

The concept of reciprocity expresses the physical fact that, in the hypothesis of rigid and frictionless contact, the wrench does not cause any work against the twist.

An interaction task can be assigned in terms of a desired end-effector twist $\mathbf{v}_d$ and wrench $\mathbf{h}_d$ that are computed as:

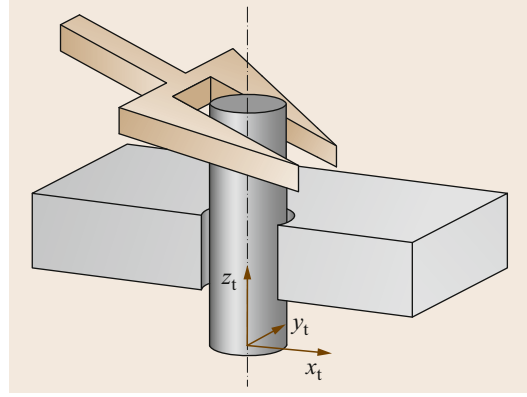
$$\mathbf{v}_d = \mathbf{S}_v \mathbf{v}_d, \quad \mathbf{h}_d = \mathbf{S}_f \boldsymbol{\lambda}_d,$$

by specifying vectors $\boldsymbol{\lambda}_d$ and $\mathbf{v}_d$.

In many robotic tasks it is possible to set an orthogonal reference frame, usually referred as *task frame* (De Schutter and Van Brussel 1988), in which the matrices $\mathbf{S}_v$ and $\mathbf{S}_f$ are constant. Moreover, the interaction task is specified by assigning a desired force/torque or a desired linear/angular velocity along/about each of the frame axes.

An example of task frame definition and task specification is given below.

Peg-in-Hole: The goal of this task is to push the peg into the hole while avoiding wedging and jamming. The peg has two degrees of motion freedom; hence, the dimension of the velocity-controlled subspace is $6 - m = 2$, while the dimension of the force-controlled subspace is



Force Control in Robotics, Fig. 3 Insertion of a cylindrical peg into a hole

$m = 4$. The task frame can be chosen as shown in Fig. 3, and the task can be achieved by assigning the following desired forces and torques:

- Zero forces along the x_t and y_t axes
- Zero torques about the x_t and y_t axes and the desired velocities
- A nonzero linear velocity along the z_t -axis
- An arbitrary angular velocity about the z_t -axis

The task continues until a large reaction force in the z_t direction is measured, indicating that the peg has hit the bottom of the hole, not represented in the figure. Hence, the matrices $\mathbf{S}_f$ and $\mathbf{S}_v$ can be chosen as

$$\mathbf{S}_f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{S}_v = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

The task frame can be chosen attached either to the end effector or to the environment.

Hybrid Force/Motion Control

The reciprocity of the velocity and force subspaces naturally leads to a control approach, known as *hybrid force/motion control* (Raibert and Craig 1981; Yoshikawa 1987), aimed at

controlling simultaneously both the contact force and the end-effector motion in two reciprocal subspaces.

The reduced order dynamics of the robot with kinematic constraints is described by $6 - m$ second-order equations

$$\mathbf{A}_v(q)\dot{\mathbf{v}} = \mathbf{S}_v^T [h_c - \boldsymbol{\mu}(q, \dot{q})],$$

where $\mathbf{A}_v = \mathbf{S}_v^T \mathbf{A} \mathbf{S}_v$ and $\boldsymbol{\mu}(q, \dot{q}) = \boldsymbol{\Gamma}(q, \dot{q})\mathbf{v}_e + \boldsymbol{\eta}(q)$, assuming constant matrices $\mathbf{S}_v$ and $\mathbf{S}_f$. Moreover, the vector $\boldsymbol{\lambda}$ can be computed as

$$\boldsymbol{\lambda} = \mathbf{S}_f^\dagger(q)[h_c - \boldsymbol{\mu}(q, \dot{q})],$$

revealing that the contact force is a constraint force which instantaneously depends on the applied input wrench h_c .

An inverse-dynamics inner control loop can be designed by choosing the control wrench h_c as

$$h_c = \mathbf{A}(q)\mathbf{S}_v\boldsymbol{\alpha}_v + \mathbf{S}_f f_\lambda + \boldsymbol{\mu}(q, \dot{q}),$$

where $\boldsymbol{\alpha}_v$ and f_λ are properly designed control inputs, which leads to the equations

$$\dot{\mathbf{v}} = \boldsymbol{\alpha}_v, \quad \dot{\boldsymbol{\lambda}} = f_\lambda,$$

showing a complete decoupling between motion control and force control.

Then, the desired force $\boldsymbol{\lambda}_d(t)$ can be achieved by setting

$$f_\lambda = \boldsymbol{\lambda}_d(t),$$

but this choice is very sensitive to disturbance forces, since it contains no force feedback. Alternative choices are

$$f_\lambda = \boldsymbol{\lambda}_d(t) + \mathbf{K}_{P\lambda}[\boldsymbol{\lambda}_d(t) - \boldsymbol{\lambda}(t)],$$

or

$$f_\lambda = \boldsymbol{\lambda}_d(t) + \mathbf{K}_{I\lambda} \int_0^t [\boldsymbol{\lambda}_d(\tau) - \boldsymbol{\lambda}(\tau)] d\tau,$$

where $\mathbf{K}_{P\lambda}$ and $\mathbf{K}_{I\lambda}$ are suitable positive-definite matrix gains. The proportional feedback is able to

reduce the force error due to disturbance forces, while the integral action is able to compensate for constant bias disturbances.

Velocity control is achieved by setting

$$\begin{aligned} \boldsymbol{\alpha}_v &= \dot{\mathbf{v}}_d(t) + \mathbf{K}_{Pv}[\mathbf{v}_d(t) - \mathbf{v}(t)] \\ &+ \mathbf{K}_{Iv} \int_0^t [\mathbf{v}_d(\tau) - \mathbf{v}(\tau)] d\tau, \end{aligned}$$

where $\mathbf{K}_{Pv}$ and $\mathbf{K}_{Iv}$ are suitable matrix gains. It is straightforward to show that asymptotic tracking of $\mathbf{v}_d(t)$ and $\dot{\mathbf{v}}_d(t)$ is ensured with exponential convergence for any choice of positive-definite matrices $\mathbf{K}_{Pv}$ and $\mathbf{K}_{Iv}$.

Notice that the implementation of force feedback requires the computation of vector $\boldsymbol{\lambda}$ from the measurement of the end-effector wrench h_e as $\mathbf{S}_f^\dagger \mathbf{v}_e$, being $\mathbf{S}_f^\dagger$ a suitable pseudoinverse of matrix $\mathbf{S}_f$. Analogously, vector $\mathbf{v}$ can be computed from $\mathbf{v}_e$ as $\mathbf{S}_v^\dagger \mathbf{v}_e$.

The hypothesis of rigid contact can be removed, and this implies that along some directions both motion and force are allowed, although they are not independent. Hybrid force/motion control schemes can be defined also in this case Villani and De Schutter (2008).

Summary and Future Directions

This entry has sketched the main approaches to force control in a unifying perspective. However, there are many aspects that have not been considered here. The two major paradigms of force control (impedance and hybrid force/motion control) are based on several simplifying assumptions that are only partially satisfied in practical implementations and that have been partially removed in more advanced control methods.

Notice that the performance of a force-controlled robotic system depends on the interaction with a changing environment which is very difficult to model and identify correctly. Hence, the standard performance indices used to evaluate a control system, i.e., stability, bandwidth, accuracy, and robustness, cannot

be defined by considering the robotic system alone, as for the case of robot motion control, but must be always referred to the particular contact situation at hand.

Force control in industrial applications can be considered as a mature technology, although, for the reason explained above, standard design methodologies are not yet available. Force control techniques are employed also in medical robotics, haptic systems, telerobotics, humanoid robotics, micro-robotics, and nano robotics. An interesting field of application is related to human-centered robotics, where control plays a key role to achieve adaptability, reaction capability, and safety. Robots and biomechatronic systems based on the novel variable impedance actuators, with physically adjustable compliance and damping, capable to react softly when touching the environment, necessitate the design of specific control laws. The combined use of exteroceptive sensing (visual, depth, proximity, force, tactile sensing) for reactive control in the presence of uncertainty represents another challenging research direction.

Cross-References

- [Robot Grasp Control](#)
- [Robot Motion Control](#)

Recommended Reading

This entry has presented a brief overview of the basic force control techniques, and the cited references represent a selection of the main pioneering contributions. A more extensive treatment of this topic with related bibliography can be found in Villani and De Schutter (2008). Besides impedance control and hybrid force/position control, an approach designed to cope with uncertainties in the environment geometry is the parallel force/position control (Chiaverini and Sciacicco 1993; Chiaverini et al. 1994). In the paper Ott et al. (2008) the passive compliance of lightweight robots is combined with the

active compliance ensured by impedance control. A systematic constraint-based methodology to specify complex tasks has been presented by De Schutter et al. (2007).

Bibliography

- Caccavale F, Natale C, Siciliano B, Villani L (1999) Six-DOF impedance control based on angle/axis representations. *IEEE Trans Robot Autom* 15:289–300
- Chiaverini S, Sciacicco L (1993) The parallel approach to force/position control of robotic manipulators, *IEEE Trans Robot Autom* 9:361–373
- Chiaverini S, Siciliano B, Villani L (1994) Force/position regulation of compliant robot manipulators. *IEEE Trans Autom Control* 39:647–652
- De Schutter J, Van Brussel H (1988) Compliant robot motion I. A formalism for specifying compliant motion tasks. *Int J Robot Res* 7(4):3–17
- De Schutter J, De Laet T, Rutgeerts J, Decré W, Smits R, Aerbeliën E, Claes K, Bruyninckx H (2007) Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *Int J Robot Res* 26(5):433–455
- Hogan N (1985) Impedance control: an approach to manipulation: parts I–III. *ASME J Dyn Syst Meas Control* 107:1–24
- Khatib O (1987) A unified approach for motion and force control of robot manipulators: the operational space formulation. *IEEE J Robot Autom* 3:43–53
- Mason MT (1981) Compliance and force control for computer controlled manipulators. *IEEE Trans Syst Man Cybern* 11:418–432
- Ott C, Albu-Schaeffer A, Kugi A, Hirzinger G (2008) On the passivity based impedance control of flexible joint robots. *IEEE Trans Robot* 24:416–429
- Raibert MH, Craig JJ (1981) Hybrid position/force control of manipulators. *ASME J Dyn Syst Meas Control* 103:126–133
- Salisbury JK (1980) Active stiffness control of a manipulator in Cartesian coordinates. In: 19th IEEE conference on decision and control, Albuquerque, pp 95–100
- Siciliano B, Villani L (1999) *Robot force control*. Kluwer, Boston
- Villani L, De Schutter J (2008) *Robot force control*. In: Siciliano B, Khatib O (eds) *Springer handbook of robotics*. Springer, Berlin, pp 161–185
- Whitney DE (1977) Force feedback control of manipulator fine motions. *ASME J Dyn Syst Meas Control* 99:91–97
- Yoshikawa T (1987) Dynamic hybrid position/force control of robot manipulators – description of hand constraints and calculation of joint driving force. *IEEE J Robot Autom* 3:386–392

Formal Methods for Controlling Dynamical Systems

Calin Belta

Mechanical Engineering, Boston University,
Boston, MA, USA

Abstract

In control theory, complicated dynamics such as systems of (nonlinear) differential equations are mostly controlled to achieve stability. This fundamental property, which can be with respect to a desired operating point or a prescribed trajectory, is often linked with optimality, which requires minimization of a certain cost along the trajectories of a stable system. In formal methods, rich specifications, such as formulas of temporal logics, are checked against simple models of software programs and digital circuits, such as finite transition systems. With the development and integration of cyber physical and safety critical systems, there is an increasing need for computational tools for verification and control of complex systems from rich, temporal logic specifications. The current approaches to formal synthesis of (optimal) control strategies for dynamical systems can be roughly divided in two classes: abstraction-based and optimization-based methods. In this entry, we provide a short overview of these techniques.

Keywords

Formal verification · Model checking ·
Hybrid systems · Optimal control

Introduction

Temporal logics, such as computation tree logic (CTL) and linear temporal logic (LTL), have been customarily used to specify the correctness of computer programs and digital circuits modeled as finite-state transition systems. The problem of analyzing such a model against a temporal

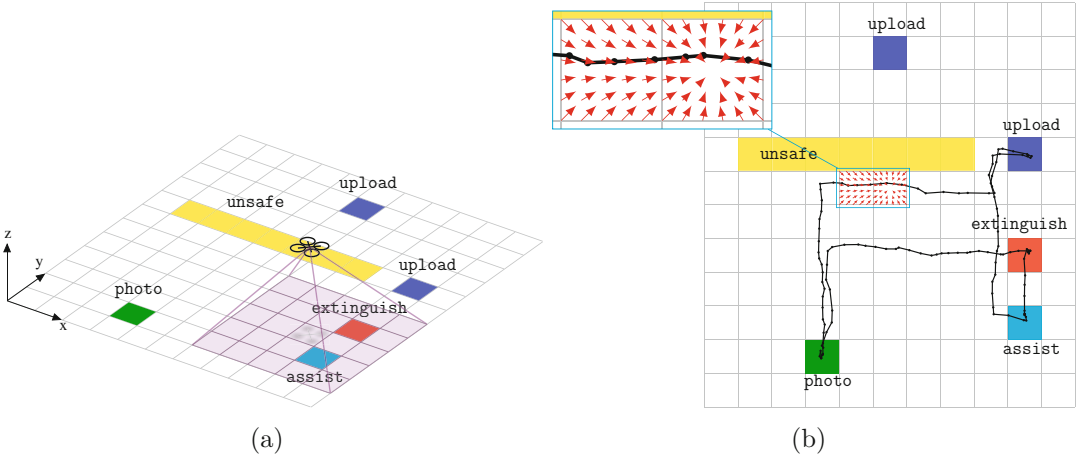
logic formula, known as formal analysis or model checking, has received a lot of attention during the past 40 years, and several efficient algorithms and software tools are available. The formal synthesis problem, in which the goal is to design or control a system from a temporal logic specification, has not been studied extensively until a few years ago.

Control and optimal control are mature research areas with many applications. They cover a large spectrum of systems, including (weighted) finite deterministic transition systems (i.e., graphs for which available transitions can be deterministically chosen at every node), finite purely nondeterministic systems (where an action at a state enables several transitions and their probabilities are not known), finite Markov decision processes (MDP), and systems with infinite state and control sets. For the latter, optimal control problems usually involve costs penalizing the deviation of the state from a reference trajectory and the control effort.

The connection between (optimal) control and formal methods is an intriguing problem with potentially high impact in several applications. By combining these two seemingly unrelated areas, the goal is to control the behavior of a system subject to correctness constraints. Consider, for example, an autonomous vehicle involved in a persistent surveillance mission in a disaster relief application, where dynamic service requests can only be sensed locally in a neighborhood around the vehicle (see Fig. 1). The goal is to accomplish the mission while at the same time maximizing the likelihood of servicing the local requests and possibly minimizing the energy spent during the motion. The correctness requirement can be expressed as a temporal logic formula (see the caption of Fig. 1), while the resource constraints translate to minimizing a cost over the feasible trajectories of the vehicle.

Formal Synthesis of Control Strategies

Current works on combining control and formal methods can be roughly divided into two



Formal Methods for Controlling Dynamical Systems,

Fig. 1 (a) An autonomous air vehicle is deployed from a high level, temporal logic global specification over a set of static, known requests (photo and upload) occurring at the regions of a known environment, e.g., “Keep taking photos and upload current photo before taking another photo.” This specification translates to the following LTL formula: $\mathbf{GF} \text{ photo} \wedge \mathbf{G} (\text{photo} \rightarrow (\text{photo U} (\neg \text{photo U upload})))$, where $\mathbf{G}$, $\mathbf{F}$, and $\mathbf{U}$ are the temporal operators Globally (Always), Future (Eventually), and Until; $\wedge$, $\rightarrow$, $\neg$ are Boolean operators for conjunction, implication, and negation, respectively. While moving in the environment, the vehicle can locally sense dynamically changing events, such as survivors, and fires, which generate (local) service

requests, and unsafe areas, which need to be avoided. The goal is to accomplish the global mission while at the same time maximizing the likelihood of servicing the local requests and staying away from unsafe areas. (b) By using an accurate quad-rotor kinematic model, input-output linearizations/flat outputs, precise state information from a motion capture system, and control-to-facet results for linear and multi-affine systems, this problem can be (conservatively) mapped to a control problem for a finite transition system. This can be deterministic or non-deterministic if single-facet or multiple-facet controllers in the output space are used, respectively. (Example adapted from Ulusoy and Belta 2014)

main classes: automata-based methods and optimization-based methods.

Automata-Based Methods

Automata-based methods are based on the observation that a temporal logic formula, such as an LTL formula, can be mapped to an automaton in such a way that the language accepted by the automaton is exactly the language satisfying the formula. Depending on the desired expressivity of the specification language, these automata can be well-known finite state automata (FSA) (the acceptance condition is reaching a set of final states), Büchi automata (the acceptance condition is reaching a set of final states infinitely often), or, in the most general case, Rabin automata (the acceptance condition is to visit a set of “good” states infinitely often and a set of “bad” states finitely many times). For finite systems, the control problem reduces to a game on the product between a system, such as a transition

system or an MDP, and the automaton obtained from the specification. The winning condition, which ensures correctness, is the Rabin (Büchi, FSA) acceptance condition of the automaton. The cost can be average reward/cost per stage and adapted objectives that reflect the semantics of the temporal logic, such as average reward/cost per cycle.

For infinite systems, automata-based approaches are, in general, hierarchical, two-level methods. The bottom level is a continuous-to-continuous abstraction procedure, in which the possibly large state space and complex dynamics are mapped to a low-dimensional output space with simple dynamics. The most used techniques are input-output linearization and differential flatness. For a differentially flat system, its state and control variables can be expressed as a function of its outputs and its derivatives. The top level is a partition-based, continuous-to-discrete abstraction procedure, in which the output and control

spaces are partitioned. The partition can be driven by the specification or by a prescribed accuracy of the approximation. The quotient of the partition is a finite system that is in some way equivalent with the original, infinite system. The most used notion of equivalence is bisimulation. An example is shown in Fig. 1. The 12-dimensional quad-rotor dynamics of the quad-rotor shown in the left are differentially flat with four flat outputs (position and yaw), and up to four derivatives of the flat output are necessary to compute the original state and input. A two-dimensional section of the partition of the four-dimensional output space is shown on the right, together with the assignment of a vector field in two adjacent cells. Note that the dynamics corresponding to these vector fields “treat” all the states in a cell “in the same way”: in the cell on the left, all the states will leave in finite time through the right facet; in the cell on the right, all states will stay inside for all times. Informally, these correspond to the bisimilarity equivalence mentioned above. The quotient of the partition is a finite transition system that is controlled from the temporal logic specification. The cost can penalize the execution time, travelled distance, etc.

The method described above is conservative. If a solution (of the automaton game) is not found at the top level, this does not mean that a controller does not exist for the original continuous system. Intuitively, the partition, and therefore the abstraction, might be too rough. Conservativeness can be reduced by refining the partition. Numerous partition techniques that exploit the connection between the dynamics of the system and the geometry of the partition have been proposed. An example is shown in Fig. 2. An example illustrating the compromise between correctness and optimality is shown in Fig. 3.

The expensive process of constructing the abstraction can be avoided for both deterministic and stochastic systems. Specifically, a dynamic programming problem can be formulated over the product of the continuous-time, continuous-state system, and the specification automaton. Approximate dynamic programming (ADP) approaches have been shown to work for both

linear and nonlinear systems. Sampling-based policy iteration has also been used for optimal planning for a subclass of LTL specifications.

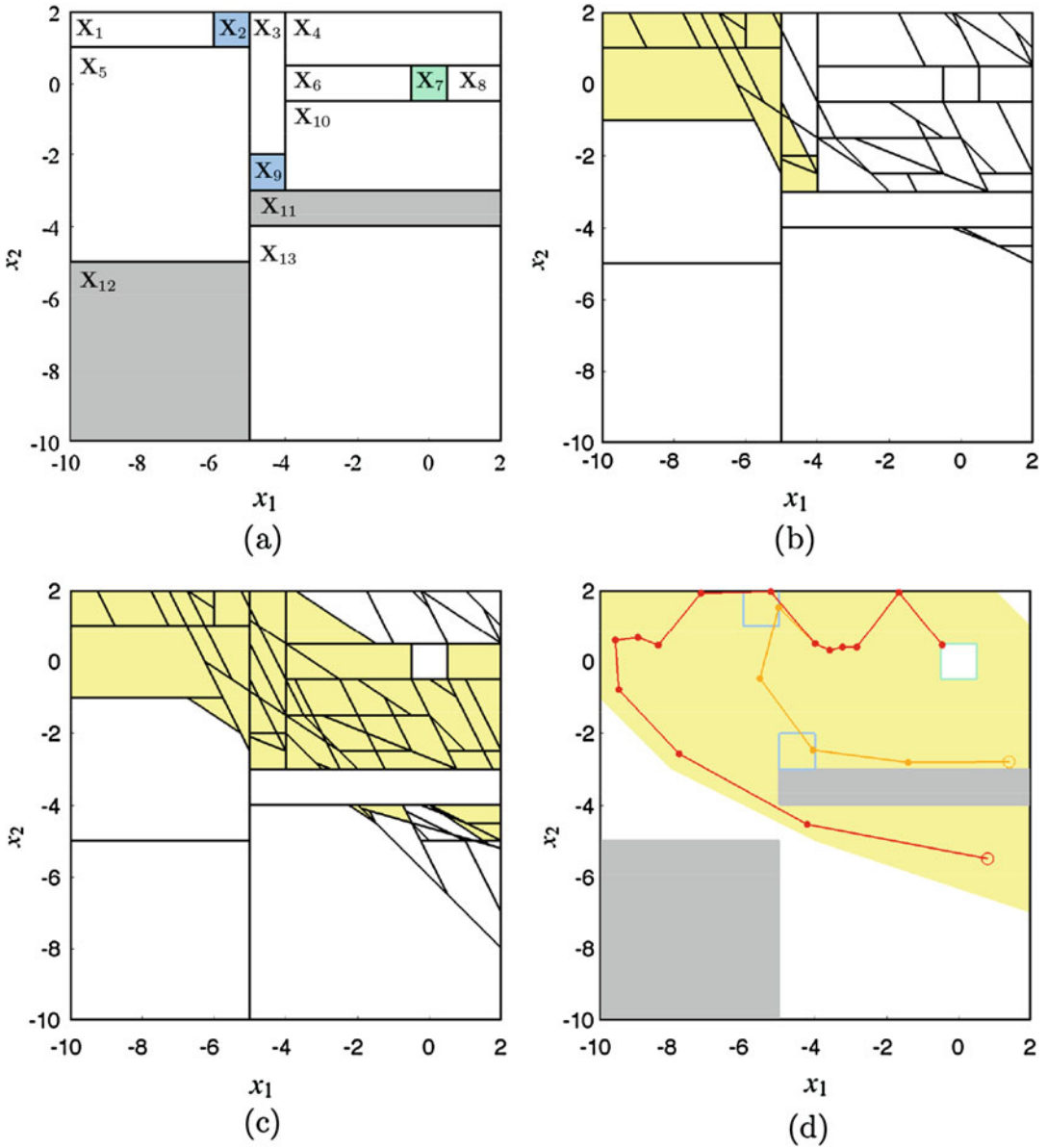
Optimization-Based Methods

There are roughly two classes of optimization-based methods for formal synthesis: mixed integer programming (MIP) methods and control barrier functions (CBF) methods.

Central to the MIP-based methods are temporal logics with semantics over finite-time signals, such as signal temporal logic (STL) and metric temporal logic (MTL). For simplicity, in this article we focus on STL. The main difference between STL and LTL (see the caption of Fig. 1 for an example of an LTL formula) is that STL temporal operators are explicit (see Fig. 4 for an example of an STL formula). In addition to Boolean semantics, in which signals either satisfy or violate a formula, STL has quantitative semantics, which allow to assess the robustness of satisfaction. Specifically, given a formula ϕ and a signal x , the robustness $\rho(\phi, x)$ quantifies how well x satisfies ϕ . The more x satisfies ϕ , the larger $\rho(\phi, x)$ is (if x satisfies ϕ in Boolean semantics, then $\rho(\phi, x) > 0$). The more x violates ϕ , the smaller $\rho(\phi, x)$ is (if x violates ϕ in Boolean semantics, then $\rho(\phi, x) < 0$).

It can be shown that Boolean satisfaction of STL (MTL) formulas over linear predicates in the state x of a system can be mapped to the feasibility part of a mixed integer linear program (MILP) (i.e., a set of linear equalities and inequalities over the state and an additional set of integers). This observation implies that controlling a linear (or almost linear, such as piecewise affine, mixed logical, etc.) system, such that a linear or quadratic cost is optimized while satisfying STL formulas over linear predicates over its state, maps to solving a mixed integer linear program (MILP) or mixed integer quadratic program (MIQP), for which there exist computationally efficient solvers.

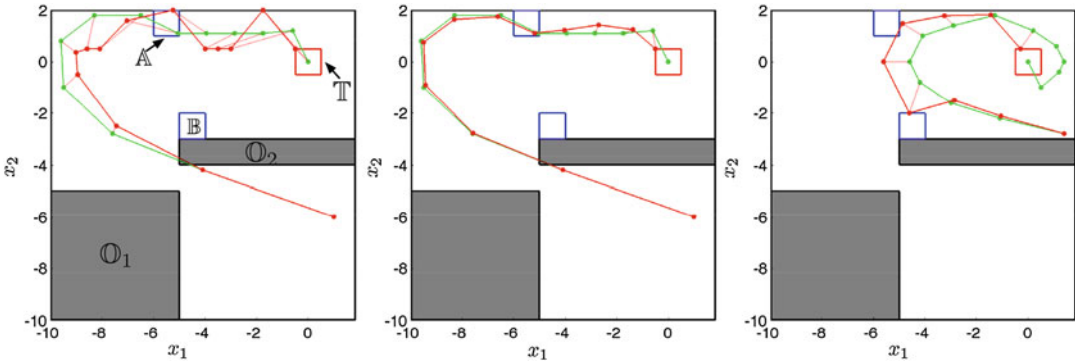
Moreover, it can be shown that the robustness function ρ is linear in state and the additional integer variables. Therefore, if robustness is



Formal Methods for Controlling Dynamical Systems,

Fig. 2 A discrete-time double integrator system $x[t + 1] = Ax[t] + Bu[t]$, with $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 0.5 & 1 \end{bmatrix}$, and $u \in [-2, 2]$, moving in a planar environment (a) partitioned into $X_1, X_2, \dots, X_{13}$ is required to satisfy the following specification: “Visit region $\mathbb{A} = X_2$ or region $\mathbb{B} = X_9$, and then the target region $\mathbb{T} = X_7$, while always avoiding obstacles $\mathbb{O}_1 = X_{11}$ and $\mathbb{O}_2 = X_{12}$, which translates to the syntactically co-safe LTL (scLTL) formula: $((\neg \mathbb{O}_1 \wedge \neg \mathbb{O}_2) \mathbf{U} \mathbb{T}) \wedge (\neg \mathbf{T} \mathbf{U} (\mathbb{A} \vee \mathbb{B}))$. Iterations 50, 80, and 108 of an automata-guided iterative partitioning procedure for the computation of the maximal

set of satisfying initial states (and corresponding control strategies) are shown in (b), (c), and (d), respectively (the sets of satisfying initial states are shown in yellow). For linear dynamics and a particular choice of polytope-to-polytope controllers, the procedure is complete (i.e., guaranteed to terminate and to find the maximal set). The yellow region from (d) is the maximal set of satisfying initial states. Sample trajectories of the closed loop system are shown in d, where the initial states are marked by circles. The trajectories coincide in the last six steps before they reach the target region. (Example adapted from Belta et al. 2017)



Formal Methods for Controlling Dynamical Systems, Fig. 3 For the system described in Fig. 2, in addition to satisfying the temporal logic specification, the system is required to minimize a quadratic cost that penalizes the Euclidean distance from desired state and control trajectories, which are available to the system over a short finite-time horizon N . The reference trajectory is shown in green (satisfying in the left and middle and violating in the right), and the trajectory of the controlled system is shown

in red. Pairs of points on the reference and controlled trajectory corresponding to the same time are connected. The left and middle cases correspond to increasing values of the horizon N for the same reference trajectory. Note that, in the situation shown in the right, where the reference trajectory violates the correctness specification, the controller “tries to compromise” between correctness and optimality. (Example adapted from Belta et al. 2017)

added (possibly weighted by a constant) to the cost defined above, the optimization problem remains a MILP (MIQP). Adding robustness in the cost allows to compromise between correctness and optimality (see Fig. 4 and its caption.)

The main advantage of MIP-based methods is the seamless combination of correctness and optimality, with the added feature of robustness to satisfaction. Another advantage is scalability. Such methods produced results for systems with hundreds of state variables in seconds. The main limitation of such methods is that they are constrained to linear dynamics.

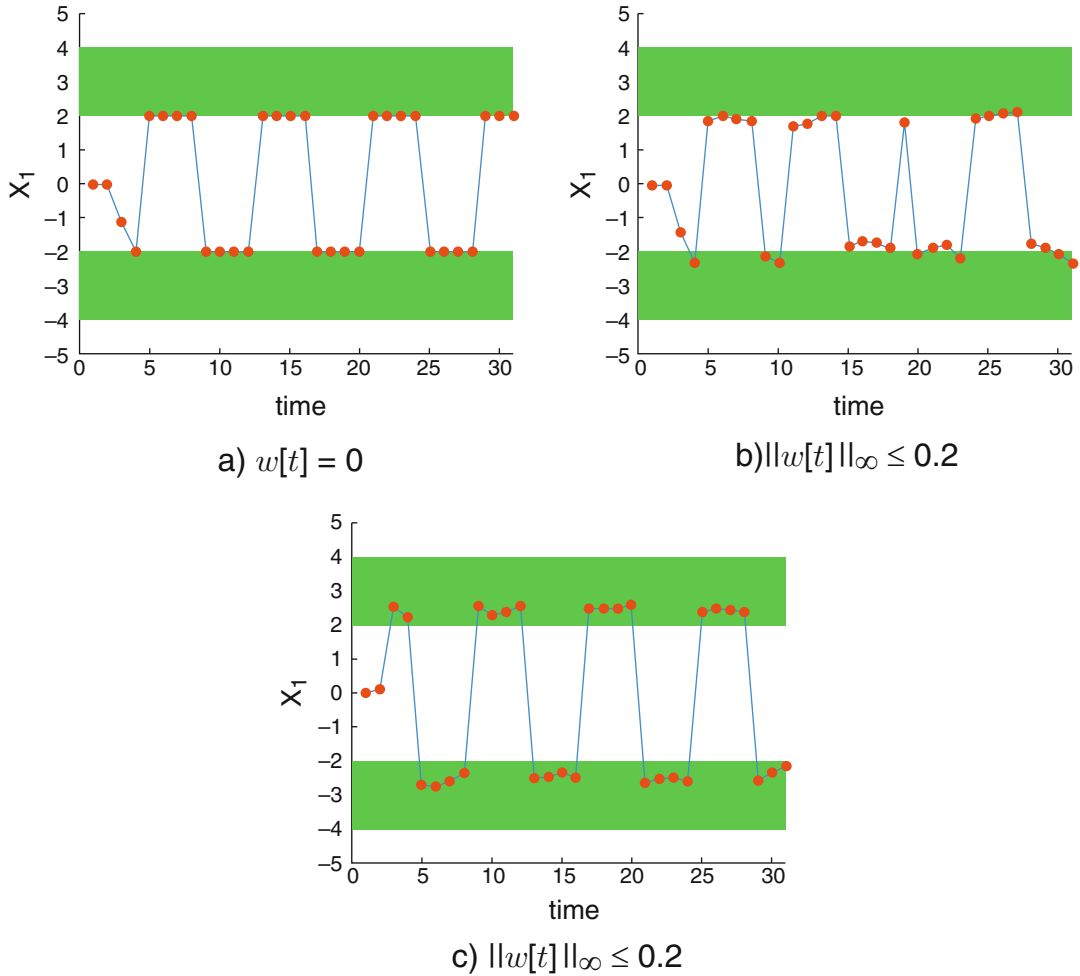
CBF-type methods address some of the limitations of the MIP-based methods. A pictorial representation of a CBF-based approach is shown in Fig. 5. In short, such methods work for a particular class of nonlinear control systems, called affine control systems, which is large enough to include many mechanical systems, such as unicycles, cars, etc., costs that are quadratic in controls, linear control constraints, and safety constraints expressed as set forward invariance. By dividing the time interval of interest into smaller time intervals, a nonlinear optimization problem can

be reduced to a set of quadratic programs (QP). Richer, temporal logic correctness constraints can also be incorporated in this framework.

Summary and Future Directions

While provably correct, automata-based approaches to formal synthesis of control strategies are computationally expensive. Current research that addresses this limitation is focused on identifying system properties that facilitate the construction of the abstraction (e.g., passivity, dissipativity) (Coogan and Arcak 2017) and compositional synthesis and assume guarantee-type approaches (Kim et al. 2017).

Optimization-based approaches based on mixed integer programming scale can quantify satisfaction and can compromise between correctness and optimality. However, they are restricted to linear dynamics. Optimization-based approaches based on control barrier functions and control Lyapunov functions can be used with nonlinear systems that are affine in controls. One of the main limitations of these types of approaches is that the optimization problems can



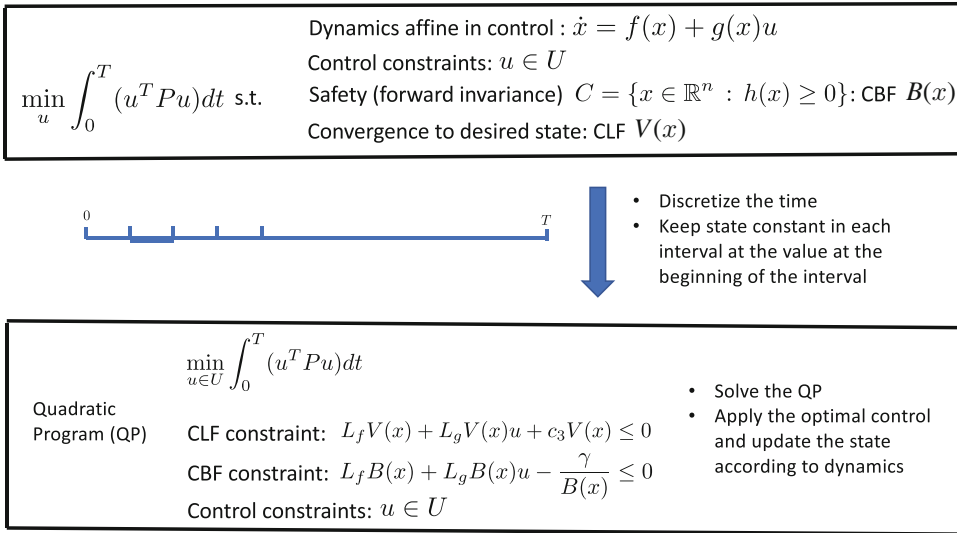
Formal Methods for Controlling Dynamical Systems,

Fig. 4 A planar discrete-time (integrator) linear system $x[t+1] = Ax[t] + Bu[t] + w[t]$ with state $x = (x_1, x_2)$, control u , $A = \begin{bmatrix} 1 & 0.5 \\ 0 & 0.8 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ is affected by noise w . The specification requires that x_1 oscillate between $2 \leq x_1 \leq 4$ and $-4 \leq x_1 \leq -2$, with each interval being visited at least once within any five consecutive time steps. The corresponding STL formula is $\mathbf{G}_{[0,\infty)}(\mathbf{F}_{[0,4]}((x_1 \geq 2) \wedge (x_1 \leq 4))) \wedge (\mathbf{F}_{[0,4]}((x_1 \geq -4) \wedge (x_1 \leq -2)))$. Here, $\mathbf{F}_{[t_1,t_2]}\psi$ requires that ψ is eventually satisfied within $[t_1, t_2]$, while $\mathbf{G}_{[t_1,t_2]}\psi$

means that ψ is true for all times in $[t_1, t_2]$. The control effort $\sum_{t=0}^{H-1} |u|^2$ is minimized in (a) and (b), while in (c) the “robust” version of the cost $\sum_{t=0}^{H-1} -M(\rho - |\rho|) + |u|^2$ is considered (H is a time horizon that is large enough to be able to decide the truth value of the formula). It can be seen that, if only the control effort is minimized, the resulting strategy is not robust to noise. If robustness is included in the cost, then the produced trajectory satisfies the specification even if noise is added to the system. (Example adapted from Sadraddini and Belta 2015)

easily become infeasible, especially when many constraints (state limitations, control constraints, CBF and CLF constraints) become active. One possible approach to this problem is to soften some constraints that are not critical, such as those induced by CLFs. Another possible

approach would be to use machine learning techniques to increase feasibility. For example, for a robot moving in an environment cluttered with obstacle, the configuration space can be sampled close to the obstacles, and the samples could be classifier depending on whether



Formal Methods for Controlling Dynamical Systems, Fig. 5 CBF-based approach to provably correct and optimal control synthesis: An affine control system is required to minimize a quadratic cost while converging to a desired final state and satisfying a safety specification expressed as the forward invariance of a set C and polyhedral control constraints U . Assuming that a CBF $B(x)$ can be constructed for C , then safety is guaranteed if the CBF constraint is satisfied. If a control Lyapunov function

(CLF) $V(x)$ can be constructed, then exponential convergence to the desired state is guaranteed if the CLF constraint is satisfied. By discretizing the time and keeping the state constant in each time interval, both constraints become linear in control, and the problem reduces to a set of QPs. L_f and L_g denote Lie derivatives along f and g , respectively. P is a positive definite matrix and c_3 and γ are positive constants

the corresponding QP is feasible or not. The (differentiable) classifier could be used as an extra CBF. Another active area of research is integrating correctness specifications given in rich, temporal logic specifications. Very recent results (Lindemann and Dimarogonas 2019) show that STL specifications can be enforced using CBF.

Cross-References

- [Discrete Event Systems and Hybrid Systems, Connections Between](#)
- [Motion Description Languages and Symbolic Control](#)
- [Supervisory Control of Discrete-Event Systems](#)
- [Validation and Verification Techniques and Tools](#)

Recommended Reading

Two popular textbooks on formal methods for finite systems are Baier et al. (2008) and Clarke et al. (1999). Research monographs and textbooks that cover abstraction-based methods include Lee and Seshia (2015), Alur (2015), Tabuada (2009) and Belta et al. (2017). The reader interested in the mixed integer programming approach to the optimization-based techniques is referred to Karaman et al. (2008), Raman et al. (2014) and Sadraddini and Belta (2015). The metrics with quantitative semantics that enable such techniques were introduced in Maler and Nickovic (2004) and Koymans (1990). Optimization-based techniques using control Lyapunov functions and control barrier functions are covered in Galloway et al. (2013) and Ames et al. (2014).

Bibliography

- Alur R (2015) Principles of cyber-physical systems. MIT Press, Cambridge
- Ames AD, Grizzle JW, Tabuada P (2014) Control barrier function based quadratic programs with application to adaptive cruise control. In: Proceedings of 53rd IEEE conference on decision and control, pp 6271–6278
- Baier C, Katoen JP, Others (2008) Principles of model checking, vol 26202649. MIT Press, Cambridge
- Belta C, Yordanov B, Gol EA (2017) Formal methods for discrete-time dynamical systems. Springer, Cham. ISBN 978-3-319-50762-0
- Clarke EM, Peled D, Grumberg O (1999) Model checking. MIT Press, Cambridge
- Coogan S, Arcak M (2017) Finite abstraction of mixed monotone systems with discrete and continuous inputs. *Nonlinear Anal Hybrid Syst* 23:254–271
- Galloway K, Sreenath K, Ames AD, Grizzle J (2013) Torque saturation in bipedal robotic walking through control lyapunov function based quadratic programs. preprint arXiv:13027314
- Karaman S, Sanfelice RG, Frazzoli E (2008) Optimal control of mixed logical dynamical systems with linear temporal logic specifications. In: 2008 47th IEEE conference on decision and control. IEEE, pp 2117–2122. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4739370>
- Kim ES, Sadraddini S, Belta C, Arcak M, Seshia SA (2017) Dynamic contracts for distributed temporal logic control of traffic networks. In: Proceedings of 56th IEEE conference on decision and control, Melbourne
- Koymans R (1990) Specifying real-time properties with metric temporal logic. *Real-Time Syst* 2(4):255–299
- Lee EA, Seshia SA (2015) Introduction to embedded systems: a cyber-physical systems approach, 2nd edn. <http://leeseshia.org>
- Lindemann L, Dimarogonas DV (2019) Control barrier functions for signal temporal logic tasks. *IEEE Control Syst Lett* 3(1):96–101. <https://doi.org/10.1109/LCSYS.2018.2853182>
- Maler O, Nickovic D (2004) Monitoring temporal properties of continuous signals. In: Formal techniques, modelling and analysis of timed and fault-tolerant systems. Springer, Berlin, pp 152–166
- Raman V, Donzé A, Maasoumy M, Murray RM, Sangiovanni-Vincentelli A, Seshia SA (2014) Model predictive control with signal temporal logic specifications. In: 53rd IEEE conference on decision and control. IEEE, pp 81–87
- Sadraddini S, Belta C (2015) Robust temporal logic model predictive control. In: 53rd annual allerton conference on communication, control, and computing, Urbana
- Tabuada P (2009) Verification and control of hybrid systems: a symbolic approach. Springer, Dordrecht
- Ulusoy A, Belta C (2014) Receding horizon temporal logic control in dynamic environments. *Int J Robot Res* 33(12):1593–1607

Frequency Domain System Identification

Johan Schoukens and Rik Pintelon
Department ELEC, Vrije Universiteit Brussel,
Brussels, Belgium

Abstract

In this chapter we give an introduction to frequency domain system identification. We start from the identification work loop in ► [System Identification: An Overview](#), Fig. 4, and we discuss the impact of selecting the time or frequency domain approach on each of the choices that are in this loop. Although there is a full theoretical equivalence between the time and frequency domain identification approach, it turns out that, from practical point of view, there can be a natural preference for one of both domains.

Keywords

Discrete and continuous time models · Experiment setup · Frequency and time domain identification · Plant and noise model

Introduction

System identification provides methods to build a mathematical model for a dynamical system starting from measured input and output signals (► [System Identification: An Overview](#); see the section “Models and System Identification”). Initially, the field was completely dominated by the time domain approach, and the frequency domain was used to interpret the results (Ljung

and Glover 1981). This picture changed in the nineteenth of the last century by the development of advanced frequency domain methods (Ljung 2006; Pintelon and Schoukens 2012), and nowadays it is widely accepted that there is a full theoretical equivalence between time and frequency domain system identification under some weak conditions (Agüero et al. 2009; Pintelon and Schoukens 2012). Dedicated toolboxes are available for both domains (Ljung 1988; Kollar 1994). This raises the question how to choose between time and frequency domain identification. Many times the choice between both approaches can be made based on the user's familiarity with one of two methods. However, for some problems it turns out that there is a natural preference for the time or the frequency domain. This contribution discusses the main issues that need to be considered when making this choice, and it provides additional insights to guide the reader to the best solutions for her/his problem.

In the identification work loop ► [System Identification: An Overview](#), Fig. 4, we need to address three important questions (Ljung 1999, Sect. 1.4; Söderström and Stoica 1989, Chap. 1; Pintelon and Schoukens 2012, Sect. 1.4) that directly interact with the choice between time and frequency domain system identification:

- What data are available? What data are needed? This discussion will influence the selection of the measurement setup, the model choice, and the design of the experiment.
- What kind of models will be used? We will mainly focus on the identification of discrete time and continuous time models, using exactly the same frequency domain tools.
- How will the model be matched to the data? This question boils down to the choice of a cost function that measures the distance between the data and the model. We will discuss the use of nonparametric weighting functions in the frequency domain.

In the next sections, we will address these and similar questions in more detail. First, we discuss

the measurement of the raw data. The choices that are made in this step will have a strong impact on many user aspects of the identification process. The frequency domain formulation will turn out to be a natural choice to propose a unified formulation of the system identification problem, including discrete and continuous time modeling. Next, a generalized frequency domain description of the system relation will be proposed. This model will be matched to the data, using a weighted least squares cost function, formulated in the frequency domain identification. This will allow for the use of nonparametric weighting functions, based on a nonparametric preprocessing of the data. Eventually, some remaining user aspects are discussed.

Data Collection

In this section, we discuss the measurement assumptions that are made when the raw data are collected. It will turn out that these will directly influence the natural choice of the models that are used to describe the continuous time physical systems.

Time Domain and Frequency Domain Measurements

The data can be collected either in the time or in the frequency domain, and we discuss briefly both options.

Time Domain Measurements

Most measurements are nowadays made in the time domain because very fast high-quality analog-to-digital convertors (ADC) became available at a low price. These allow us to sample and discretize the continuous time input and output signals and process these on a digital computer. Also the excitation signals are mostly generated from a discrete time sequence with a digital-to-analog convertor (DAC).

The continuous time input and output signals $u_c(t)$, $y_c(t)$ of the system to be modeled are measured at the sampling moments $t_k = kT_s$, with $T_s = 1/f_s$ the sampling period and

f_s the sampling frequency: $u(k) = u_c(kT_s)$, and $y(k) = y_c(kT_s)$. The discrete time signals $u(k), y(k)$, $k = 1, \dots, N$ are transformed to the frequency domain using the discrete Fourier transform (DFT) (► [Nonparametric Techniques in System Identification](#), Eq. 1), resulting in the DFT spectra $U(l), Y(l)$, at the frequencies $f_l = l \frac{f_s}{N}$. Making some abuse of notation, we will reuse the same symbols later in this text, to denote the Z-transform of the signals, for example, $Y(z)$ will also be used for the Z-transform of $y(k)$.

Frequency Domain Measurements

A major exception to this general trend toward time domain measurements are the (high-frequency) network analyzers that measure the transfer function of a system frequency by frequency, starting from the steady-state response to a sine excitation. The frequency is stepped over the frequency band of interest, resulting directly in a measured frequency response function at a user selected set of frequencies $\omega_k, k = 1, \dots, F$:

$$G(\omega_k).$$

From the identification point of view, we can easily fit the latter situation in the frequency domain identification framework, by putting

$$U(k) = 1, Y(k) = G(\omega_k).$$

For that reason we will focus completely on the time domain measurement approach in the remaining part of this contribution.

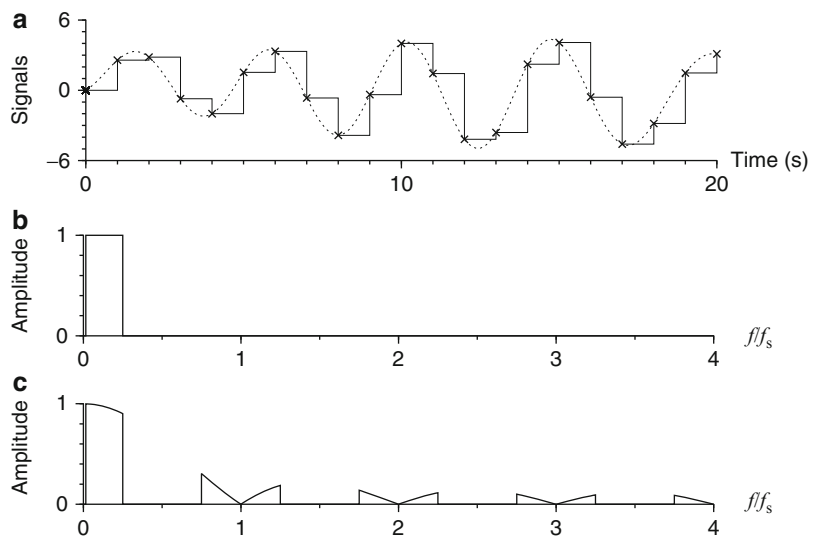
Zero-Order-Hold and Band-Limited Setup: Impact on the Model Choice

No information is available on how the continuous time signals $u_c(t), y_c(t)$ vary in between the measured samples $u(k), y(k)$. For that reason we need to make an assumption and make sure that the measurement setup is selected such that the intersample assumption is met. Two intersample assumptions are very popular (Schoukens et al. 1994; Pintelon and Schoukens 2012, pp. 498–512): the zero-order-hold (ZOH) and the band-limited assumption (BL). Both options are shown in Fig. 1 and discussed below. The choice of these assumptions does not only affect the measurement setup; it has also a strong impact on the selection between a continuous or a discrete time model choice.

Zero-Order Hold

The ZOH setup assumes that the excitation remains constant in between the samples. In practice, the model is identified between the discrete time reference signal in the memory

Frequency Domain System Identification, Fig. 1 Comparison of the ZOH and BL signal reconstruction of a discrete time sequence: (a) time domain: $\cdots$ BL, $-$ ZOH; (b) spectrum ZOH signal; (c) spectrum BL signal



of the generator and the sampled output. The intersample behavior is an intrinsic part of the model: if the intersample behavior changes, also the corresponding model will change. The ZOH assumption is very popular in digital control. In that case the sampling frequency f_s is commonly chosen 10 times larger than the frequency band of interest.

A discrete time model gives, in case of noise-free data, an exact description between the sampled input $u(k)$ and output $y(k)$ of the continuous time system:

$$y(k) = G(q, \theta)u(k)$$

in the time domain (► [System Identification: An Overview](#), Eq. 6). In this expression, q denotes the shift operator (time domain), and it is replaced by z in the z -domain description (transfer function description):

$$Y(z) = G(z, \theta)U(z).$$

Evaluating the transfer function at the unit circle by replacing $z = e^{i\omega}$ results in the frequency domain description of the system:

$$Y(e^{i\omega}) = G(e^{i\omega}, \theta)U(e^{i\omega})$$

(► [System Identification: An Overview](#), Eq. 34).

Band-Limited Setup

The BL setup assumes that above a given frequency $f_{\max} < f_s/2$, there is no power in the signals. The continuous time signals are filtered by well-tuned anti-alias filters (cutoff frequency $f_{\max} < f_s/2$), before they are sampled. Outside the digital control world, the BL setup is the standard choice for discrete time measurements. Without using anti-alias filters, large errors can be created due to aliasing effects: the high frequency ($f > f_s/2$) content of the measured signals is folded down in the frequency band of interest and act there as a disturbance. For that reason it is strongly advised to use always anti-alias filters in the measurement setup.

The exact relation between BL signals is described by a continuous time model, for

example, in the frequency domain:

$$Y(\omega) = G(\omega, \theta)U(\omega)$$

(Schoukens et al. 1994).

Combining Discrete Time Models and BL Data

It is also possible to identify a discrete time model between the BL data, at a cost of creating (very) small model errors (Schoukens et al. 1994). This is the standard setup that is used in digital signal procession applications like digital audio processing. The level of the model errors can be reduced by lowering the ratio $f_{\max}/f_s$ or by increasing the model order. In a properly designed setup, the discrete time model errors can be made very small, e.g., relative errors below 10^{-5} . In Table 1 an overview of the models corresponding to the experimental conditions is given.

Extracting Continuous Time Models from ZOH Data

Although the most robust and practical choice to identify a continuous time model is to start from BL data, it is also possible to extract a continuous time model under the ZOH setup. A first possibility is to assume that the ZOH assumption is perfectly met, which is a very hard assumption to realize in practice. In that case the continuous time model can be retrieved by a linear step invariant transformation of the discrete time model. A second possibility is to select a very high sample frequency with respect to the bandwidth of the system. In that case it is advantageous to describe the discrete time model using the delta operator (Goodwin 2010), and we have that the coefficients of the discrete time model converge to those of the continuous time model.

Models of Cascaded Systems

In some problems we want to build models for a cascade of two systems G_1, G_2 . It is well known that the overall transfer function G is given by the product $G(\omega) = G_1(\omega)G_2(\omega)$. This result holds also for models that are identified under the BL signal assumption: the model for the cascade

Frequency Domain System Identification, Table 1 Relations between the continuous time system $G(s)$ and the identified models as a function of the signal and model choices

	DT-model (Assuming ZOH-setup)	CT-model (Assuming BL-setup)
ZOH setup	Exact DT-model $G(z) = (1 - z^{-1}) Z \left\{ \frac{G(s)}{s} \right\}$ ‘standard conditions DT modelling’	Not studied
BL setup	Approximate DT model $\tilde{G}(z) \tilde{G}(z = e^{j\omega T_s}) \approx G(s = j\omega),$ $ \omega < \frac{\omega_s}{2}$ ‘digital signal processing field’	Exact CT-model $G(s)$ ‘standard conditions CT modelling’

F

will be the product of the models of the individual systems. However, the result does not hold under the ZOH assumption, because the intermediate signal between G_1, G_2 does not meet the ZOH assumption. For that reason, the ZOH model of a cascaded system is not obtained by cascading the ZOH models of the individual systems.

Experiment Design: Periodic or Random Excitations?

In general, arbitrary data can be used to identify a system as long as some basic requirements are respected (► [System Identification: An Overview](#), section on Experiment Design). Imposing periodic excitations can be an important restriction of the user’s freedom to design the experiment, but we will show in the next sections that it offers also major advantages at many steps in the identification work loop (► [System Identification: An Overview](#), Fig. 4).

With the availability of arbitrary wave form generators, it became possible to generate arbitrary periodic signals. The user should make two major choices during the design of the periodic excitation: the selection of the amplitude spectrum (How is the available power distributed over the frequency?) and the choice of the frequency resolution (What is the frequency step between two successive points of the measured FRF?) (Pintelon and Schoukens 2012, Sect. 5.3).

The amplitude spectrum is mainly set by the requirement that the excited frequency band should cover the frequency band of interest. A white noise excitation covers the full frequency

band, including those bands that are of no interest for the user. This is a waste of power and it should be avoided. Designing a good power spectrum for identification and control purposes is discussed in ► [Experiment Design and Identification for Control](#).

The frequency resolution $f_0 = 1/T$ is set by the inverse of the period of the signal. It should be small enough so that no important dynamics are missed, e.g., a very sharp mechanical resonance.

The reader should be aware that exactly the same choices have to be made during the design of nonperiodic excitations. If, for example, a random noise excitation is used, the frequency resolution is also restricted by the length of the experiment T_m and the corresponding frequency resolution is again $f_0 = 1/T_m$. The power spectrum of the noise excitation should be well shaped using a digital filter.

Nonparametric Preprocessing of the Data in the Frequency Domain

Before a parametric model is identified from the raw data, a lot of information can be gained, almost for free, by making a nonparametric analysis of the data. This can be done with very little user interaction. Some of these methods are explicitly linked to the periodic nature of the data, other methods apply also to random excitations.

Nonparametric Frequency Analysis of Periodic Data

By using simple DFT techniques, the following frequency domain information is extracted

from sampled time domain data $u(t), y(t), t = 1, \dots, N$ (Pintelon and Schoukens 2012):

- The signal information: $U(l), Y(l)$, the DFT spectra of the input and output, evaluated at the frequencies lf_0 , with $k = 1, 2, \dots, F$.
- Disturbing noise variance information: The full data record is split, so that each sub-record contains a single period. For each of these, the DFT spectrum is calculated. Since the signals are periodic, they do not vary from one period to the other, so that the observed variations can be attributed to the noise. By calculating the sample mean and variance over the periods at each frequency, a nonparametric noise analysis is available. The estimated variances $\hat{\sigma}_U^2(k), \hat{\sigma}_Y^2(k)$ measure the disturbing noise power spectrum at frequency f_k on the input and the output respectively. The covariance $\hat{\sigma}_{YU}^2(k)$ characterizes the linear relations between the noise on the input and the output.

This is very valuable information because, even before starting the parametric identification step, we get already full access to the quality of the raw data. As a consequence, there is also no interference between the plant model estimation and the noise analysis: plant model errors do not affect the estimated noise model. It is also important to realize that there is no user interaction requested to make this analysis and it follows directly from a simple DFT analysis of the raw data. These are two major advantages of using periodic excitations.

Nonlinear Analysis

Using well-designed periodic excitations, it is possible to detect the presence of nonlinear distortions during the nonparametric frequency step. The level of the nonlinear distortions at the output of the system is measured as a function of the frequency, and it is even possible to differentiate between even (e.g., x^2) and odd (e.g., x^3) distortions. While the first only act as disturbing noise in a linear modeling framework, the latter will also affect the linearized dynamics and can

change, for example, the pole positions of a system (Pintelon and Schoukens 2012, Sect. 4.3).

Noise and Data Reduction

By averaging the periodic signals over the successive periods, we get a first reduction of the noise. An additional noise reduction is possible when not all frequencies are excited. If a very wide frequency band has to be covered, a fine frequency resolution is needed at the low frequencies, whereas in the higher frequency bands, the resolution can be reduced. Signals with a logarithmic frequency distribution are used for that purpose. Eliminating the unexcited frequencies does not only reduce the noise, it also reduces significantly the amount of raw data to be processed. By combining different experiments that cover each a specific frequency band, it is possible to measure a system over multiple decades, e.g., electrical machines are measured from a few mHz to a few kHz.

In a similar way, it is also possible to focus the fit on the frequency band of interest by including only those frequencies in the parametric modeling step.

High-Quality Frequency Response Function Measurements

For periodic excitations, it is very simple to obtain high-quality measurements of the nonparametric frequency response function of the system. These results can be extended to random excitations at a cost of using more advanced algorithms that require more computation time (Pintelon and Schoukens, Chap. 7). This approach is discussed in detail in ► [Nonparametric Techniques in System Identification](#) (Eq. 1).

Generalized Frequency Domain Models

A very important step in the identification work loop is the choice of the model class (► [System Identification: An Overview](#), Fig. 4). Although most physical systems are continuous time, the models that we need might be either discrete time (e.g., digital control, computer simulations,

digital signal processing) or continuous time (e.g., physical interpretation of the model, analog control) (Ljung 1999, Sects. 2.1 and 4.3). A major advantage of the frequency domain is that both model classes are described by the same transfer function model. The only difference is the choice of the frequency variable. For continuous time models we operate in the Laplace domain, and the frequency variable is retrieved on the imaginary axis by putting $s = j\omega$. For discrete time systems we work on the unit circle that is described by the frequency variable $z = e^{j2\pi f/f_s}$. The application class can be even extended to include diffusion phenomena by putting $\Omega = \sqrt{j\omega}$ (Pintelon et al. 2005). From now on we will use the generalized frequency variable Ω , and depending on the selected domain, the proper substitution $j\omega$, $e^{j2\pi f/f_s}$, $\sqrt{j\omega}$ should be made. The unified discrete and continuous time description

$$Y(k) = G(\Omega_k, \theta)U(k).$$

illustrates also very nicely that in the frequency domain there is a strong similarity between discrete and continuous time system identification.

To apply this model to finite length measurements, it should be generalized to include the effect of the initial conditions (time domain) or the begin and end effects (called leakage in the frequency domain). See ► [Nonparametric Techniques in System Identification](#), “The Leakage Problem” section. The amazing result is that in the frequency domain, both effects are described

by exactly the same mathematical expression. This leads eventually to the following model in the frequency domain that is valid for periodic and arbitrary (nonperiodic) BL or ZOH excitations (Pintelon et al. 1997; McKelvey 2002; Pintelon and Schoukens 2012, Chap. 6):

$$Y(k) = G(\Omega, \theta)U(k) + T_G(\Omega, \theta)$$

which becomes for SISO (single-input-single-output) systems:

$$G(\Omega, \theta) = \frac{B(\Omega, \theta)}{A(\Omega, \theta)}, \text{ and } T_G(\Omega, \theta) = \frac{I(\Omega, \theta)}{A(\Omega, \theta)}.$$

A, B, I are all polynomials in Ω . The transient term $T_G(\Omega, \theta)$ models transient and leakage effects. It is most important for the reader to realize that this is an exact description for noise free data. Observe that it is very similar to the description in the time domain: $y(t) = G(q, \theta)u(t) + t_G(t, \theta)$. In that case the transient term $t_G(t, \theta)$ models the initial transient that is due to the initial conditions.

Parametric Identification

Once we have the data and the model available in the frequency domain, we define the following weighted least squares cost function to match the model to the data (Schoukens et al. 1997):

$$V(\theta) = \frac{1}{F} \sum_{k=1}^F \frac{|\hat{Y}(k) - G(\Omega_k, \theta)\hat{U}(k) - T_G(\Omega_k, \theta)|^2}{\hat{\sigma}_Y^2(k) + \hat{\sigma}_U^2(k)|G(\Omega_k, \theta)|^2 - 2\text{Re}(\hat{\sigma}_{YU}^2(k)G(\Omega_k, \theta))}.$$

The properties of this estimator are fully studied in Schoukens et al. (1999) and Pintelon and Schoukens (2012, Sect. 10.3), and it is shown that it is a consistent and almost efficient estimator under very mild conditions.

The formal link with the time domain cost function, as presented in ► [System Identification: An Overview](#), can be made by assuming that the

input is exactly known ($\hat{\sigma}_U^2(k) = 0$, $\hat{\sigma}_{YU}^2(k) = 0$), and replacing the nonparametric noise model on the output by a parametric model:

$$\hat{\sigma}_Y^2(k) = \lambda |H(\Omega_k, \theta)|^2.$$

These changes reduce the cost function, within a parameter independent constant λ to

$$V_F(\theta) = \frac{1}{F} \sum_{k=1}^F \frac{\left| \hat{Y}(k) - G(\Omega_k, \theta) U_0(k) - T_G(\Omega_k, \theta) \right|^2}{|H(\Omega_k, \theta)|^2},$$

which is exactly the same expression as Eq. 37 in [►System Identification: An Overview](#), provided that the frequencies Ω_k cover the full unit circle and the transient term T_G is omitted. The latter models the initial condition effects in the time domain. The expression shows the full equivalence with the classical discrete time domain formulation. If only a subsection of the full unit circle is used for the fit, the additional term $N \log \det \Lambda$ in [►System Identification: An Overview](#), Eq. 21 should be added to the cost function $V_F(\theta)$.

Additional User Aspects in Parametric Frequency Domain System Identification

In this section we highlight some additional user aspects that are affected by the choice for a time or frequency approach to system identification.

Nonparametric Noise Models

The use of a nonparametric noise model is a natural choice in the frequency domain. It is of course also possible to use the parametric noise model in the frequency domain formulation, but then we would lose two major advantages of the frequency domain formulation: (i) For periodic excitations, there is no interaction between the identification of the plant model and the nonparametric noise model. Plant model errors do not show up in the noise model. (ii) The availability of the nonparametric noise models eliminates the need for tuning the parametric noise model order, resulting in algorithms that are easier to use.

A disadvantage of using a nonparametric noise model is that we can no longer express that the noise model can share some dynamics with the plant model, for example, when the disturbance is an unobserved plant input.

It is also possible to use a nonparametric noise model in the time domain. This leads to a

Toeplitz weighting matrix, and the fast numerical algorithms that are used to deal with these make use internally of FFT (fast Fourier transform) algorithms which brings us back to the frequency domain representation of the data.

Stable and Unstable Plant Models

In the frequency domain formulation, there is no special precaution needed to deal with unstable models, so that we can tolerate these models without any problem. There are multiple reasons why this can be advantageous. The most obvious one is the identification of an unstable system, operating in a stabilizing closed loop. It can also happen that the intermediate models that are obtained during the optimization process are unstable. Imposing stability at each iteration can be too restrictive, resulting in estimates that are trapped in a local minimum. A last significant advantage is the possibility to split the identification problem (extract a model from the noisy data) and the approximation problem (approximate the unstable model by a stable one). This allows us to use in each step a cost function that is optimal for that step: maximum noise reduction in the first step, followed by a user-defined approximation criterion in the second step.

Model Selection and Validation

An important step in the system identification procedure is the tuning of the model complexity, followed by the evaluation of the model quality on a fresh data set. The availability of a nonparametric noise model and a high-quality frequency response function measurement simplifies these steps significantly:

Absolute Interpretation of the Cost Function:

Impact on Model Selection and Validation

The weighted least squares cost function, using the nonparametric noise weighting, is a normalized function. Its expected value equals

$E\{V(\hat{\theta})\} = (F - n_{\theta}/2)/F$, with n_{θ} the number of free parameters in the model and F the number of frequency points. A cost function that is far too large points to remaining model errors. Unmodeled dynamics result in correlated residuals (difference between the measured and the modeled FRF): the user should increase the model order to capture these dynamics in the linear model. A cost function that is far too large, while the residuals are white, points to the presence of nonlinear distortions: the best linear approximation is identified, but the user should be aware that this approximation is conditioned on the actual excitation signal. A cost function that is far too low points to an error in the preprocessing of the data, resulting in a bad noise model.

Missing Resonances

Some systems are lightly damped, resulting in a resonant behavior, for example, a vibrating mechanical structure. By comparing the parametric transfer function model with the nonparametric FRF measurements, it becomes clearly visible if a resonance is missed in the model. This can be either due to a too simple model structure (the model order should be increased), or it can appear because the model is trapped in a local minimum. In the latter case, better numerical optimization and initialization procedures should be looked for.

Identification in the Presence of Noise on the Input and Output Measurements

Within the band-limited measurement setup, both the input and the output have to be measured. This leads in general to an identification framework where both the input and the output are disturbed by noise. Such problems are studied in the errors-in-variables (EIV) framework (Soderstrom 2012). A special case is the identification of a system that is captured in a feedback loop. In that case we have that the noisy output measurements are fed back to the input of the system which creates a dependency between the input and output disturbance. We discuss both situations briefly below.

Errors-in-Variables Framework

The major difficulty of the EIV framework is the simultaneous identification of the plant model describing the input-output relations, the noise models that describe the input and the output noise disturbances, and the signal model describing the coloring of the excitation (Soderstrom 2012). Advanced identification methods are developed, but today it is still necessary to impose strong restrictions on the noise models, e.g., correlations between input and output noise disturbances are not allowed. The periodic frequency domain approach encapsulates the general EIV, including mutually correlated colored input-output noise. Again, a full nonparametric noise model is obtained in the preprocessing step. This reduces the complexity of the EIV problem to that of a classical weighted least squares identification problem which makes a huge difference in practice (Söderström et al. 2010; Pintelon and Schoukens 2012).

Identification in a Feedback Loop

Identification under feedback conditions can be solved in the time domain prediction error method (Ljung 1999, Sect. 13.4; Söderström and Stoica 1989, Chap. 10). This leads to consistent estimates, provided that the exact plant and noise model structure and order is retrieved. In the periodic frequency domain approach, a nonparametric noise model is extracted (variances input and output noise, and covariance between input and output noise) in the preprocessing step, without any user interaction. Next, these are used as a weighting in the weighted least squares cost function which leads to consistent estimates provided that the plant model is flexible enough to capture the true plant transfer function (Pintelon and Schoukens 2012, Sect. 9.18).

Summary and Future Directions

Theoretically, there is a full equivalence between the time and frequency domain formulation of the system identification problem. In many practical situations the user can make a free choice between both approaches, based on

nontechnical arguments like familiarity with one of both domains. However, some problems can be easier formulated in the frequency domain. Identification of continuous time models is not more involved than retrieving a discrete time model. The frequency domain formulation is also the natural choice to use nonparametric noise models. This eliminates the request to select a specific noise model structure and order, although this might be a drawback for experienced users who can take advantage of a clever choice of this structure. The advantages that are directly linked to periodic excitation signals can be explored most naturally in the frequency domain: the noise model is available for free, EIV identification is not more involved than the output error identification problem, and identification under feedback conditions does not differ from open-loop identification. Nonstationary effects are an example of a problem that will be easier detected in the time domain. In general, we advise the reader to take the best of both approaches and to swap from one domain to the other whenever it gives some advantage to do so. In the future, it will be necessary to extend the framework to include a characterization of nonlinear and time-varying effects.

Cross-References

- [Experiment Design and Identification for Control](#)
- [Nonparametric Techniques in System Identification](#)
- [System Identification: An Overview](#)

Recommended Reading

We recommend the reader the books of Ljung (1999) and Söderström and Stoica (1989) for a systematic study of time domain system identification. The book of Pintelon and Schoukens (2012) gives a comprehensive introduction to frequency domain identification. An extended discussion of the basic choices (intersample behavior, measurement setup) is given in Chap. 13 of

Pintelon and Schoukens (2012) or in Schoukens et al. (1994). The other references in this list highlight some of the technical aspects that were discussed in this text.

Acknowledgments This work was supported in part by the Fund for Scientific Research (FWO-Vlaanderen), by the Flemish Government (Methusalem), by the Belgian Government through the Inter university Poles of Attraction (IAP VII) Program, and the ERC Advanced Grant SNLSID.

Bibliography

- Agüero JC, Yuz JI, Goodwin GC et al (2009) On the equivalence of time and frequency domain maximum likelihood estimation. *Automatica* 46:260–270
- Goodwin GC (2010) Sampling and sampled-data models. In: American control conference, ACC 2010, Baltimore
- Kollar I (1994) Frequency domain system identification toolbox for use with MATLAB. The MathWorks Inc., Natick
- Ljung L (1988) System identification toolbox for use with MATLAB. The MathWorks Inc., Natick
- Ljung L (1999) System identification: theory for the user, 2nd edn. Prentice Hall, Upper Saddle River
- Ljung L (2006) Frequency domain versus time domain methods in system identification – revisited. In: Workshop on control of uncertain systems location, University of Cambridge, 21–22 Apr
- Ljung L, Glover K (1981) Frequency-domain versus time domain methods in system identification. *Automatica* 17:71–86
- McKelvey T (2002) Frequency domain identification methods. *Circuits Syst Signal Process* 21: 39–55
- Pintelon R, Schoukens J (2012) System identification: a frequency domain approach, 2nd edn. Wiley, Hoboken/IEEE, Piscataway
- Pintelon R, Schoukens J, Vandersteen G (1997) Frequency domain system identification using arbitrary signals. *IEEE Trans Autom Control* 42:1717–1720
- Pintelon R, Schoukens J, Pauwels L et al (2005) Diffusion systems: stability, modeling, and identification. *IEEE Trans Instrum Meas* 54:2061–2067
- Schoukens J, Pintelon R, Van hamme H (1994) Identification of linear dynamic systems using piecewise-constant excitations – use, misuse and alternatives. *Automatica* 30:1153–1169
- Schoukens J, Pintelon R, Vandersteen G et al (1997) Frequency-domain system identification using non-parametric noise models estimated from a small number of data sets. *Automatica* 33: 1073–1086
- Schoukens J, Pintelon R, Rolain Y (1999) Study of conditional ML estimators in time and frequency-domain system identification. *Automatica* 35:91–100

- Söderström T (2012) System identification for the errors-in-variables problem. *Trans Inst Meas Control* 34: 780–792
- Söderström T, Stoica P (1989) *System identification*. Prentice-Hall, Englewood Cliffs
- Söderström T, Hong M, Schoukens J et al (2010) Accuracy analysis of time domain maximum likelihood method and sample maximum likelihood method for errors-in-variables and output error identification. *Automatica* 46:721–727

Frequency-Response and Frequency-Domain Models

Abbas Emami-Naeini¹, Christina M. Ivler², and J. David Powell³

¹Electrical Engineering Department, Stanford University, Stanford, CA, USA

²Shiley School of Engineering, University of Portland, Portland, OR, USA

³Aero/Astro Department, Stanford University, Stanford, CA, USA

Abstract

A major advantage of using frequency response is the ease with which experimental information can be used for design purposes. Raw measurements of the output amplitude and phase of a plant undergoing a sinusoidal input excitation are sufficient to design a suitable feedback control. No intermediate processing of the data (such as finding poles and zeros or determining system matrices) is required to arrive at the system model. The wide availability of computational and system identification software tools has rendered this advantage less important now than it was years ago; however, for relatively simple systems, frequency response is often still the most cost-effective design method. The method is most effective for systems that are stable in open loop but can also be applied to unstable systems. Yet another advantage is that it is the easiest method to use for designing dynamic compensation as seen in the **Cross-Reference** article: [1] ▶ “Classical Frequency-Domain Design Methods”.

Keywords

Frequency response · Magnitude · Phase · Bode plot · Stability · Bandwidth · Resonant peak · Phase margin (PM) · Gain margin (GM) · Disturbance rejection bandwidth

Introduction

A very common way to use the exponential response of linear time-invariant systems (LTIs) is in finding the **frequency response**, or response to a sinusoid. First we express the sinusoid as a sum of two exponential expressions (Euler's relation):

$$A \cos(\omega t) = \frac{A}{2} (e^{j\omega t} + e^{-j\omega t}). \quad (1)$$

Suppose we have an LTI system with input u and output y , where we define the transfer function

$$G(s) = \frac{Y(s)}{U(s)}. \quad (2)$$

If we let $s = j\omega$ in $G(s)$, then the response to $u(t) = e^{j\omega t}$ is $y(t) = G(j\omega)e^{j\omega t}$; similarly, the response to $u(t) = e^{-j\omega t}$ is $G(-j\omega)e^{-j\omega t}$. By superposition, the response to the sum of these two exponentials, which make up the cosine signal, is the sum of the responses:

$$y(t) = \frac{A}{2} [G(j\omega)e^{j\omega t} + G(-j\omega)e^{-j\omega t}]. \quad (3)$$

The transfer function $G(j\omega)$ is a complex number that can be represented in polar form or in magnitude-and-phase form as $G(j\omega) = M(\omega)e^{j\varphi(\omega)}$ or simply $G = Me^{j\varphi}$. With this substitution, Eq. (3) becomes for a specific input frequency $\omega = \omega_o$

$$\begin{aligned} y(t) &= \frac{A}{2} M \left(e^{j(\omega_o t + \varphi)} + e^{-j(\omega_o t + \varphi)} \right), \\ &= AM \cos(\omega t + \varphi), \\ M &= |G(j\omega)| = |G(s)|_{s=j\omega_o} \end{aligned} \quad (4)$$

$$= \sqrt{\{\operatorname{Re}[G(j\omega_o)]\}^2 + \{\operatorname{Im}[G(j\omega_o)]\}^2},$$

$$\varphi = \angle G(j\omega) = \tan^{-1} \left[\frac{\operatorname{Im}[G(j\omega_o)]}{\operatorname{Re}[G(j\omega_o)]} \right]. \quad (5)$$

This means that if an LTI system represented by the transfer function $G(s)$ has a sinusoidal input with magnitude A , the output will be sinusoidal at the *same* frequency with magnitude AM and will be shifted in phase by the angle φ . M is usually referred to as the **amplitude ratio** or **magnitude**, and φ is referred to as the **phase**, and they are both functions of the input frequency, ω . The frequency response can be measured experimentally quite easily in the laboratory by driving the system with a known sinusoidal input, letting the transient response die, and measuring the steady-state amplitude and phase of the system's output as shown in Fig. 1. The input frequency is set to sufficiently many values so that curves such as the one in Fig. 2 are obtained. Bode suggested that we plot $\log |M|$ vs $\log \omega$ and $\varphi(\omega)$ vs $\log \omega$ to best show the essential features of $G(j\omega)$. Hence such plots are referred to as Bode plots. Bode plotting techniques are discussed in Franklin et al. (2019).

We are interested in analyzing the frequency response not only because it will help us understand how a system responds to a sinusoidal input but also because evaluating $G(s)$ with s taking on values along the $j\omega$ axis will prove to be very useful in determining the stability of a closed-loop system. Evaluating $G(j\omega)$ provides

information that allows us to determine closed-loop stability from the open-loop $G(s)$.

For the standard second-order system

$$G(s) = \frac{1}{(s/\omega_n)^2 + 2\zeta(s/\omega_n) + 1}, \quad (6)$$

the Bode plot is shown in Fig. 3 for various values of ζ .

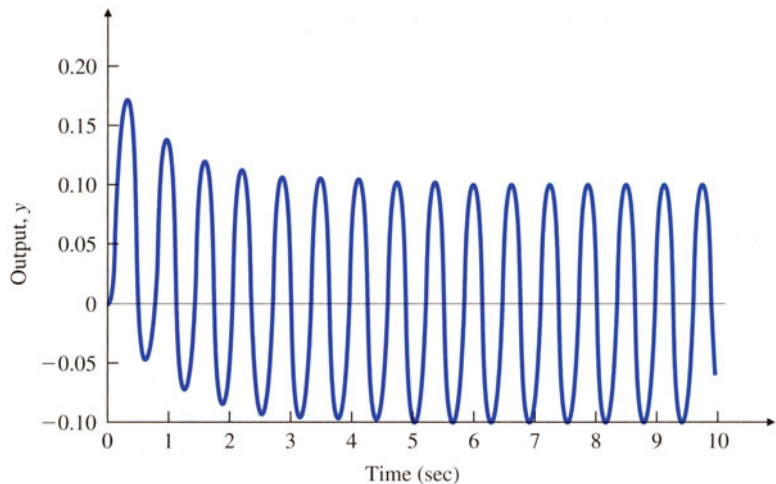
A natural specification for system performance in terms of frequency response is the **bandwidth**, defined to be the maximum frequency at which the output of a system will track an input sinusoid in a satisfactory manner. By convention, for the system shown in Fig. 4 with a sinusoidal input R , the bandwidth is the frequency of R at which the output Y is attenuated to a factor of 0.707 times the input (If the output is a voltage across a $1\text{-}\Omega$ resistor, the power is v^2 and when $|v| = 0.707$, the power is reduced by a factor of 2. By convention, this is called the half-power point.). Figure 5 depicts the idea graphically for the frequency response of the *closed-loop* transfer function

$$\frac{Y(s)}{R(s)} \triangleq T(s) = \frac{KG(s)}{1 + KG(s)}. \quad (7)$$

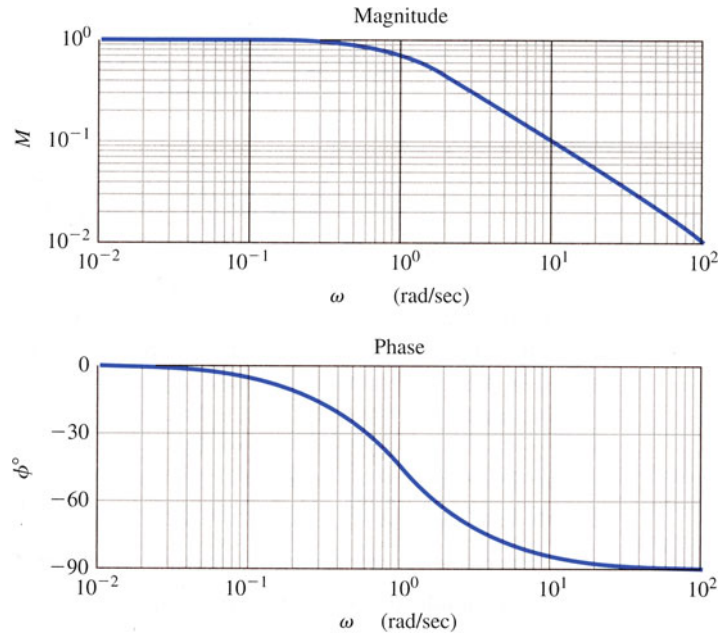
The plot is typical of most closed-loop systems in that (1) the output follows the input ($|T| \cong 1$) at the lower excitation frequencies and (2) the output ceases to follow the input ($|T| < 1$) at

Frequency-Response and Frequency-Domain Models, Fig. 1

Response of $G(s) = \frac{1}{(s+1)}$ to the input $u = \sin 10t$. (Source: Franklin, Powell, Emami-Naeini, Feedback Control of Dynamic Systems, 8th Ed., © 2019, p-334, Reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)



Frequency-Response and Frequency-Domain Models, Fig. 2 Frequency response for $G(s) = \frac{1}{s+1}$. (Source: Franklin, Powell, Emami-Naeini, Feedback Control of Dynamic Systems, 8th Ed., © 2019, p-102, Reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)



the higher excitation frequencies. The maximum value of the frequency-response magnitude is referred to as the **resonant peak** M_r .

Bandwidth is a measure of speed of response and is therefore similar to time-domain measures such as rise time and peak time or the s -plane measure of dominant root(s) natural frequency. In fact, if the $K G(s)$ in Fig. 4 is such that the closed-loop response is given by Fig. 3a, we can see that the bandwidth will equal the natural frequency of the closed-loop root (i.e., $\omega_{BW} = \omega_n$ for a closed-loop damping ratio of $\zeta = 0.7$). For other damping ratios, the bandwidth is approximately equal to the natural frequency of the closed-loop roots, with an error typically less than a factor of 2.

For a second-order system, the time responses are functions of the pole-location parameters ζ and ω_n . If we consider the curve for $\zeta = 0.5$ to be an average, the rise time (t_r) from $y = 0.1$ to $y = 0.9$ is approximately $\omega_n t_r = 1.8$. Thus we can say that

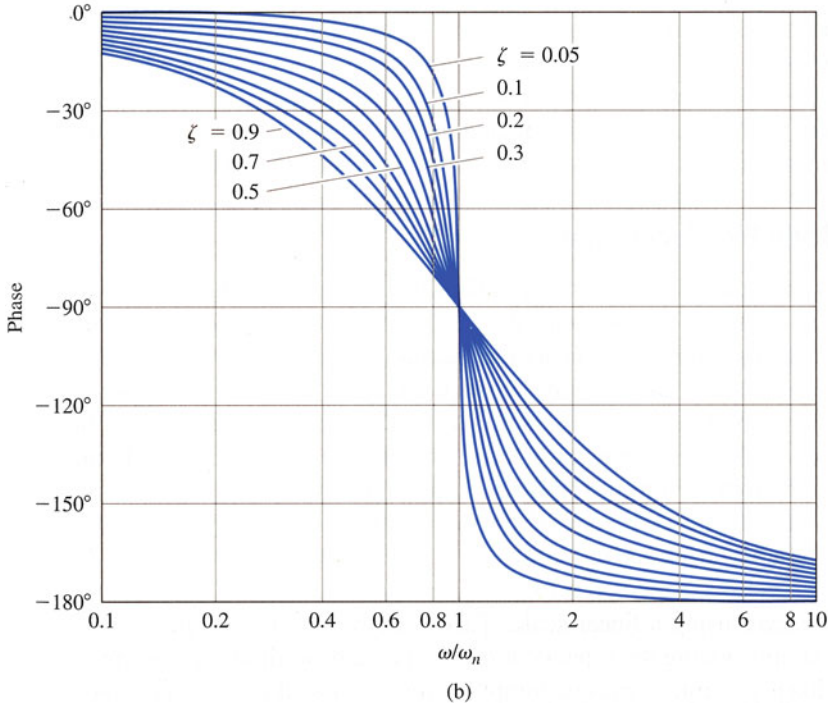
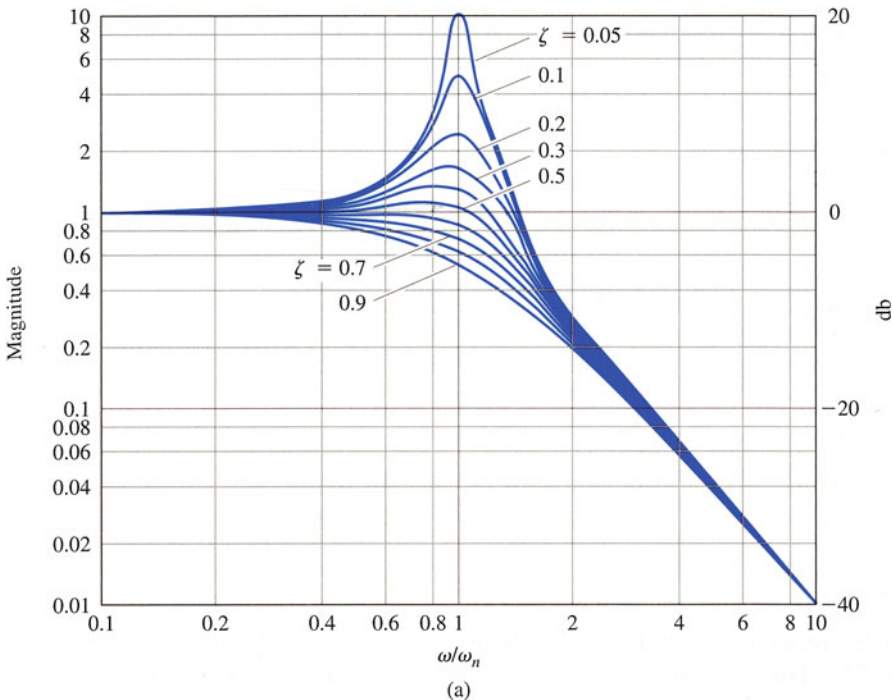
$$t_r \approx \frac{1.8}{\omega_n}. \quad (8)$$

Although this relationship could be embellished by including the effect of the damping ratio, it is important to keep in mind how Eq. (8) is

typically used. It is accurate only for a second-order system with no zeros; for all other systems, it is a rough approximation to the relationship between t_r and ω_n . Most systems being analyzed for control systems design are more complicated than the pure second-order system, so designers use Eq. (8) with the knowledge that it is a rough approximation only. For a second-order system, the bandwidth is inversely proportional to the rise time, t_r . Hence we are able to link the time and frequency domain quantities in this way.

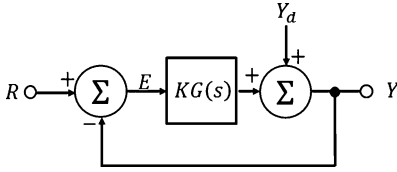
The definition of the bandwidth stated here is meaningful for systems that have a low-pass filter behavior, as is the case for any physical control system. In other applications the bandwidth may be defined differently. Also, if the ideal model of the system does not have a high-frequency roll-off (e.g., if it has an equal number of poles and zeros), the bandwidth is infinite; however, this does not occur in nature as physical systems don't respond well at infinite frequencies.

In many cases, the designer's primary concern is the error in the system due to disturbances rather than the ability to track an input. For error analysis, we are more interested in the sensitivity function, $S(s) = 1 - T(s)$, rather than $T(s)$. In a feedback control system, as in Fig. 4, the



Frequency-Response and Frequency-Domain Models, Fig. 3 Frequency responses of standard second-order systems (a) magnitude, (b) phase. (Source: Franklin, Pow-

ell, Emami-Naeini, Feedback Control of Dynamic Systems, 8th Ed., © 2019, p-338, Reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)



Frequency-Response and Frequency-Domain Models, Fig. 4 Unity feedback control system

sensitivity can be interpreted as either the error response to a reference input or the system response to an output disturbance:

$$S(s) = \frac{E(s)}{R(s)} = \frac{Y(s)}{Y_d(s)} = \frac{1}{1 + KG(s)}.$$

For most open-loop systems with high gain at low frequencies, $S(s)$ for a disturbance input has very low values at low frequencies, indicating an ability to attenuate low-frequency disturbances. The response to the input or disturbance grows as the frequency of the input or disturbance approaches ω_{DRB} . The **disturbance rejection bandwidth** ω_{DRB} characterizes the speed of the recovery from a disturbance, describing the frequency where the response to a disturbance has been reduced to 0.707 of the input disturbance, defined by Fig. 5. As such, the time to recover from a disturbance is inversely proportional to ω_{DRB} . Considering that $T(s) + S(s) = 1$, disturbances are attenuated so that $S(s) \rightarrow 0$ when the output tracks the input $T(s) \rightarrow 1$, as illustrated in Fig. 5. Therefore, an increase in bandwidth also increases the disturbance rejection bandwidth and typically $\omega_{DRB} < \omega_{BW}$. The maximum value of $S(s)$ is referred to as the disturbance rejection peak, M_d , which represents the maximum possible overshoot during a recovery from a disturbance.

For analysis of either $T(s)$ or $S(s)$, it is typical to plot their response versus the frequency of the input as shown in Fig. 5. Frequency response for control system design can either be evaluated using the computer, or it can be quickly sketched for simple systems using the efficient methods described in Franklin et al. (2019). The methods described next are also useful to expedite the

design process as well as to perform sanity checks on the computer output.

Neutral Stability: Gain and Phase Margins

In the early days of electronic communications, most instruments were judged in terms of their frequency response. It is therefore natural that when the feedback amplifier was introduced, techniques to determine stability in the presence of feedback were based on this response.

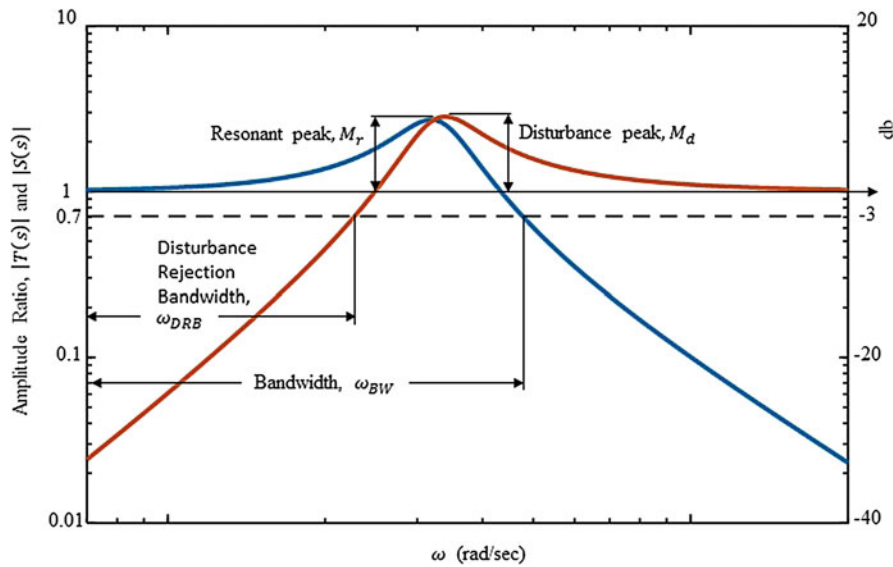
Suppose the closed-loop transfer function of a system is known. We can determine the stability of a system by simply inspecting the denominator in factored form (because the factors give the system roots directly) to observe whether the real parts are positive or negative. However, the closed-loop transfer function is usually not known; in fact, the whole purpose behind understanding the root-locus technique (see Franklin et al. 2019) is to be able to find the factors of the denominator in the closed-loop transfer function, given only the open-loop transfer function. Another way to determine closed-loop stability is to evaluate the frequency response of the *open-loop* transfer function $KG(j\omega)$ and then perform a test on that response. Note that this method also does not require factoring the denominator of the closed-loop transfer function. In this section we will explain the principles of this method.

Suppose we have a system defined by Fig. 6a and whose root locus behaves as shown in Fig. 6b; that is, instability results if K is larger than 2. The neutrally stable points lie on the imaginary axis – that is, where $K = 2$ and $s = j1.0$. Furthermore, all points on the root locus have the property that

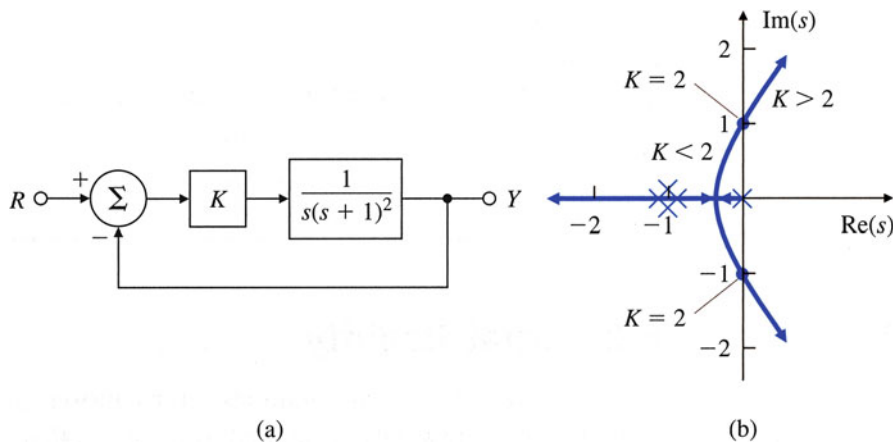
$$|KG(s)| = 1 \quad \text{and} \quad \angle G(s) = -180^\circ.$$

At the point of neutral stability, we see that these root-locus conditions hold for $s = j\omega$, so

$$|KG(j\omega)| = 1 \quad \text{and} \quad \angle G(j\omega) = -180^\circ. \quad (9)$$



Frequency-Response and Frequency-Domain Models, Fig. 5 Definitions of bandwidth and disturbance rejection bandwidth, resonant peak, and disturbance peak. ($T(s)$ is blue, $S(s)$ is red)

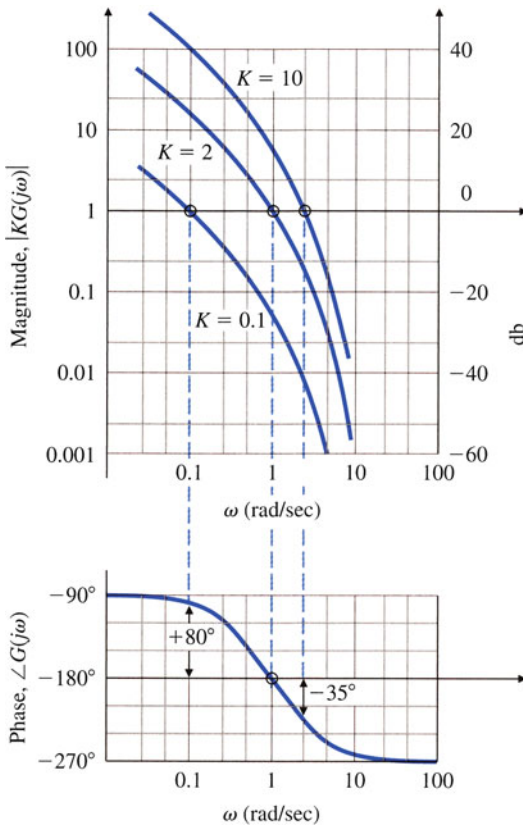


Frequency-Response and Frequency-Domain Models, Fig. 6 Stability example: (a) system definition; (b) root locus. (Source: Franklin, Powell, Emami-Naeini,

Feedback Control of Dynamic Systems, 8th Ed., © 2019, p-355, Reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)

Thus a Bode plot of a system that is neutrally stable (i.e., with K defined such that a closed-loop root falls on the imaginary axis) will satisfy the conditions of Eq. (9). Figure 7 shows the frequency response for the system whose root locus is plotted in Fig. 6 for various values of K . The magnitude response corresponding to $K = 2$ passes through 1 at the same frequency ($\omega = 1$ rad/sec) at which the phase passes through -180° , as predicted by Eq. (9).

Having determined the point of neutral stability, we turn to a key question: Does increasing the gain increase or decrease the system's stability? We can see from the root locus in Fig. 6b that any value of K less than the value at the neutrally stable point will result in a stable system. At the frequency ω where the phase $\angle G(j\omega) = -180^\circ$ ($\omega = 1$ rad/sec), the magnitude $|KG(j\omega)| < 1.0$ for stable values of K and > 1 for unstable values of K . Therefore, we have the following



Frequency-Response and Frequency-Domain Models, Fig. 7 Frequency-response magnitude and phase for the system in Fig. 6. (Source: Franklin, Powell, Emami-Naeini, *Feedback Control of Dynamic Systems*, 8th Ed., © 2019, p-356, Reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)

trial stability condition, based on the character of the open-loop frequency response:

$$|KG(j\omega)| < 1 \quad \text{at} \quad \angle G(j\omega) = -180^\circ. \quad (10)$$

This stability criterion holds for all systems for which increasing gain leads to instability and $|KG(j\omega)|$ crosses the magnitude ($=1$) once, the most common situation. However, there are systems for which an increasing gain can lead from instability to stability; in this case, the stability condition is

$$|KG(j\omega)| > 1 \quad \text{at} \quad \angle G(j\omega) = -180^\circ. \quad (11)$$

Based on the above ideas, we can now define the robustness metrics gain and phase margins:

Phase Margin Suppose at ω_1 , $|G(j\omega_1)| = \frac{1}{K}$. How much more *phase* could the system tolerate (as a time delay, perhaps) before reaching the stability boundary? The answer to this question follows from Eq. (9), i.e., the phase margin (PM) is defined as

$$\text{PM} = \angle G(j\omega_1) - (-180^\circ). \quad (12)$$

Gain Margin Suppose at ω_2 , $\angle G(j\omega_2) = -180^\circ$. How much more *gain* could the system tolerate (as an amplifier, perhaps) before reaching the stability boundary? The answer to this question follows from Eq. (10), i.e., the gain margin (GM) is defined as

$$\text{GM} = \frac{1}{K|G(j\omega_2)|}. \quad (13)$$

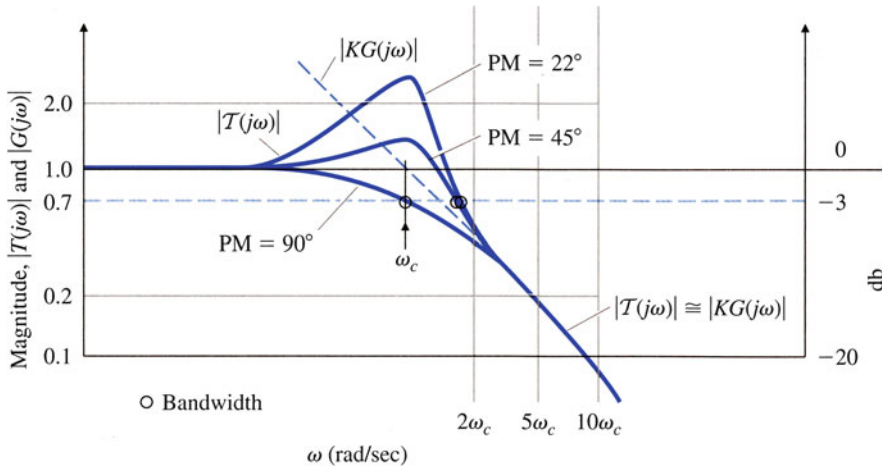
There are also rare cases when $|KG(j\omega)|$ crosses magnitude ($=1$) more than once or where an increasing gain leads to instability. A rigorous way to resolve these situations is to use the Nyquist stability criterion as discussed in Franklin et al. (2019).

Closed-Loop Frequency Response

The closed-loop bandwidth was defined above. The natural frequency is typically within a factor of two of the bandwidth for a second-order system. We can help establish a more exact correspondence by making a few observations. Consider a system in which $|KG(j\omega)|$ shows the typical behavior

$$\begin{aligned} |KG(j\omega)| &\gg 1 \quad \text{for} \quad \omega \ll \omega_c, \\ |KG(j\omega)| &\ll 1 \quad \text{for} \quad \omega \gg \omega_c, \end{aligned}$$

where ω_c is the crossover frequency where $|G(j\omega)| = 1$. The closed-loop frequency-response magnitude is approximated by



Frequency-Response and Frequency-Domain Models, Fig. 8 Closed-loop bandwidth with respect to PM. (Source: Franklin, Powell, Emami-Naeini, Feedback Con-

trol of Dynamic Systems, 8th Ed., © 2019, p-386, Reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)

$$|T(j\omega)| = \left| \frac{KG(j\omega)}{1 + KG(j\omega)} \right| \cong \begin{cases} 1, & \omega \ll \omega_c, \\ |KG|, & \omega \gg \omega_c. \end{cases} \quad (14)$$

In the vicinity of crossover, where $|KG(j\omega)| = 1$, $|T(j\omega)|$ depends heavily on the PM. A PM of 90° means that $\angle G(j\omega_c) = -90^\circ$, and therefore $|T(j\omega_c)| = 0.707$. On the other hand, PM = 45° yields $|T(j\omega_c)| = 1.31$.

The exact evaluation of Eq. (14) was used to generate the curves of $|T(j\omega)|$ in Fig. 8. It shows that the bandwidth for smaller values of PM is typically somewhat greater than ω_c , though usually it is less than $2\omega_c$; thus

$$\omega_c \leq \omega_{BW} \leq 2\omega_c. \quad (15)$$

Another specification related to the closed-loop frequency response is the resonant-peak magnitude M_r , defined in Fig. 5. For linear systems, M_r is generally related to the damping of the system. In practice, M_r is rarely used; most designers prefer to use the PM to specify the damping of a system, because the imperfections that make systems nonlinear or cause delays usually erode the phase more significantly than the magnitude.

It is also important in the design to achieve certain error characteristics, and these are often evaluated as a function of the input or disturbance

frequency. In some cases, the primary function of the control system is to regulate the output to a certain constant value in the presence of disturbances. For these situations, the key item of interest for the design would be the closed-loop frequency response of the error with respect to disturbance inputs and associated disturbance rejection bandwidth ω_{DRB} as well as the disturbance rejection peak M_d defined in Fig. 5.

Summary and Future Directions

The frequency response methods are the most popular because they easily account for model uncertainty and can be measured in the laboratory. A wide range of information about the system can be displayed in a Bode plot. The dynamic compensation can be carried out directly from the Bode plot. Extension of the ideas to multivariable systems has been done via singular value plots. Extension to nonlinear systems is still the subject of current research.

Cross-References

- [Classical Frequency-Domain Design Methods](#)
- [Control System Optimization Methods in the Frequency Domain](#)

- [Polynomial/Algebraic Design Methods](#)
- [Quantitative Feedback Theory](#)
- [Spectral Factorization](#)

Bibliography

Franklin GF, Powell JD, Emami-Naeini A (2019) Feed-back control of dynamic systems, 8th edn. Pearson Education, New York

FTC

- [Fault-Tolerant Control](#)

Fuel Cell Vehicle Optimization and Control

Miriam A. Figueroa-Santos and Anna G. Stefanopoulou
University of Michigan, Ann Arbor, MI, USA

Abstract

Fuel cell (FC) systems with onboard hydrogen storage offer long-range, fast refueling, and high power capability. These attributes make FC vehicles (FCVs) the best option for shared mobility and fleet vehicles with high utilization. A FC system consists of the stack, which performs the electrical conversion of hydrogen to electricity, the balance of plant (BOP) components including pumps and blowers which are responsible for supplying reactants (hydrogen and air) at the correct rates, humidification hardware, and thermal management. FCVs are typically hybridized with a battery to gain the regenerative energy and to improve the system durability by reducing the (a) transient and high current spikes, (b) idling at high open circuit potential, and (c) the number of startup and shutdown cycles. At the vehicle level, satisfying driver's torque/power demand is achieved by choosing the power split between the fuel cell and

battery. Additional improvements in fuel consumption can be achieved through preview or simultaneous optimization of vehicle speed and power split to regulate battery state of charge and fuel cell thermal management. In this entry, different optimal control strategies are reviewed and compared.

Keywords

Fuel cell vehicles · Rule-based strategies · Optimal control strategies · Fuel cell system · Balance of plant components

F

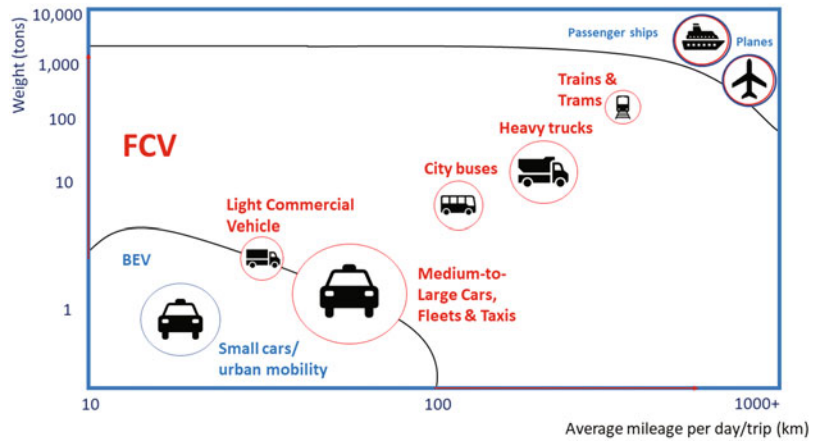
Introduction

Alternatives to the traditional internal combustion engine (ICE) are required to decarbonize the transportation, which accounts for almost 30% of all greenhouse gas emissions in the USA (Tanc et al. 2019) and nearly 20% in the rest of the world. Fuel cell electric vehicles (FCVs) are examples of the alternative to internal combustion engine vehicles (ICE) that are predominantly powered by fossil fuel. Fuel cell vehicles are mainly driven by fuel cells which convert chemical energy contained in hydrogen into electricity, providing a hybrid solution that combines the advantages of internal combustion engines and batteries (O'Hayre et al. 2016). Widespread adoption of FCVs is limited by a lack of hydrogen distribution and refueling infrastructure and high technology cost. Furthermore, the low volumetric energy density of compressed hydrogen storage onboard the vehicle limits the range to approximately 500 miles (Salahuddin et al. 2018). The main limiting factor in hydrogen production via hydrolysis of water is the amount of electricity required. If all petrol cars in the USA were replaced by fuel cell vehicles, an estimated 1.4–2.1 TW would be required to produce the needed hydrogen. This is far greater than the current total US electricity generating capacity of 1.2 TW (Chapman et al. 2019).

The current state of the art for light-duty fuel cell vehicles is the Toyota Mirai, Honda Clarity, and Hyundai Nexa. They have a range of 350 miles and can be refueled in 3–5 min (James 2020). A popular alternative vehicle con-

Fuel Cell Vehicle Optimization and Control, Fig. 1

Fuel cell vehicle (FCV) in the transportation sector. (Figure adapted from McK 2017.) Circle size represents the relative annual energy consumption of this vehicle type



figuration that will rely on renewable electric energy generation is the battery electric vehicle (BEV). Examples of the state of the art for BEVs are currently the Tesla Model 3 and S, Nissan Leaf, among others, which have a range of around 107–315 miles and recharge in 60–75 min. The BEVs' limitation compared to FCVs is their longer charging time and shorter mileage range, but they have lower cost, and their fuel infrastructure has been well established compared to the FCVs (Salahuddin et al. 2018). As Fig. 1 summarizes, most of the small cars will be BEVs, except robotic or autonomous vehicles that can and should operate with much higher utilization to become cost-effective, requiring multiple fast refueling events per day (James 2020). Applications such as shared-services, taxis, and buses, among others, can benefit by using fuel cell vehicles. Medium-duty vehicles (MDV) that operate continuously and in a fleet configuration with depot refueling would also benefit from transitioning to FC powertrains.

While fuel cell light-duty vehicles (LDVs) are commercially produced (though not widespread success), the development of heavy-duty fuel cell vehicles in long-haul trucks is accelerating. The US Department of Energy (DOE) in their efforts to implement FCVs determined a target of \$60/kW, a power density of 1.7 kWh/L, and a durability target of 30,000 h for heavy-duty vehicles (James 2020). They have expanded the focus towards heavy-duty applications due to

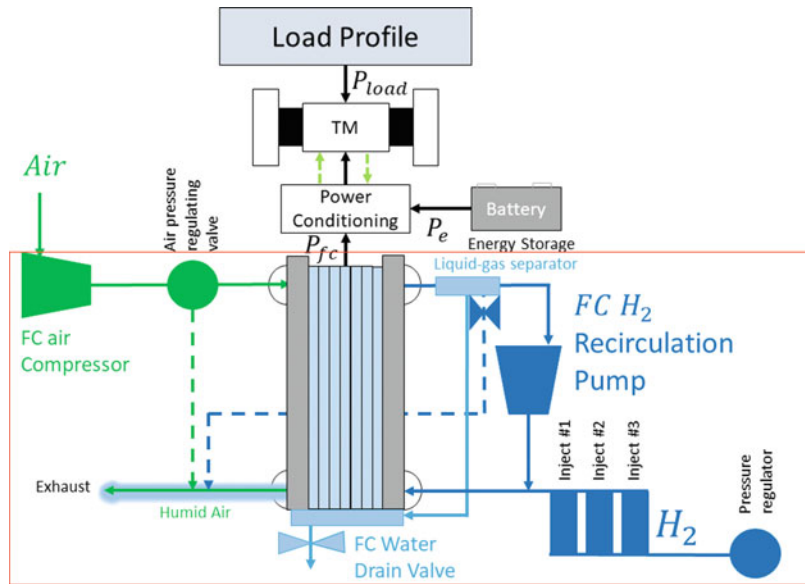
the stringent durability requirements compared to light-duty vehicle applications.

The durability and cost of the fuel cell system can be improved through optimization of the balance of plant (BOP) components such as air compressors, fuel processors, sensors, and water and heat management that affect fuel cell's performance. Great progress has been made toward achieving the Department of Energy's (DOEs) targets for reduction in stack component cost, including reducing the Pt in the catalyst. However, this attention has not been directed toward the BOP components, which is projected to reach 60% of the total cost by 2025 (James et al. 2018). Therefore, optimal control of the power split and the pumps, fans, and blowers would enable component downsizing and system simplification but may require careful coordination to protect the fuel cell from overheating or reactant starvation. This is important since degradation and hybridization affect the total cost of ownership (Lü et al. 2020).

Fuel cells are combined with another energy source as shown in Fig. 2 to meet stringent vehicle cold startup and transient response requirements and improve system durability (Liv 2011). Fuel cell and supercapacitor configurations for FCVs were investigated due to the high power density, fast transient response, longer life cycle, and high-frequency rate of charging and discharging (Hannan et al. 2014). A combination of fuel cell, battery, and supercapacitor has also

Fuel Cell Vehicle Optimization and Control, Fig. 2

Example of an automotive fuel cell system and its balance of plant (BOP) comprised of hydrogen and air paths and cooling and humidification systems based on the Toyota Mirai (Arg 2018) and (Pukrushpan et al. 2002)



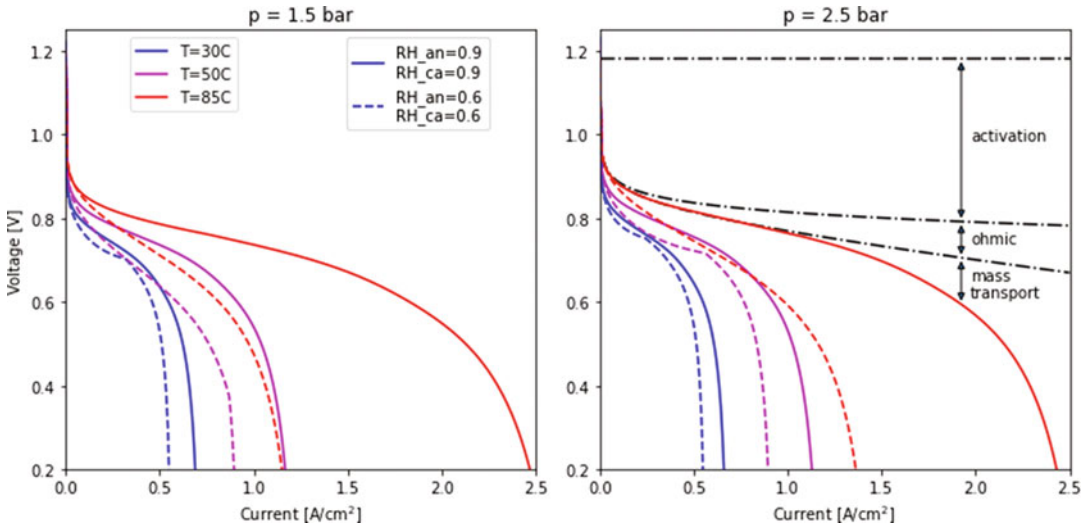
been proposed at a very high cost to take advantage of both good qualities of the two energy sources. After determining the correct energy source for the FCV, a control strategy that minimizes the hydrogen consumption while ensuring highly efficient operation of the FCV and longevity of the energy sources needs to be determined.

Fuel Cell Vehicle System

An example of a fuel cell vehicle powertrain with the balance of plant is shown in Fig. 2 highlighting at the top the power splitting and energy management between a battery and the fuel cell. The dynamic response of the fuel cell and its ability to follow the load required to power a vehicle depend on managing the air and hydrogen paths that supply the reactants within the BOP. Textbook material with publicly available software to simulate the air and hydrogen path, along with the membrane humidity and temperature, can be found in Pukrushpan et al. (2002). The primary goal of the BOP is to avoid starvation in the cathode and anode that lead to fuel cell degradation (Daud et al. 2017). Combination of fuel cell to other energy sources (such as batteries) can provide load leveling and allows the system

to operate at lower cathode stoichiometric ratios (lower reactant flows), thus reducing the parasitic losses associated with the air compressor (Suhn and Stefanopoulou 2005). Although the air and hydrogen paths have been addressed extensively in the past, the integration of cooling, thermal, and humidification to maintain optimal membrane hydration and avoid flooding even during startup, shutdown, and storage periods across a wide range of environmental conditions is still the frontier in the control and optimization field.

The problem of how much power is required from the fuel cell and the battery also arises. The power split has to maintain efficient operation of its components for driving from point A to point B. The load profile as shown in Fig. 2 is converted into electrical power through the motor (P_{load}). Then, the power split of the battery power and fuel cell for an FCV is determined by: $P_{fc,net} = P_{load} - P_e$ where P_e is the power required by other onboard energy sources such as batteries and $P_{fc,net}$ is the fuel cell's net stack power. A schematic of how the power flow occurs in an FCV is shown in Fig. 2. Further terms can be added to this power split rule, such as DC/DC converter power, depending on the fuel cell vehicle configuration and components. The battery power is given by: $P_b = n_s n_p I_b V_t$ where n_p is the number of battery cells in parallel in



Fuel Cell Vehicle Optimization and Control, Fig. 3 Impact of the fuel cell temperature, relative humidity, and operating pressure on fuel cell polarization curve

a module, n_s is the number of battery modules in series for forming a stack, I_b is the battery current, and V_t is the terminal voltage of a battery cell.

The power required from the fuel cell is obtained through a fuel cell current demand (input) and a voltage V_{fc} (output) on each cell in a fuel cell stack. As shown in Fig. 3, the fuel cell voltage decreases as the current drawn from the fuel cell increases and depends in a nonlinear way on cell temperature humidity and pressure. The total voltage of the fuel cell stack is then given as $V_{st} = n_{fc} V_{fc}$, where V_{st} is the fuel cell stack voltage and n_{fc} is the number of fuel cells in series in a fuel cell stack. The fuel cell stack power is given as $P_{fc} = n_{fc} V_{fc} I_{fc}$, where P_{fc} is the fuel cell stack power and I_{fc} is the fuel cell current and is the same for all cells if they are all connected in series.

The fuel cell efficiency is calculated as $\eta = \frac{P_{fc,net}}{E_{(H_2)}}$, where $P_{fc,net} = P_{fc} - P_{aux}$ is the net power of the fuel cell stack, P_{aux} is the power for the auxiliary components managing the BOP with $E_{(H_2)} = \frac{(\Delta H I_{fc})}{nF}$ being the energy value of hydrogen consumed, $\Delta H = 285.8 \frac{kJ}{mol}$ is the hydrogen higher heating value, $F = 96,485 \frac{C}{mol}$ is the Faraday constant, and $n = 2$ is the number of moles of hydrogen. The fuel cell efficiency

increases until the maximum value of 55–60% in the low to medium range of fuel cell power (typically near 20% of peak power). Then it drops on the high power region of fuel cell operation. The polarization losses shaping the fuel cell voltage response shown in Fig. 3 are the fundamental reason behind the FC peak efficiency at approximately 20% of peak power. Control techniques to maintain the FC system operation automatically around that optimum have been investigated in the past and include full FC system optimization or FC pack efficiency peak-finding via extremum seeking.

Fuel Cell Operation

The FC performance is described by a voltage versus current (or current density) relationship called polarization and shown in Fig. 3. From the polarization curve expression, it is evident that pressure, temperature, and relative humidity will affect the fuel cell's performance. In Fig. 3, simulated polarization curves at different temperatures are shown for a fuel cell. It can be appreciated that from 65 to 80 °C, the fuel cell's voltage on the polarization curve increases, which means the fuel cell will operate at a higher efficiency. From 85 °C and onward, dehydration occurs within the fuel cell. Consequently, a decrease in voltage is

produced in the polarization curve. At operation lower than 60 °C, water vapor condensation could occur, and flooding in the fuel cell is probable also, causing stack loss performance (Li et al. 2012) similar to the ones at 85 °C. Therefore, the fuel cell requires a management system to maintain the fuel cell within the operating temperature range (60–80 °C), pressure, and humidity which is called the balance of plant and is discussed in the next section.

Balance of Plant (BOP)

The balance of plant (BOP) components are responsible for cooling the fuel cell stack, providing enough reactants, and allowing the reactants' humidification before entering the fuel cell stack. BOP also allows adequate recirculation for hydrogen and water to allow fuel cell operation efficiently. The balance of plant (BOP) components usually consist of a compressor for the air delivery system and a pump/heat-exchanger/fan for the cooling system. They also include a pump for hydrogen recirculation and water pumps or humidifiers for humidification. The BOP components for liquid-cooled stacks are shown in Fig. 2 and denoted by the red lines. The combination of the BOP with the fuel cell is known as the fuel cell system.

The reactants need to be controlled to improve the performance of the fuel cell. The hydrogen (provided by pump) and air (provided by compressor or fan) need to be maintained at a certain stoichiometric ratio to ensure the efficient performance of the fuel cell. High hydrogen and airflow rates result in higher power density and greater stack efficiency but lower net power from the fuel cell due to the higher power consumption for the balance of plant components. High airflow rates also produce more water causing mass transport limitation and fuel cell degradation by water accumulation in the gas diffusion layer and catalyst layer. High hydrogen rates can help by having more frequent purging from the anode side, but it can cause dehydration on the cell and consequently fuel cell degradation.

The temperature of the fuel cell also needs to be controlled (through a fan or heat-exchanger/radiator) in the optimum operating

range to keep the humidity level in the membrane. As was previously shown, a higher temperature than the operating range can cause dehydration, shrinkage, wrinkles, and even rupture in the membrane. The humidity of the reactants needs to be controlled (through humidifiers) to maintain the membrane hydrated and efficient water distribution to avoid dehydration or flooding. Currently, the state of the art is using a self-humidifying fuel cell as the one used on the Toyota Mirai to reduce the balance of plant cost (Yoshida and Kojima 2015). Finally, the fuel cell power needs to be regulated using DC/DC converters since the voltage changes with load power. In addition the converter is responsible for enforcing the current split and can be used to avoid abrupt transients or changes in fuel cell power demand to avoid fuel cell degradation. The goal is to operate the fuel cell at its maximum efficiency point within most of its operation.

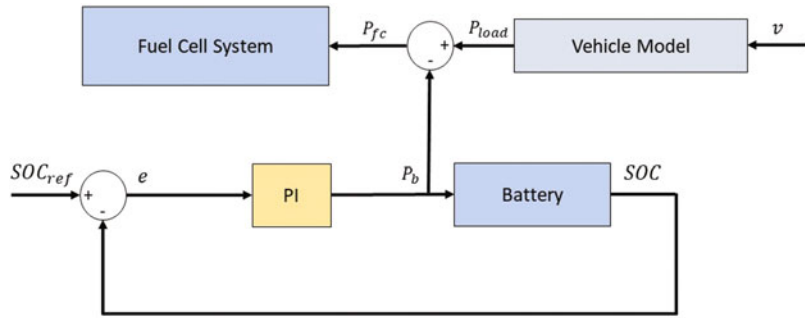
Power Split

Different control strategies have been used to determine the optimal power split required for an FCV. The overall hydrogen consumption of the fuel cell vehicle needs to be minimized to achieve efficient fuel cell and vehicle operation. Generally, the cost function consists of minimizing the hydrogen consumption over an entire trip:

$$J = \int \dot{m}_f dt = \frac{I_{fc} n_{fc} M_{H_2}}{2F} \quad (1)$$

where J is the cost, $\dot{m}_f$ is the hydrogen consumption, $M_{H_2} = 2.016 \times 10^{-3} \frac{kg}{mol}$ is the hydrogen molar mass, and $F = 96,485 \frac{C}{mol}$ is the Faraday's constant. This cost function can have other terms to minimize and weights to penalized these terms. The limitation of these strategies is that the driving cycle of the vehicle needs to be known beforehand. Other formulations consist of defining an instantaneous cost function, which is updated with time. The limitation with the instantaneous cost function is that the models and formulation need to be as simple as possible to avoid the high computational cost.

Fuel Cell Vehicle Optimization and Control, Fig. 4 Schematic of simple proportional-integral control implementation



Rule-Based Control (RBC) Strategies

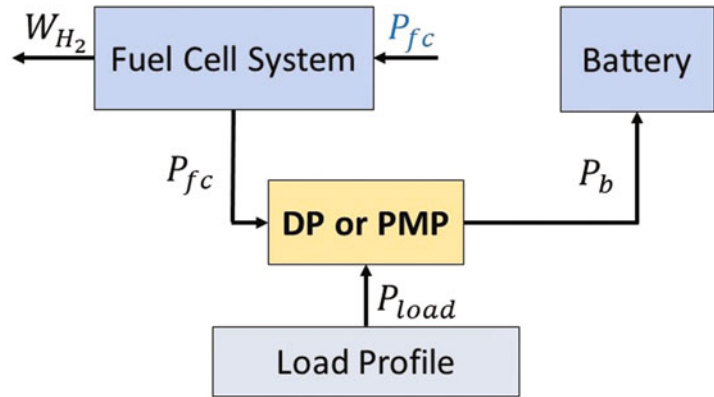
Rule-based strategies consist of rules based on experience with the goal to obtain an efficient power split between the fuel cell and an energy source storage such as battery or capacitor. These rules are easy to implement, but it is hard to determine whether they are optimal or not since they aim at maintaining a desired battery state of charge and not necessarily minimizing the overall energy consumption. The key point here is that the usefulness of consuming battery energy at a given point in time depends on future energy (hydrogen) used later to supplement this energy. The instantaneous conditions for the power split are thus hard to decide. Examples of rule-based strategies are proportional-integral control as shown in Fig. 4.

A proportional-integral (PI) controller is a feedback control loop that consists in calculating the error ($e(t) = r - y(t)$) between a reference value $r = SOC_{ref}$ and the output of the system $y(t) = SOC$. The SOC_{ref} is based on rules derived heuristically and aims to balance the use of battery to offset the use of FC and hence hydrogen. These rules are often expressed via IF-THEN rules and membership functions, which define certain situations and conditions that would enable either one or two energy sources between the main and the auxiliary energy sources. This rule-based method has become popular due to the easy implementation of simple rules without having to understand the complexity of the problem completely (Daud et al. 2017). Many researchers have implemented fuzzy logic controllers in FCV (Xiao et al. 2012). Once the rules guide to a fixed or slowly varying

reference battery state of charge, the remaining tuning relies on classical control techniques. Due to their low computational load, rule-based strategies can be easily implemented in real time in a FCV. The Toyota Mirai power split can be deduced from Arg (2018) and the modeling work of Davis and Hayes (2019). The Mirai strategy consists of a set of basic rules where the fuel cell is turned off during braking and idling. During driving conditions, the energy management is achieved through a load-following strategy that maintains the battery state of charge. Despite having a strong load following energy management, Davis and Hayes (2019) showed in simulations lower fuel cell degradation compared to a more load-leveling strategy.

In subsequent sections, we discuss how the battery state of charge can be defined based on an equivalent consumption minimization strategy. In either case, tuning the PI controller gains when the reference state of charge varies can be challenging. A proportional-integral controller was implemented in Wang et al. (2015) for full hybrid power system. The system included a PEMFC Horizon H-100, lithium-ion battery pack, and DC/DC unidirectional and bidirectional converter. The system's states consist of the inductor current and the output voltage of the DC/DC converter. The fuel cell model consisted of an equivalent electrical circuit with a variable double layer capacitor to display the transient characteristics. As typically done in such power split controllers, the fuel cell operating temperature was kept constant at 60 °C. The fuel cell auxiliary components were not specified, and neither the battery nor the fuel cell degradation was considered.

Fuel Cell Vehicle Optimization and Control, Fig. 5 In the dynamic programming (DP) and Pontryagin's minimum principle (PMP) formulation in Fares et al. (2015) and Zheng and Cha (2017), fuel cell power is used as an input (P_{fc}) and state of charge (SOC) as a state



F

Dynamic Programming (DP)

Dynamic programming (DP) is a numerical method that finds the optimal control sequence that minimizes a given cost function, in this case hydrogen consumption given constraints and a desired final battery state of charge. Since the optimal decision at every instant depends on the future demands and constraints, DP starts from the final step and propagates backward assuming knowledge of the entire load trajectory. Different papers applying DP have been developed with schemes similar to Fig. 5. In Fares et al. (2015), lookup tables based on experimental testing of an actual fuel cell (FC) test bench and the fuel cell power include the auxiliary power of the components used, such as water pumps, air-conditioner, and compressor. The operating temperature of the fuel cell is not specified. It is shown that the weighted DP improves hydrogen consumption compared to a state machine control algorithm for a fuel cell battery vehicle. However, only Matlab/Simulink simulations were performed.

Pontryagin's Minimum Principle (PMP)

Even though DP gives the optimal solution, its computational time exponentially increases as the number of states increases. Therefore, Pontryagin's minimum principle (PMP) can be an alternative solution to solve the control problem on FCVs. PMP consists of solving a set of conditions necessary for optimality. In Zheng and Cha (2017), PMP used fuel cell power as an input (P_{fc}) and state of charge (SOC) as

a state. A schematic of this is in Fig. 5. The cooling system is composed of a compressor and other auxiliary components. Degradation of the batteries and fuel cells was not considered, but a comparison between PMP to a rule-based energy management strategy was performed with respect to lower hydrogen consumption.

Equivalent Consumption Minimization Strategy (ECMS)

The ECMS has been proven to be equivalent under certain conditions to Pontryagin's minimum principle. This is a heuristic method that consists of reducing the global minimization problem into an instantaneous minimization problem. It is solved at each instant, only using a weighted value of the electric energy and on the actual energy flow in the powertrain. In Zhang et al. (2017), energy management using ECMS was used for a fuel cell hybrid vehicle using batteries and a supercapacitor. The ECMS only considered an optimum power of the fuel cell and battery by adding a penalty factor in terms of the battery SOC and fuel cell's mean efficiency. Fuel cell and battery degradation are not considered, and the simulation tool is not specified. More importantly, the FC system assumes a constant power of 11 kW for the total power of the auxiliary components, namely, an air-conditioner and a compressor. The fuel cell model consists of lookup tables, but the operating temperature is not specified pointing to the need for more comprehensive effort and experimentation for this promising work.

Model Predictive Control (MPC)

MPC uses model prediction to satisfy constraints on the input and output variables and measurements to adjust the actions (Seborg et al. 2011). MPC was used for buses and other applications that depend on the dynamic response of fuel cell and batteries (He et al. 2020). Many comparisons of the performance of a battery charge control, fuel cell dynamics, and fuel cell thermal dynamics were performed. These constraints were applied to avoid fuel cell degradation. The simulation results explored these constraints and their effect on hydrogen consumption, including auxiliary component losses for the cooling system.

Machine Learning

Machine learning consists of tools that detect patterns in data, and these patterns are used to predict future behavior or perform decisions under uncertainty. This method has been used, despite requiring large amounts of data, because of the excellent performance of trained models on the FCV. Reinforcement learning was used in Hsu et al. (2016) for the dynamic energy management of an FCV with batteries. This learning method consists of an agent observing the behavior of the data and updating the values of the parameters of the model through a reward function. One experiment where the reward function only accounts for charge sustaining SOC is included in Hsu et al. (2016). It was possible to show that on different driving cycles, reduced hydrogen consumption was achieved compared to the first experimental formulation.

Needs and Important Directions

Future work should use models that weigh the auxiliary losses and cost associated with membrane hydration and other physical phenomena that accelerate degradation to inform optimal control strategies (Goshtasbi and Ersal 2020) and total cost of ownership. Experimental validation on test benches or FCVs should also be performed to validate optimal control strategies

that consider auxiliary losses (Siegel et al. 2018). To reduce the cost of FCVs, the sizing of the FC and the battery need to be addressed for various drive cycles as in Sundström and Stefanopoulou (2007). Last but not least, connectivity and automation will allow even higher gains for these advanced powertrains, as shown in Kim et al. (2020).

Cross-References

- [Model Predictive Control in Practice](#)
- [Optimal Control and the Dynamic Programming Principle](#)

Bibliography

- Argonne (2018) Technology assessment of a fuel cell vehicle: 2017 Toyota Mirai. Technical report, Argonne National Laboratory
- Chapman A et al (2019) A review of four case studies assessing the potential for hydrogen penetration of the future energy system. *Int J Hydrog Energy* 44:6371–6382
- Daud WRW et al (2017) PEM fuel cell system control: a review. *Renew Energy* 113:620–638
- Davis K, Hayes JG (2019) Fuel cell vehicle energy management strategy based on the cost of ownership. *IET Electr Syst Transp* 9(4):226–236
- Fares D et al (2015) Dynamic programming technique for optimizing fuel cell hybrid vehicles. *Int J Hydrog Energy* 40:7777–7790
- Goshtasbi A, Ersal T (2020) Degradation-conscious control for enhanced lifetime of automotive polymer electrolyte membrane fuel cells. *J Power Sources* 457:227996
- Hannan MA et al (2014) Hybrid electric vehicles and their challenges: a review. *Renew Sust Energy Rev* 29:135–150
- He H, Quan S, Sun F, Wang Y-X (2020) Model predictive control with lifetime constraints based energy management strategy for proton exchange membrane fuel cell hybrid power systems. *IEEE Trans Ind Electron* 67(10):9012–9023
- Hsu RC et al (2016) A reinforcement learning based dynamic power management for fuel cell hybrid electric vehicle. In: 2016 joint 8th international conference on soft computing and intelligent systems and 2016 17th international symposium on advanced intelligent systems
- James BD (2020) 2020 DOE hydrogen and fuel cells program review presentation fuel cell systems analysis

- James BD et al (2018) Mass production cost estimation of direct H₂ PEM fuel cell systems for transportation applications: 2018 update. Technical report
- Kim Y et al (2020) Co-optimization of speed trajectory and power management for a fuel-cell/battery electric vehicle. *Int J Precis Eng Manuf* 260:1–17
- Liv (2011) Electric vehicles modelling and simulations. InTech
- Li X et al (2020) Energy management of hybrid electric vehicles: a review of energy optimization of fuel cell hybrid power system based on genetic algorithm. *Energy Convers Manag* 205:112474
- Li H et al (2012) A review of water flooding issues in the proton exchange membrane fuel cell. *J. Power Sources* 178:103–117
- McKinsey (2017) Hydrogen scaling up. Technical report, Hydrogen Council
- O'Hayre R et al (2016) Fuel cell fundamentals. Wiley
- Pukrushpan JT et al (2002) Simulation and analysis of transient fuel cell system performance based on a dynamic reactant flow model. In: Proceedings of IMECE2002 ASME international mechanical engineering congress & exposition
- Salahuddin U et al (2018) Grid to wheel energy efficiency analysis of battery and fuel cell-powered vehicles. *Int J Energy Res* 42(5):2021–2028
- Seborg DE et al (2011) Process dynamics and control. Wiley
- Siegel JB et al (2018) Cooling parasitic considerations for optimal sizing and powersplit strategy for military robot powered by hydrogen fuel cells. SAE Technical Paper
- Suhn K-W, Stefanopoulou AG (2005) Coordination of converter and fuel cell controllers. *Int J Energy Res* 29:1167–1189
- Sundström O, Stefanopoulou A (2007) Optimum battery size for fuelcell hybrid electric vehicle-part II. *J Electrochem En Conv Stor* 4(2):176–184
- Tanc B et al (2019) Overview of the next quarter century vision of hydrogen fuel cell electric vehicles. *Int J Hydrog Energy* 44(20):10120–10128
- Wang Y-X et al (2015) Modeling and experimental validation of hybrid proton exchange membrane fuel cell/batterysystem for power management control. *Int J Hydrog Energy* 40:11713–11721
- Xiao D et al (2012) The research of energy management strategy for fuel cell hybrid vehicle. In: 2012 international conference on industrial control and electronics engineering
- Yoshida T, Kojima K (2015) Toyota MIRAI fuel cell vehicle and progress toward a future hydrogen society. *Electrochem Soc Interface* 24(2):44–49
- Zhang W et al (2017) Optimization for a fuel cell/battery/capacity tram with equivalent consumption minimization strategy. *Energy Convers Manag* 134:59–69
- Zheng C, Cha SW (2017) Real-time application of ponytryagin's minimum principle to fuel cell hybrid buses based on driving characteristics of buses. *Int J Precision Eng Manuf Green Technol* 4(2):199–209

Fundamental Limitation of Feedback Control

Jie Chen

City University of Hong Kong, Hong Kong, China

Abstract

Feedback systems are designed to meet many different objectives. Yet, not all design objectives are achievable as desired due to the fact that they are often mutually conflicting and that system properties themselves may impose design constraints and thus limitations on the performance attainable. An important step in the control design process is then to assess and analyze what and how system characteristics may impose constraints and, accordingly, how to make tradeoffs between different objectives by judiciously navigating between the constraints. Fundamental limitation of feedback control is an area of research that addresses these constraints, limitations, and tradeoffs.

Keywords

Bode integrals · Design tradeoff · Performance limitation · Tracking and regulation limits

Introduction

Fundamental control limitations are referred to those intrinsic of feedback that can neither be overcome nor circumvent regardless how it may be designed. By this nature, the study of fundamental limitations dwells on a Hamletian question: Can or can't it be done? What can and cannot be achieved? What is the best and the worst possible? To be more specific, yet still general enough, at heart here are issues concerning the benefit and cost of feedback. We ask such questions as “(1) What system characteristics may impose inherent limitations regardless of controller design?”, “(2) What inherent

constraints may exist in design and what kind of tradeoffs are to be made?”, “(3) What are the best achievable performance limits?”, and “(4) How can the constraints, limitations, and limits be quantified in ways meaningful for control analysis and design?”. Needless to say, issues of this kind are very broad and in fact are omnipresent in science and engineering. Analogies can be made, for example, to Shannon’s theorems in communications theory, the Cramer-Rao bound in statistics, and Heisenberg’s uncertainty principle in quantum mechanics; they all address the fundamental limits and limitations, though for different problems and in different contexts. The search for fundamental limitations of feedback control, as such, may be considered a quest for an “ultimate truth” or the “law of feedback.”

For their fundamentality and importance, inquiries into control performance limitations have persisted over time and continue to be of vital interest. It is worth emphasizing, however, that performance limitation studies are not merely driven by intellectual curiosity, but are tantamount to better and more realistic feedback systems design and hence of tangible practical value. An analysis of performance limitations can aid control design in several aspects. First, it may provide a fundamental limit on the best performance attainable irrespective of controller design, thus furnishing a guiding benchmark in the design process. Secondly, it helps a designer assess what and how system properties may be inherently conflicting and thus pose inherent difficulties to performance objectives, which in turn helps the designer specify reasonable goals, and make judicious modifications and revisions on the design. In this process, the theory of fundamental control limitations promises to provide valuable insights and analytical justifications to long-held design heuristics and, indeed, to extend such heuristics further beyond. This has become increasingly more relevant, as modern control design theory and practice relies heavily on automated, optimization-based numerical routines and tools.

Systematic investigation and understanding of fundamental control limitations began with the classical work of Bode in the 1940s on

logarithmic sensitivity integrals, known as the Bode integrals. Bode’s work has had a lasting impact on the theory and practice of control and has inspired continued research endeavor dated most recently, leading to a variety of extensions and new results which seek to quantify design constraints and performance limitations by frequency-domain logarithmic integrals. On the other hand, the search for the best achievable performance is a natural goal, and indeed an innate instinct, in the study of optimal control problems, which has lend bounds on optimal performance indices defined under various criteria. Likewise, developments in this latter area of study have also been substantial and are continuing to branch to different problems and different system categories.

In this entry we attempt to provide a summary overview of the key developments in the study of fundamental limitations of feedback control. While the understanding on this subject has been compelling and the results are rather prolific, we focus on Bode-type integral relations and the best achievable performance limits, two branches of the study that are believed to be most well-developed. Roughly speaking, the Bode-type integrals are most useful for quantifying the inherent design constraints and tradeoffs in the frequency domain, while the performance results provide fundamental limits of canonical control objectives defined using frequency- and time-domain criteria. Invariably, the two sets of results are intimately related and reinforce each other. The essential message then is that despite its many benefits, feedback has its own limitations and is subject to various constraints. Feedback design, for that sake, requires oftentimes a hard tradeoff.

Control Design Specifications

We begin by introducing the basic notation to be used in the sequel. Let $\mathbb{C}_+ := \{z : \text{Re}(z) > 0\}$ denote the open right half plane (RHP) and $\overline{\mathbb{C}}_+$ the closed RHP (CRHP). For a complex number z , we denote its conjugate by $\bar{z}$. For a complex vector x , we denote its conjugate transpose by

x^H , and its Euclidean norm by $\|x\|_2$. The largest singular value of a matrix A will be written as $\bar{\sigma}(A)$. If A is a Hermitian matrix, we denote by $\bar{\lambda}(A)$ its largest eigenvalue. For any unitary vectors $u, v \in \mathbb{C}^n$, we denote by $\angle(u, v)$ the *principal angle* between the two one-dimensional subspaces, called the *directions*, spanned by u and v :

$$\cos \angle(u, v) := |u^H v|.$$

For a stable continuous-time system with transfer function matrix $G(s)$, we define its $\mathcal{H}_\infty$ norm by

$$\|G\|_\infty := \sup_{\operatorname{Re}(s) > 0} \bar{\sigma}(G(s)).$$

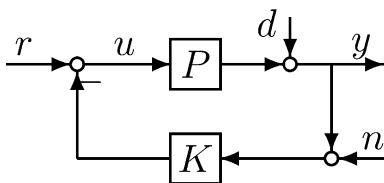
We consider the standard configuration of finite-dimensional linear time-invariant (LTI) feedback control systems given in Fig. 1. In this setup, P and K represent the transfer functions of the plant model and controller, respectively, r is a command signal, d a disturbance, n a noise signal, and y the output response. Define the open-loop transfer function, the sensitivity function, and the complementary sensitivity function by

$$L = PK, \quad S = (I + L)^{-1}, \quad T = L(I + L)^{-1},$$

respectively. Then the output can be expressed as

$$y = Sd - Tn + SP r.$$

The goal of feedback control design is to design a controller K so that the closed-loop system is stable and that it achieves certain performance specifications. Typical design objectives include:



Fundamental Limitation of Feedback Control, Fig. 1
Feedback configuration

- *Disturbance attenuation.* The effect of the disturbance signal on the output should be kept small, which translates into the requirement that the sensitivity function be small in magnitude at the frequencies of interest. For a single-input single-output (SISO) system, this mandates that

$$|S(j\omega)| < 1, \quad \forall \omega \in [0, \omega_1].$$

The sensitivity magnitude $|S(j\omega)|$ is to be kept as small as possible in the low-frequency range, where the disturbance signal concentrates.

- *Noise reduction.* The noise response should be reduced at the output. This requires that the complementary sensitivity function be small in magnitude at frequencies of interest. For a SISO system, the objective is to achieve

$$|T(j\omega)| < 1, \quad \forall \omega \in [\omega_2, \infty).$$

Similarly, the magnitude $|T(j\omega)|$ is desired to be small at high frequencies, since noise signals typically consist of significant high-frequency harmonics.

Moreover, feedback can be introduced to achieve many other objectives including regulation, command tracking, improved sensitivity to parameter variations, and, more generally, system robustness, all by manipulating the three key transfer functions: the open-loop transfer function, the sensitivity function, and the complementary sensitivity function.

The design and implementation of feedback systems, on the other hand, are also subject to many constraints, which include:

1. *Causality:* A system must be causal for it to be implementable. The causality requirement limits the means of compensation to those that are physically realizable; for example, no ideal filter can be used, and the system's relative degree and delay cannot be cancelled but must be preserved.

2. **Stability:** The closed-loop system must be stable. Mathematically, this dictates that every closed-loop transfer function must be bounded and analytic in CRHP.
3. **Interpolation:** There should be no unstable pole-zero cancelation between the plant and controller, in order to rid of hidden instability. The requirement is also necessary for system robustness, as in reality no perfect cancellation is possible. Thus, for a SISO system, it is necessary that

$$S(p_i) = 0, \quad T(p_i) = 1,$$

$$S(z_i) = 1, \quad T(z_i) = 0$$

at each CRHP pole p_i and zero z_i of the plant.

4. **Structural constraints:** These constraints arise from the feedback structure itself; for example, $S(s) + T(s) = 1$. The implication is that the closed-loop transfer functions cannot be independently designed, thus resulting in conflicting design objectives.

For a given plant, each of these constraints is unalterable and hence is fundamental, and each will constrain the performance attainable in one way or another. The question we face then is how the constraints may be captured in a form that is directly pertinent and useful to feedback design.

Bode Integral Relations

In the classical feedback control theory, Bode's gain-phase formula (Bode 1945) is used to express the aforementioned design constraints for SISO systems.

Bode Gain-Phase Integral *Suppose that $L(s)$ has no pole and zero in $\overline{\mathbb{C}}_+$. Then at any frequency ω_0 ,*

$$\angle L(j\omega_0) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{d \log |L(jv)|}{dv} \log \coth \frac{|v|}{2} dv,$$

where $v = \log(\omega/\omega_0)$.

A special form of the Hilbert transform, this gain-phase formula relates the gain and phase of the open-loop transfer function evaluated along the imaginary axis, whose implication may be explained as follows. In order to make the sensitivity response small in the low-frequency range, the open-loop transfer function is required to have a high gain; the higher, the better. On the other hand, for noise reduction and robustness purposes, we need to keep the loop gain low at high frequencies, the lower the better. Evidently, to maximize these objectives, we want the two frequency bands as wide as possible. This then requires a steep decrease of the loop gain and hence a rather negative slope in the intermediate frequency range. Take ω_0 as the crossover frequency. The gain-phase relationship tells that a more negative derivative of the loop gain near ω_0 (implying a faster decrease of the gain) will lead to a more negative phase, thus driving the phase closer to the negative 180 degrees, namely, the critical point of stability. It consequently reduces the phase margin and may even cause instability. As a result, the gain-phase relationship demonstrates a conflict between the two design objectives. It is safe to claim that much of the classical feedback design theory came as a consequence of this simple relationship, aiming to shape the open-loop frequency response in the crossover region by trial and error, using lead or lag filters.

The gain-phase formula allows us to quantify, indirectly, the closed-loop design constraints and performance tradeoffs via open-loop data and by loopshaping design. In doing so, the design specifications imposed on closed-loop transfer functions are translated approximately into the requirements on the open-loop transfer function, and the tradeoff between different design goals is achieved by shaping the open-loop gain and phase. A more direct vehicle to accomplish this same goal is Bode's sensitivity integral (Bode 1945).

Bode Sensitivity Integrals *Let $p_i \in \mathbb{C}_+$ be the unstable poles and $z_i \in \mathbb{C}_+$ the nonminimum phase zeros of $L(s)$. Suppose that the closed-loop system in Fig. 1 is stable.*

- (i) If $L(s)$ has relative degree greater than one, then

$$\int_0^\infty \log |S(j\omega)| d\omega = \pi \sum_i p_i.$$

- (ii) If $L(s)$ contains no less than two integrators, then

$$\int_0^\infty \frac{\log |T(j\omega)|}{\omega^2} d\omega = \pi \sum_i \frac{1}{z_i}.$$

Bode's original work concerns the sensitivity integral for open-loop stable systems only. The integral relations shown herein, which are attributed to Freudenberg and Looze (1985) and Middleton (1991), respectively, provide generalizations to open-loop unstable and nonminimum phase systems.

Why are Bode integrals important? What is the hidden message behind the mathematical formulas? Simply put, the Bode sensitivity integral relation (*vis-a-vis* the complementary sensitivity integral relation) exhibits that a feedback system's sensitivity must abide some kind of conservation law or invariance principle: the integral of the logarithmic sensitivity magnitude over the entire frequency range must be a nonnegative constant, determined by the open-loop unstable poles. This principle mandates a tradeoff between sensitivity reduction and sensitivity amplification in different frequency bands. Indeed, to achieve disturbance attenuation at low frequencies, the logarithmic sensitivity magnitude must be kept below zero db, the lower the better. For noise reduction and system robustness, however, the sensitivity magnitude must decrease to zero db at high frequencies. Since, in light of the sensitivity integral relation, the total area under the logarithmic sensitivity magnitude curve is nonnegative, the logarithmic magnitude must rise above zero db in the intermediate frequency range so that under its curve, the positive and negative areas may cancel each other to yield a nonnegative value. As such, an undesirable sensitivity amplification occurs, resulting in a fundamental tradeoff between the desirable sensitivity reduction and

the undesirable sensitivity amplification, known colloquially as the *waterbed effect*.

MIMO Integral Relations

For a multi-input multi-output (MIMO) system depicted in Fig. 1, the sensitivity and complementary sensitivity functions, which now are transfer function matrices, satisfy similar interpolation constraints: at each RHP pole p_i and zero z_i of $L(s)$, the equations

$$\begin{aligned} S(p_i)\eta_i &= 0, & T(p_i)\eta_i &= \eta_i, \\ w_i^H S(z_i) &= w_i^H, & w_i^H T(z_i) &= 0 \end{aligned}$$

hold with some unitary vectors η_i and w_i , where η_i is referred to as a *right pole direction vector* associated with p_i and w_i a *left zero direction vector* associated with z_i . The frequency-domain design specifications are also similar but are now quantified using the singular values of the closed-loop transfer function matrices. Specifically, the disturbance attenuation and noise reduction requirements may now be specified as

$$\bar{\sigma}[S(j\omega)] < 1, \quad \forall \omega \in [0, \omega_1]$$

and

$$\bar{\sigma}[T(j\omega)] < 1, \quad \forall \omega \in [\omega_2, \infty),$$

respectively.

The classical Bode integrals are only applicable to SISO systems. While it seems both natural and tempting, the extension of Bode integrals to MIMO systems has been highly nontrivial a task. Deep at the root is the complication resulted from the directionality properties of MIMO systems. Unlike in a SISO system, the measure of frequency response magnitude is now the largest singular value of a transfer function matrix, which represents the worst-case amplification of energy-bounded signals, a direct counterpart to the gain of a scalar transfer function. This fact alone casts a fundamental difference and poses a formidable obstacle. From a technical standpoint, the logarithmic function of the largest singular value is no longer a harmonic function

as in the SISO case, but only a subharmonic function. Much to our chagrin then, familiar tools from the analytic function theory, such as Cauchy and Poisson theorems, the very backbones in developing the scalar Bode integrals, cease to be useful. Indeed, the lack of the comparably strong analyticity of singular value functions is perhaps one of the main barriers why MIMO versions of the Bode integrals remained an open, unanswered question for several decades. Nevertheless, it remains possible to extend Bode integrals in their essential spirit. Advances are made by Chen (1995, 1998, 2000).

MIMO Bode Sensitivity Integrals *Let $p_i \in \mathbb{C}_+$ be the unstable poles of $L(s)$ and $z_i \in \mathbb{C}_+$ the nonminimum phase zeros of $L(s)$. Suppose that the closed-loop system in Fig. 1 is stable.*

- (i) *If $L(s)$ has relative degree greater than one, then*

$$\int_0^\infty \log \bar{\sigma}(S(j\omega)) d\omega \geq \pi \bar{\lambda} \left(\sum_i p_i \zeta_i \zeta_i^H \right).$$

- (ii) *If $L(s)$ contains no less than two integrators, then*

$$\int_0^\infty \frac{\log \bar{\sigma}(T(j\omega))}{\omega^2} d\omega \geq \pi \bar{\lambda} \left(\sum_i \frac{1}{z_i} \xi_i \xi_i^H \right)$$

where ζ_i and ξ_i are some unitary vectors related to the right pole direction vectors associated with p_i and the left zero direction vectors associated with z_i , respectively.

From these extensions, it is evident that same limitations and tradeoffs on the sensitivity and complementary sensitivity functions carry over to MIMO systems; in fact, both integrals reduce to the scalar Bode integrals when specialized to SISO systems. Yet there is something additional to and unique of MIMO systems: the integrals now depend on not only the locations but also the directions of the zeros and poles. In particular, it can be shown that they depend on the mutual orientation of these directions and the dependence can be explicitly characterized geometrically by

the principal angles between the directions. This new phenomenon, which finds no analog in SISO systems, thus highlights the important role of directionality in sensitivity tradeoff and more generally, in the design of MIMO systems.

A more sophisticated and, accordingly, more informative variant of Bode integrals is the Poisson integral for sensitivity and complementary sensitivity functions (Freudenberg and Looze 1985), which can be used to provide quantitative estimates of the waterbed effect. MIMO versions of Poisson integrals are also available (Chen 1995, 2000).

Frequency-Domain Performance Bounds

Performance bounds complement the integral relations and provide fundamental thresholds to the best possible performance ever attainable. Such bounds are useful in providing benchmarks for evaluating a system's performance prior to and after controller design. In the frequency domain, fundamental limits can be specified as the minimal peak magnitude of the sensitivity and complementary sensitivity functions achievable by feedback or, formally, the minimal achievable $\mathcal{H}_\infty$ norms:

$$\gamma_{\min}^S := \inf \{ \|S(s)\|_\infty : K(s) \text{ stabilizes } P(s) \},$$

$$\gamma_{\min}^T := \inf \{ \|T(s)\|_\infty : K(s) \text{ stabilizes } P(s) \}.$$

Drawing upon *Nevanlinna-Pick interpolation* theory for analytic functions, one can obtain exact performance limits under rather general circumstances (Chen 2000).

$\mathcal{H}_\infty$ Performance Limits *Let $z_i \in \mathbb{C}_+$ be the nonminimum phase zeros of $P(s)$ with left direction vectors w_i and $p_i \in \mathbb{C}_+$ the unstable poles of $P(s)$ with right direction vectors η_i , where z_i and p_i are all distinct. Then,*

$$\gamma_{\min}^S = \gamma_{\min}^T = \sqrt{1 + \bar{\sigma}^2 \left(Q_p^{-1/2} Q_{zp} Q_z^{-1/2} \right)},$$

where Q_z , Q_p , and Q_{zp} are the matrices given by

$$Q_z := \left[\frac{w_i^H w_j}{z_i + \bar{z}_j} \right], \quad Q_p := \left[\frac{\eta_i^H \eta_j}{\bar{p}_i + p_j} \right],$$

$$Q_{zp} := \left[\frac{w_i^H \eta_j}{z_i - p_j} \right].$$

More explicit bounds showing how zeros and poles may interact to have an effect on these limits can be obtained, e.g., as

$$\gamma_{\min}^S = \gamma_{\min}^T \geq \sqrt{\sin^2 \angle(w_i, \eta_j) + \left| \frac{p_j + \bar{z}_i}{p_j - z_i} \right|^2 \cos^2 \angle(w_i, \eta_j)},$$

F

which demonstrates once again that the pole and zero directions play an important role in MIMO systems. Note that for RHP poles and zeros located in the close vicinity, this bound can become excessively large, which serves as another vindication why unstable pole-zero cancelation must be prohibited. Note also that for MIMO systems, however, whether near pole-zero cancelation is problematic depends additionally on the mutual orientation of the pole and zero directions.

The bounds given above are the solutions to the well-known *sensitivity minimization problem* in the $\mathcal{H}_\infty$ optimal control theory. For SISO systems, they reduce to Zames and Francis (1983) and Khargonecker and Tannenbaum (1985)

$$\gamma_{\min}^S \geq \prod_i \left| \frac{z + \bar{p}_i}{z - p_i} \right|,$$

where p_i are the unstable poles and z is a non-minimum phase zero of $P(s)$, and

$$\gamma_{\min}^T \geq \prod_i \left| \frac{p + \bar{z}_i}{p - z_i} \right|,$$

with z_i being the nonminimum phase zeros and p an unstable pole of $P(s)$. For more general, the so-called weighted sensitivity minimization problem, less explicit, computational formulas are available to determine the best achievable sensitivity and complementary sensitivity magnitudes.

Time-Domain Tracking and Regulation Limits

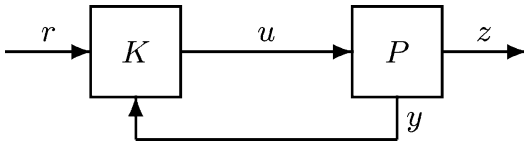
Tracking and regulation are two canonical objectives of servo mechanisms and constitute chief criteria in assessing the performance of feedback control systems (Kwakernaak and Sivan 1972; Qiu 1993). Understandings gained from these problems shed light into more general issues indicative of feedback design. In its full generality, a tracking system can be depicted as in Fig. 2, in which a 2-DOF (degree of freedom) controller $K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$ is designed for the output z to track a given reference input r , based on the feedforward of the reference signal r and the feedback of the measured output y :

$$u = K_1 r + K_2 y.$$

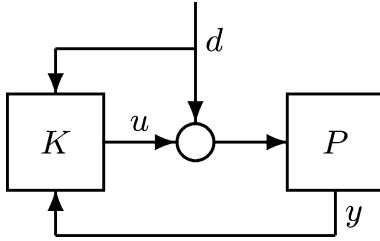
The tracking performance is defined in the time domain by the integral square error

$$J = \int_0^\infty \|z(t) - r(t)\|_2^2 dt,$$

Typically, we take r to be a step signal, which in the MIMO setting corresponds to a unitary constant vector, i.e., $r(t) = v$, $t > 0$ and $r(t) = 0$, $t < 0$, where $\|v\|_2 = 1$. We assume P to be LTI. But K can be arbitrarily general, as long as it is causal and stabilizing. We want z to not only track r asymptotically but also minimize J . But how small can it be?



Fundamental Limitation of Feedback Control, Fig. 2
2-DOF tracking control structure



Fundamental Limitation of Feedback Control, Fig. 3
2-DOF regulator

For the regulation problem, a general setup is given in Fig. 3. Likewise, K may be taken as a 2-DOF controller. The control energy is measured by the quadratic cost

$$E = \int_0^\infty \|u(t)\|_2^2 dt.$$

We consider a disturbance signal d , typically taken as an impulse signal, $d(t) = v\delta(t)$, where v is a unitary vector. In this case, the disturbance can be interpreted as a nonzero initial condition, and the controller K is to regulate the system's zero-input response. Similarly, we assume that P is LTI but allow K to be any causal, stabilizing controller. Evidently, for a stable P , the problem is trivial; the response will restore itself to the origin, and thus no energy is required. But what if the system is unstable? How much energy does the controller must generate to combat the disturbance? What is the smallest amount of energy required? These questions are answered by the best achievable limits of the tracking and regulation performance (Chen et al. 2000, 2003).

Tracking and Regulation Performance Limits

Let $p_i \in \mathbb{C}_+$ and $z_i \in \mathbb{C}_+$ be the RHP poles and zeros of $P(s)$, respectively. Then,

$$\inf\{E: K \text{ stabilizes } P(s)\} = \sum_i p_i \cos^2 \angle(\zeta_i, v),$$

$$\inf\{J: K \text{ stabilizes } P(s)\} = \sum_i \frac{1}{z_i} \cos^2 \angle(\xi_i, v),$$

where ζ_i and ξ_i are some unitary vectors related to the right pole direction vectors associated with p_i and the left zero direction vectors associated with z_i , respectively.

It becomes instantly clear that the optimal performance depends on both the pole/zero locations and their directions. In particular, it depends on the mutual orientation between the input and pole/zero directions. This sheds some interesting light. Take the tracking performance for an example. For a SISO system, the minimal tracking error can never be made zero for a nonminimum phase plant; in other words, perfect tracking can never be achieved. Yet this is possible for MIMO systems, when the input and zero directions are appropriately aligned, specifically when they are orthogonal. Similarly, for SISO unstable systems, a certain amount of energy is necessary to achieve regulation, and the minimal regulation energy cannot be zero. But for a MIMO system, it can be zero; no energy is required at all when the disturbance and pole directions are orthogonal. While seemingly somewhat counterintuitive, this scenario is in fact intuitively sensible; it merely says that the disturbance happens to enter from a direction where it does not excite the unstable modes, and so the system will be at rest all by itself.

Interestingly, the optimal performance in both cases can be achieved by LTI controllers, though allowed to be more general. As a consequence, the results provide the true fundamental limits that cannot be further improved by any causal feedback, using any other more general control laws such as nonlinear, time-varying feedforward and feedback. They are simply the best one can ever hope for, and the LTI controllers turn out to be optimal, provided that the plant itself is LTI. Moreover, it is worth pointing out the optimal tracking and regulation performance problems posed herein are formulated in an extreme, idealistic setting, with no practical consideration. The optimal controllers so obtained are likely to result

in excessive energy or magnitude peaking. This, of course, is seldom possible nor viable in practice. Tracking and regulation performances in more realistic formulations that take into account additional constraints, such as constraints on the control energy, were considered in, e.g., Chen et al. (2003).

Summary and Future Directions

Whether in time or frequency domain, while the results under survey may differ in forms and contexts, they unequivocally point to the fact that inherent constraints exist in feedback design, and fundamental limitations will necessarily arise, limiting the performance achievable regardless of controller design. Such constraints and limitations are exacerbated by the nonminimum phase zeros and unstable poles in the system. Understanding of these constraints and limitations proves essential to the success of control design. Additional perspectives and insights into this topic can be found in, e.g., Seron (1997), Åström (2000), and Chen et al. (2019).

Both the Bode-type integrals and the performance bounds presented herein admit representations in terms of input-output spectral densities or such information-theoretic measures as *entropy rates*, offering alternative interpretations based on the spectral or information contents of a system's input and output signals, instead of the system's transfer functions. The information-theoretic representation is especially susceptible to characterizing constraints that arise in information-limited and, more generally, networked feedback systems, where noisy and limited information transmission within the feedback loop results in additional limitations on the achievable control performance. The study of such systems requires incorporation of information and communication constraints into control performance analysis and, for that matter, the integration of information and control theories. We refer this undertaking to Fang et al. (2017) and Chen et al. (2019).

For both its intrinsic appeal and practical importance, the study of fundamental control limitations has been a topic of enduring vitality,

and the interest will continue unabated. New technological advances and applications in networked control, multi-agent systems, complex networks, cyber physical systems, and Internet of things (IoT) will usher in new issues to the study of control performance limitations. How control and communication constraints in such large-scale, distributed interconnected networks manifest themselves and how they may be characterized from a control performance perspective present daunting challenges. Performance limitation studies will undoubtedly find expansion and outgrowth into these broad fields. The integration of information and control theories, long awaited since the time of Bode, Shannon, and Wiener, is only beginning to emerge and will likely require a conceptual leap and new mathematical tools.

Cross-References

- [H-Infinity Control](#)
- [H₂ Optimal Control](#)
- [Linear Quadratic Optimal Control](#)

Bibliography

- Åström KJ (2000) Limitations on control system performance. *Eur J Control* 6:2–20
- Bode HW (1945) Network analysis and feedback amplifier design. Van Nostrand, Princeton
- Chen J (1995) Sensitivity integral relations and design tradeoffs in linear multivariable feedback systems. *IEEE Trans Autom Control* 40(10):1700–1716
- Chen J (1998) Multivariable gain-phase and sensitivity integral relations and design tradeoffs. *IEEE Trans Autom Control* 43(3):373–385
- Chen J (2000) Logarithmic integrals, interpolation bounds, and performance limitations in MIMO systems. *IEEE Trans Autom Control* 45:1098–1115
- Chen J, Qiu L, Toker O (2000) Limitations on maximal tracking accuracy. *IEEE Trans Autom Control* 45(2):326–331
- Chen J, Hara S, Chen G (2003) Best tracking and regulation performance under control energy constraint. *IEEE Trans Autom Control* 48(8):1320–1336
- Chen J, Fang S, Ishii H (2019) Fundamental limitations and intrinsic limits of feedback: an overview in an information age. *Annu Rev Control* 47:155–177
- Fang S, Chen J, Ishii H (2017) Towards integrating control and information theories: from information-theoretic

- measures to control performance limitations. Springer, Cham
- Freudenberg JS, Looze DP (1985) Right half plane zeros and poles and design tradeoffs in feedback systems. *IEEE Trans Autom Control* 30(6):555–565
- Khargonekar PP, Tannenbaum A (1985) Non-Euclidian metrics and the robust stabilization of systems with parameter uncertainty. *IEEE Trans Autom Control* 30(10):1005–1013
- Kwakernaak H, Sivan R (1972) *Linear optimal control systems*. Wiley, New York
- Middleton RH (1991) Trade-offs in linear control system design. *Automatica* 27(2):281–292
- Qiu L, Davison EJ (1993) Performance limitations of non-minimum phase systems in the servomechanism problem. *Automatica* 29(2):337–349
- Seron MM, Braslavsky JH, Goodwin GC (1997) *Fundamental limitations in filtering and control*. Springer, London
- Zames G, Francis BA (1983) Feedback, minimax sensitivity, and optimal robustness. *IEEE Trans Autom Control* 28(5):585–601

Game Theory for Security

Tansu Alpcan
Department of Electrical and Electronic
Engineering, The University of Melbourne,
Melbourne, VIC, Australia

Abstract

Game theory provides a mature mathematical foundation for making security decisions in a principled manner. Security games help formalizing security problems and decisions using quantitative models. The resulting analytical frameworks lead to better allocation of limited resources and result in more informed responses to security problems in complex systems and organizations. The game-theoretic approach to security is applicable to a wide variety of cyberphysical systems and critical infrastructures such as electrical power systems, financial services, transportation, data centers, and communication networks.

Keywords

Security games · Game theory ·
Cyberphysical system security · Complex
systems

Introduction

Securing a cyberphysical system involves making numerous decisions whether the system is a computer network, is part of a business process in an organization, or belongs to a critical infrastructure. One has to decide on, for example, how to configure sensors for surveillance, collect further information on system properties, and allocate resources to secure a critical segment or who should be able to access a specific function in the system. The decision-maker can be, depending on the setting, a regular employee, a system administrator, or the chief technical officer of an organization. In many cases, the decisions are made automatically by a computer program such as allowing a packet pass the firewall or filtering it out. The time frame of these decisions exhibits a high degree of variability from milliseconds, if made by software, to days and weeks, e.g., when they are part of a strategic plan. Each security decision has a cost, and any decision-maker is always constrained by limited amount of available resources. More importantly, each decision carries a security risk that needs to be taken into account when balancing the costs and the benefits.

Security games facilitate building analytical models which capture the interaction between

malicious attackers, who aim to compromise networks, owners, or administrators defending them. Attacks exploiting vulnerabilities of the underlying systems and defensive countermeasures constitute the moves of the game. Cost-benefit analysis of potential attacks and defenses, including perceived risks, provides the criteria, which the players act upon when choosing actions. Thus, the strategic struggle between attackers and defenders is formalized quantitatively based on the solid mathematical foundation provided by the field of game theory.

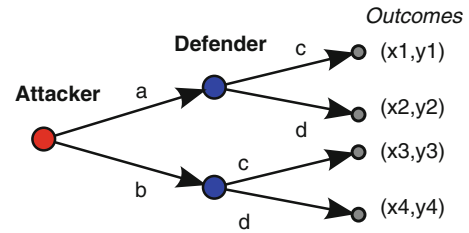
An important aspect of security games is the allocation of limited available resources from the perspectives of both attackers and defenders. If the players had access to unlimited resources (e.g., time, computing power, bandwidth), then the resulting security games would be trivial. In real-world security settings, however, both attackers and defenders have to act strategically and make numerous decisions when allocating their respective resources. Unlike in an optimization approach, security games take into account the decisions and resource limitations of both the attackers and the defenders.

Security Games

A security game is defined with four components: the players, the set of possible actions or strategies for each player, the outcome of the game for each player as a result of their action-reaction, and information structures in the game. The players have their own (selfish or malicious) motivations and inherent resource constraints. Based on these motivations and information available, they choose most beneficial strategies for themselves and act accordingly. The actions of players could be concurrent or sequential in which a player takes the lead and the other player follows. Hence, game theory helps analyzing decision-makers interacting on a system in a quantitative manner.

An Example Formulation

A simple security game can be formulated as a two-player and strategic (noncooperative) one,



Game Theory for Security, Fig. 1 A simple, two-player security game

where one player is the *attacker* and the other one is the *defender* protecting a system. Let the discrete actions available to the attacker and the defender be $\{a, b\}$ and $\{c, d\}$, respectively. Each attack-defense pair leads to one of the outcome pairs for the attacker and the defender $\{(x1, y1), (x2, y2), (x3, y3), (x4, y4)\}$, which represent the respective player's gains (or losses). This classic security game is depicted graphically in Fig. 1. It can also be represented as a matrix game as follows, where the attacker is the row player and the defender is the column player:

$$\begin{array}{cc} & \begin{matrix} (c) & (d) \end{matrix} \\ \begin{bmatrix} (x1, y1) & (x2, y2) \\ (x3, y3) & (x4, y4) \end{bmatrix} & \begin{matrix} (a) \\ (b) \end{matrix} \end{array}$$

If the decision variables of the players are continuous, for example, $x \in [0, a]$ and $y \in [0, c]$ denote attack and defense intensity, respectively, and then the resulting continuous-kernel game is described using functions instead of a matrix. Then, $J^{\text{attacker}}(x, y)$ and $J^{\text{defender}}(x, y)$ quantify the cost of the attacker and defender as a function of their actions, respectively.

Security Game Types

In its simplest formulation, the conflict between those defending a system and malicious attackers targeting it can be modeled as a two-person zero-sum security game, where the loss of a player is the gain of the other. Alternatively, two- and multi-person non-zero-sum security games generalize this for capturing a broader range of interactions. Static game formulations and their repeated versions are helpful for modeling myopic behavior of players in fast-changing

situations where planning future actions is of little use. In the case where the underlying system dynamics are predictable and available to the players, dynamic security game formulations can be utilized. If there is an order of actions in the game, for example, a purely reactionary defender, then leader-follower games can be used to formulate such cases where the attacker takes the lead and the defender follows.

Within the framework of security games, the concept of Nash equilibrium, where no player gains from deviating from its own Nash equilibrium strategy if others stick with theirs, provides a solid foundation. However, there are refinements and additional solution concepts when the game is dynamic or when there is more than one Nash equilibrium or information limitations in the game. These constitute an open and ongoing research topic. A related research question is the design of security games to ensure a favorable outcome from a global perspective while taking into account the independence of individual players in their decisions.

In certain cases, it is useful to analyze the player interactions in multiple layers. For example, in some security games, there may be defenders and malicious attackers trying to influence a population of other players by indirect means. In order to circumvent modeling complexity of the problem, evolutionary games have been suggested. While evolutionary games forsake modeling individual player actions, they provide valuable insights to collective behavior of populations of players and ensure tractability. Such models are useful, for example, in the analysis of various security policies affecting many users or security of large-scale critical systems.

Another important aspect of security decisions is the availability and acquisition of information on the properties of the system at hand, the actions of other players, and the incentives behind them. Clearly, the amount of information available has a direct influence on the decisions, yet acquiring information is often costly or even infeasible in some cases. Then, the decisions have to be made with partial information, and information collection becomes part of the decision process itself, creating a complex feedback loop.

Statistical or machine learning techniques and system identification are other useful methods in analysis of security games where players use the acquired information iteratively to update their own model of the environment and other players. The players then decide on their best courses of action. Existing work on fictitious play and reinforcement learning methods such as (deep) Q-learning are applicable and useful. A unique feature of security games is the fact that players try to hide their actions from others. These observability issues and distortions in observations can be captured by modeling the interaction between players who observe each other's actions as a noisy communication channel.

Applications

An early application of the decision and game-theoretic approach has been to the well-defined jamming problem, where malicious attackers aim to disrupt wireless communication between legitimate parties (Zander 1990; Kashyap et al. 2004). Detection of security intrusion and anomalies due to attacks is another problem, where the interaction between attackers and defenders has been modeled successfully using game theory (Alpcan and Başar 2011; Kodialam and Lakshman 2003; Han et al. 2017). Decision and game-theoretic approaches have been applied to a broad variety of networked system security problems such as security investments in organizations (Miura-Ko et al. 2008; Fielder et al. 2014), (location) privacy (Buttayan and Hubaux 2008; Kantarcioglu et al. 2011), distributed attack detection, attack trees and graphs, adversarial control (Altman et al. 2010), and topology planning in the presence of adversaries (Gueye et al. 2010), transportation security (Pita et al. 2009), as well as to other types of security games and decisions. Security games have been used to investigate cyberphysical security of (smart) power grid (Law et al. 2012). The well-cited survey paper Manshaei et al. (2013) presents an extensive segment of the literature on the subject.

The proceedings of the Conferences on Decision and Game Theory for Security (GameSec)

have been published as edited volumes in Lecture Notes for Computer Science (LNCS) since 2010. *GameSec* (www.gamesec-conf.org) has emerged as one of the main conferences that brings together active researchers focusing on security applications of game theory, and its proceedings contain the latest research results in this field. Increasingly, there are sessions involving papers on game theory in prominent machine learning conferences, which focus more on computational aspects of game theory often in relation to adversarial machine learning.

Analytical risk management is a related emerging research subject. Analytical methods and game theory have been applied to the field only recently but with increasing success (Guikema 2009; Mounzer et al. 2010). Another emerging topic is the adversarial mechanism design (Chorppath and Alpcan 2011; Roth 2008), where the goal is to design mechanisms resistant to malicious behavior.

Summary and Future Directions

Game theory provides quantitative methods for studying the players in security problems such as attackers, defenders, and users as well as their interaction and incentives. Hence, it facilitates making decisions on the best courses of action in addressing security problems while taking into account resource limitations, underlying incentive mechanisms, and security risks. Thus, security games and associated quantitative models have started to replace the prevalent ad hoc decision processes in a wide variety of security problems from safeguarding critical infrastructure to risk management, trust, and privacy in networked systems.

Game theory for security is a young and active research area as evidenced by the conference series “Conference on Decision and Game Theory for Security” (www.gamesec-conf.org) celebrating its tenth edition in 2019, the increasing number of journal and conference articles, as well as the published books (Alpcan and Başar 2011; Buttyan and Hubaux 2008; Tambe 2011).

Cross-References

- [Dynamic Noncooperative Games](#)
- [Evolutionary Games](#)
- [Network Games](#)
- [Stochastic Games and Learning](#)
- [Strategic Form Games and Nash Equilibrium](#)

Bibliography

- Alpcan T, Başar T (2011) Network security: a decision and game theoretic approach. Cambridge University Press. <http://www.tansu.alpcan.org/book.php>
- Altman E, Başar T, Kavitha V (2010) Adversarial control in a delay tolerant network. In: Alpcan T, Buttyán L, Baras J (eds) Decision and game theory for security. Lecture notes in computer science, vol 6442. Springer, Berlin/Heidelberg, pp 87–106. https://doi.org/10.1007/978-3-642-17197-0_6
- Buttyan L, Hubaux JP (2008) Security and cooperation in wireless networks. Cambridge University Press, Cambridge. <http://secowinet.epfl.ch>
- Chorppath AK, Alpcan T (2011) Adversarial behavior in network mechanism design. In: Proceedings of 4th International workshop on game theory in communication networks (Gamecomm). ENS, Cachan
- Fielder A, Panaousis E, Malacaria P, Hankin C, Smeraldi F (2014) Game theory meets information security management. In: IFIP international information security conference. Springer, pp 15–29
- Gueye A, Walrand J, Anantharam V (2010) Design of network topology in an adversarial environment. In: Alpcan T, Buttyán L, Baras J (eds) Decision and game theory for security. Lecture notes in computer science, vol 6442. Springer, Berlin/Heidelberg, pp 1–20. https://doi.org/10.1007/978-3-642-17197-0_1
- Guikema SD (2009) Game theory models of intelligent actors in reliability analysis: an overview of the state of the art. In: Game theoretic risk analysis of security threats. Springer, pp 1–19. <https://doi.org/10.1007/978-0-387-87767-9>
- Han Y, Chan J, Alpcan T, Leckie C (2017) Using virtual machine allocation policies to defend against co-resident attacks in cloud computing. IEEE Trans Dependable Secure Comput 14(1):95–108. <https://doi.org/10.1109/TDSC.2015.2429132>
- Kantarcioglu M, Bensoussan A, Hoe S (2011) Investment in privacy-preserving technologies under uncertainty. In: Baras J, Katz J, Altman E (eds) Decision and game theory for security. Lecture notes in computer science, vol 7037. Springer, Berlin/Heidelberg, pp 219–238. https://doi.org/10.1007/978-3-642-25280-8_17
- Kashyap A, Başar T, Srikant R (2004) Correlated jamming on MIMO Gaussian fading channels. IEEE Trans Inf Theory 50(9):2119–2123. <https://doi.org/10.1109/TIT.2004.833358>

- Kodialam M, Lakshman TV (2003) Detecting network intrusions via sampling: a game theoretic approach. In: Proceedings of 22nd IEEE conference on computer communications (Infocom), San Fransisco, vol 3, pp 1880–1889
- Law YW, Alpcan T, Palaniswami M (2012) Security games for voltage control in smart grid. In: 2012 50th annual Allerton conference on communication, control, and computing (Allerton), pp 212–219. <https://doi.org/10.1109/Allerton.2012.6483220>
- Manshaei MH, Zhu Q, Alpcan T, Başar T, Hubaux JP (2013) Game theory meets network security and privacy. *ACM Comput Surv* 45(3):25:1–25:39. <https://doi.org/10.1145/2480741.2480742>
- Miura-Ko RA, Yolken B, Bambos N, Mitchell J (2008) Security investment games of interdependent organizations. In: 46th annual Allerton conference
- Mounzer J, Alpcan T, Bambos N (2010) Dynamic control and mitigation of interdependent IT security risks. In: Proceedings of the IEEE conference on communication (ICC). IEEE Communications Society
- Pita J, Jain M, Ordóñez F, Portway C, Tambe M, Western C, Paruchuri P, Kraus S (2009) Using game theory for Los Angeles airport security. *AI Mag* 30(1):43–43
- Roth A (2008) The price of malice in linear congestion games. In: WINE'08: proceedings of the 4th international workshop on internet and network economics, pp 118–125
- Tambe M (2011) Security and game theory: algorithms, deployed systems, lessons learned. Cambridge University Press. <http://books.google.com.au/books?id=hijkJ7u209YC>
- Zander J (1990) Jamming games in slotted Aloha packet radio networks. In: IEEE military communications conference (MILCOM), vol 2, pp 830–834. <https://doi.org/10.1109/MILCOM.1990.117531>

Game Theory: A General Introduction and a Historical Overview

Tamer Başar

Coordinated Science Laboratory, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA

Abstract

This entry provides an overview of the aspects of game theory that are covered in this *Encyclopedia*, which includes a broad spectrum of

topics on static and dynamic game theory. The entry starts with a brief overview of game theory, identifying its basic ingredients, and continues with a brief historical account of the development and evolution of the field. It concludes by providing pointers to other entries in the *Encyclopedia* on game theory and a list of references.

Keywords

Game theory · Dynamic games · Historical evolution of game theory · Nash equilibrium · Stackelberg equilibrium · Cooperation · Evolutionary games

What Is Game Theory?

Game theory deals with strategic interactions among multiple decision-makers, called *players* (and in some contexts, *agents*), with each player's preference ordering among multiple alternatives captured in an objective function for that player, which she either tries to maximize (in which case the objective function is a *utility* function or a *benefit* function) or minimize (in which case we refer to the objective function as a *cost* function or a *loss* function). For a nontrivial game, the objective function of a player depends on the choices (*actions* or equivalently *decision variables*) of at least one other player, and generally of all the players, and hence a player cannot simply optimize her own objective function independent of the choices of the other players. This thus brings in a coupling among the actions of the players and binds them together in decision-making even in a noncooperative environment. If the players are able to enter into a cooperative agreement so that the selection of actions or decisions is done collectively and with full trust and so that all players would benefit to the extent possible, then we would be in the realm of *cooperative game theory*, where issues such as bargaining and characterization of *fair* outcomes, coalition formation, and excess utility distribution are of relevance; an entry in this *Encyclopedia* (by Haurie) discusses cooperation and cooperative outcomes

in the context of dynamic games. Other aspects of cooperative game theory can be found in several standard texts on game theory, such as Owen (1995), Vorob'ev (1977), or Fudenberg and Tirole (1991). See also the 2009 survey entry Saad et al. (2009), which emphasizes applications of cooperative game theory to communication networks.

If no cooperation is allowed among the players, then we are in the realm of *noncooperative game theory*, where first, one has to introduce a satisfactory solution concept. Leaving aside for the moment the issue of how the players can reach such a solution point, let us address the issue of what would be the minimum features one would expect to see there. To first order, such a solution point should have the property that if all players but one stay put, then the player who has the option of moving away from the solution point should not have any incentive to do so because she cannot improve her payoff. Note that we cannot allow two or more players to move collectively from the solution point, because such a collective move requires cooperation, which is not allowed in a noncooperative game. Such a solution point where none of the players can improve her payoff by a unilateral move is known as a *noncooperative equilibrium* or *Nash equilibrium*, named after John Nash, who introduced it and proved that it exists in finite games (i.e., games where each player has only a finite number of alternatives), over 60 years ago; see Nash (1950, 1951). This result and its various extensions for different frameworks as well as its computation (both offline and online) are discussed in several entries in this *Encyclopedia*. Another noncooperative equilibrium solution concept is the *Stackelberg equilibrium*, introduced in von Stackelberg (1934), and predating the Nash equilibrium, where there is a hierarchy in decision-making among the players, with some of the players, designated as *leaders*, having the ability to first announce their actions (and make a commitment to play them), and the remaining players, designated as *followers*, taking these actions as given in the process of computation of their noncooperative (Nash) equilibria (among themselves). Before announcing their

actions, the leaders would of course anticipate these responses and determine their actions in a way that the final outcome will be most favorable to them (in terms of their objective functions). For a comprehensive treatment of Nash and Stackelberg equilibria for different classes of games, see Başar and Olsder (1982).

We say that a noncooperative game is strictly (or genuinely) *nonzero-sum*, if the sum of the players' objective functions cannot be made zero after appropriate positive scaling and/or translation that do not depend on the players' decision variables. We say that a two-player game is *zero-sum* if the sum of the objective functions of the two players is *zero* or can be made zero by appropriate positive scaling and/or translation that do not depend on the decision variables of the players. On a broader scale (and with a possible abuse of terminology), we can view two-player zero-sum games as constituting a special subclass of two-player nonzero-sum games, and in this case, the Nash equilibrium becomes the *saddle-point equilibrium*. A game is a *finite game* if each player has only a finite number of alternatives, that is, the players pick their actions out of finite sets (action sets); otherwise the game is an *infinite game*; finite games are also known as *matrix games*. An infinite game is said to be a *continuous-kernel game* if the action sets of the players are continua and the players' objective functions are continuous with respect to action variables of all players. A game is said to be *deterministic* if the players' actions uniquely determine the outcome, as captured in the objective functions, whereas if the objective function of at least one player depends on an additional variable (state of nature) with an underlying probability distribution, then we have a *stochastic game*. A game is a *complete information game* if the description of the game (i.e., the players, the objective functions, and the underlying probability distributions (if stochastic)) is common information to all players; otherwise we have an *incomplete information game*. We say that a game is *static* if players have access to only the a priori information (shared by all) and none of the players has access to information on the actions of any of the other players; otherwise what we

have is a *dynamic game*. A game is a *single-act game* if every player acts only once; otherwise the game is *multi-act*. Note that it is possible for a single-act game to be dynamic and for a multi-act game to be static. A dynamic game is said to be a *differential game* if the evolution of the decision process (controlled by the players over time) takes place in continuous time and generally involves a differential equation; if it takes place over a discrete-time horizon, the dynamic game is sometimes called a *discrete-time game*.

In dynamic games, as the game progresses, players acquire information (complete or partial) on past actions of other players and use this information in selecting their own actions (also dictated by the equilibrium solution concept at hand). In finite dynamic games, for example, the progression of a game involves a *tree structure* (also called *extensive form*) where each node is identified with a player along with the time when she acts and branches emanating from a node show the possible moves of that particular player. A player, at any point in time, could generally be at more than one node, which is a situation that arises when the player does not have complete information on the past moves of other players and hence may not know with certainty which a particular node she is at at any particular time. This uncertainty leads to a clustering of nodes into what is called *information sets* for that player. What players decide on within the framework of the extensive form is not their actions but their *strategies*, that is, what action they would take at each information set (in other words, correspondences between their information sets and their allowable actions). They then take specific actions (or actions are executed on their behalf), dictated by the strategies chosen as well as the progression of the game (decision) process along the tree. The equilibrium is then defined in terms of not actions but strategies.

The notion of a *strategy*, as a mapping from the collection of information sets to action sets, extends readily to infinite dynamic games, and hence in both differential games and difference games, Nash equilibria are defined in terms of strategies. Several entries in this *Encyclopedia* discuss such equilibria, for both zero-sum and

nonzero-sum dynamic games, with and without the presence of probabilistic uncertainty. An extensive treatment of dynamic games can also be found in the two-volume *Handbook of Dynamic Game Theory*, whose volume I (Başar and Zaccour 2018a) covers theory and volume II (Başar and Zaccour 2018b) covers various applications.

In the broad scheme of things, game theory, and particularly noncooperative game theory, can be viewed as an extension of two fields, both covered in this *Encyclopedia: Mathematical Programming and Optimal Control Theory*. Any problem in game theory collapses to a problem in one of these disciplines if there is only one player. One-player static games are essentially mathematical programming problems (linear programming or nonlinear programming), and one-player difference or differential games can be viewed as optimal control problems.

Highlights on the History and Evolution of Game Theory

Game Theory has enjoyed over 70 years of scientific development, with the publication of the *Theory of Games and Economic Behavior* by John von Neumann and Oskar Morgenstern (1947) generally acknowledged to *kick start* the field. It has experienced incessant growth in both the number of theoretical results and the scope and variety of applications. As a recognition of the vitality of the field, through 2012, a total of ten Nobel Prizes were given in Economic Sciences for work primarily in game theory, with the first such recognition bestowed in 1994 on John Harsanyi, John Nash, and Reinhard Selten “for their pioneering analysis of equilibria in the theory of noncooperative games.” The second round of Nobel Prizes in game theory went to Robert Aumann and Thomas Schelling in 2005 “for having enhanced our understanding of conflict and cooperation through game-theory analysis.” The third round recognized Leonid Hurwicz, Eric Maskin, and Roger Myerson in 2007 “for having laid the foundations of mechanism design theory.” And the most recent one was in 2012, recognizing

Alvin Roth and Lloyd Shapley “for the theory of stable allocations and the practice of market design.” To this list of highest-level awards related to contributions to game theory, one should also add the 1999 Crafoord Prize (which is the highest prize in Biological Sciences), which went to John Maynard Smith (along with Ernst Mayr and G. Williams) “for developing the concept of evolutionary biology,” where Smith’s recognized contributions had a strong game-theoretic underpinning, through his work on evolutionary games and evolutionary stable equilibrium (Smith and Price 1973; Smith 1974, 1982); this is the topic of one of the entries in this *Encyclopedia* (by Altman). Several other “game theory” entries in the *Encyclopedia* also relate to the contributions of the Nobel Laureates mentioned above.

Even though von Neumann and Morgenstern’s 1944 book is taken as the starting point of the scientific approach to game theory, game-theoretic notions and some isolated key results date back to earlier years and even centuries. Sixteen years earlier, in 1928, John von Neumann himself had resolved completely an open fundamental problem in zero-sum games, that *every finite two-player zero-sum game admits a saddle point in mixed strategies*, which is known as the *Minimax Theorem* (von Neumann 1928) – a result which Emile Borel had conjectured to be false 8 years before. Some early traces of game-theoretic thinking can be seen in the 1802 work (*Considérations sur la théorie mathématique du jeu*) of André-Marie Ampère (1775–1836), who was influenced by the 1777 writings (*Essai d’Arithmétique Morale*) of Georges Louis Buffon (1707–1788).

Which event or writing has really started game-theoretic thinking or approach to decision-making (in law, politics, economics, operations research, engineering, etc.) may be a topic of debate, but what is indisputable is that in (zero-sum) differential games (which is most relevant to control theory), the starting point was the work of Rufus Isaacs in the RAND Corporation in the early 1950s, which remained classified for at least a decade, before being made accessible to a broad readership in 1965 (Isaacs 1965); see also the review Ho (1965) which first introduced

the book to the control community. One of the entries in this *Encyclopedia* (by Bernhard) talks about this history and the theory developed by Isaacs, within the context of pursuit-evasion games, and another entry (by Bernhard) discusses the impact the zero-sum differential game framework has made on robust control design (Başar and Bernhard 1995). Extension of the game-theoretic framework to nonzero-sum differential games with Nash equilibrium as the solution concept was initiated in Starr and Ho (1969) and with Stackelberg equilibrium as the solution concept in Simaan and Cruz (1973). Systematic study of the role information structures play in the existence of such equilibria and their uniqueness or nonuniqueness (termed *informational nonuniqueness*) was carried out in Başar (1974, 1976, 1977).

Related Entries on Game Theory in the Encyclopedia

Several entries in the *Encyclopedia* introduce various sub-areas of game theory and discuss important developments (past and present) in each corresponding area.

The entry ▶ “[Strategic Form Games and Nash Equilibrium](#)” by Ozdaglar introduces the static game framework along with the Nash equilibrium concept, for both finite and infinite games, and discusses the issues of existence and uniqueness as well as efficiency. The entry ▶ “[Dynamic Noncooperative Games](#)” by Castañón focuses on dynamic games, again for both finite and infinite games, and discusses extensive form descriptions of the underlying dynamic decision process, either as trees (in finite games) or difference equations (in discrete-time infinite games). Bernhard, in two entries, discusses continuous-time dynamic games, described by differential equations (so-called differential games), but in the two-person zero-sum case. One of these entries ▶ “[Pursuit-Evasion Games and Zero-Sum Two-Person Differential Games](#)” describes the framework initiated by Isaacs, and several of its extensions for pursuit-evasion games, and the other one ▶ “[Linear Quadratic Zero-Sum Two-Person Differential Games](#)” presents results

on the special case of linear quadratic differential games, with an important application of that framework to robust control and more precisely H^∞ -optimal control.

When the number of players in a nonzero-sum game is countably infinite, or even just sufficiently large, some simplifications arise in the computation and characterization of Nash equilibria. The mathematical framework applicable to this context is provided by *mean field theory*, which is the topic of the entry ▶ [“Mean Field Games”](#) by Caines, who discusses this relatively new theory within the context of stochastic differential games.

Cooperative solution concepts for dynamic games are discussed in the entry ▶ [“Cooperative Solutions to Dynamic Games”](#) by Haurie, who introduces Pareto optimality, the bargaining solution concept by Nash, characteristic functions, core, and C-optimality, and presents some selected results using these concepts. In the entry ▶ [“Evolutionary Games”](#) by Altman, the foundations of, as well as the recent advances in, evolutionary games are presented, along with examples showing their potential as a tool for capturing and modeling interactions in complex systems.

The entry ▶ [“Learning in Games”](#) by Shamma addresses the online computation of Nash equilibrium through an iterative process which takes into account each player’s response to choices made by the remaining players, with built-in learning and adaptation rules; one such scheme that is discussed in the entry is the well-known *fictional play*. Learning is also the topic of the entry ▶ [“Stochastic Games and Learning”](#) by Szajowski, who presents a framework and a set of results using the stochastic games formulation introduced by Shapley in the early 1950s.

The entry ▶ [“Network Games”](#) by Srikant shows how game theory plays an important role in modeling interactions between entities on a network, particularly communication networks, and presents a simple mathematical model to study one such instance, namely, resource allocation in the Internet. How to design a game so as to obtain a desired outcome (as captured by say a Nash equilibrium) is a question central to mechanism design, which is covered in the entry ▶ [“Mechanism Design”](#) by Johari, who discusses

as a specific example the Vickrey-Clarke-Groves (VCG) mechanism.

Two other applications of game theory are to the design of auctions and to security. The entry ▶ [“Auctions”](#) by Hajek addresses the former, discussing general auction theory along with equilibrium strategies and more specifically combinatorial auctions. The latter is addressed in the entry ▶ [“Game Theory for Security”](#) by Alpcan, who discusses how the game-theoretic approach leads to more effective responses to security in complex systems and organizations, with applications to a wide variety of systems and critical infrastructures such as electricity, water, financial services, and communication networks.

Future of Game Theory

The second half of the twentieth century was a golden era for game theory, and all evidence so far in the twenty-first century indicates that the next half century is destined to be a *platinum* era. In all respects game theory is on an upward slope in terms of its vitality, the wealth of topics that fall within its scope, the richness of the conceptual framework it offers, the range of applications, and the challenges it presents to an inquisitive mind.

Bibliography

- Başar T (1974) A counter example in linear-quadratic games: existence of non-linear Nash solutions. *J Optim Theory Appl* 14(4):425–430
- Başar T (1976) On the uniqueness of the Nash solution in linear-quadratic differential games. *Int J Game Theory* 5:65–90
- Başar T (1977) Informationally nonunique equilibrium solutions in differential games. *SIAM J Control* 15(4):636–660
- Başar T, Bernhard P (1995) H^∞ optimal control and related minimax design problems: a dynamic game approach, 2nd edn. Birkhäuser, Boston
- Başar T, Olsder GJ (1999) *Dynamic noncooperative game theory. Classics in applied mathematics*, 2nd edn. SIAM, Philadelphia (1st edn. Academic Press, London, 1982)
- Başar T, Zaccour G (eds) (2018a) *Handbook of dynamic game theory. Volume I: theory of dynamic games*. Cham
- Başar T, Zaccour G (eds) (2018b) *Handbook of dynamic game theory. Volume II: applications of dynamic games*. Springer, Cham

- Fudenberg D, Tirole J (1991) *Game theory*. MIT Press, Boston
- Ho Y-C (1965) Review of 'differential games' by R. Isaacs. *IEEE Trans Autom Control* AC-10(4):501–503
- Isaacs R (1975) *Differential games*, 2nd edn. Kruger, New York (1st edn.: Wiley, New York, 1965)
- Nash JF Jr (1950) Equilibrium points in n -person games. *Proc Natl Acad Sci* 36(1):48–49
- Nash JF Jr (1951) Non-cooperative games. *Ann Math* 54(2):286–295
- Owen G (1995) *Game theory*, 3rd edn. Academic Press, New York
- Saad W, Han Z, Debbah M, Hjørungnes A, Başar T (2009) Coalitional game theory for communication networks [A tutorial]. *IEEE Signal Process Mag Spec Issue Game Theory* 26(5):77–97
- Simaan M, Cruz JB Jr (1973) On the Stackelberg strategy in nonzero sum games. *J Optim Theory Appl* 11: 533–555
- Smith JM (1974) The theory of games and the evolution of animal conflicts. *J Theor Biol* 47:209–221
- Smith JM (1982) *Evolution and the theory of games*. Cambridge University Press, Cambridge, Great Britain
- Smith JM, Price GR (1973) The logic of animal conflict. *Nature* 246:15–18
- Starr AW, Ho Y-C (1969) Nonzero-sum differential games. *J Optim Theory Appl* 3:184–206
- von Neumann J (1928) *Zur theorie der Gesellschaftsspiele*. *Mathematische Annalen* 100:295–320
- von Neumann J, Morgenstern O (1947) *Theory of games and economic behavior*, 2nd edn. Princeton University Press, Princeton (1st edn.: 1944)
- von Stackelberg H (1934) *Marktform und Gleichgewicht*. Springer, Vienna (An English translation appeared in 1952 entitled "The theory of the market economy," published by Oxford University Press, Oxford)
- Vorob'ev NH (1977) *Game theory*. Springer, Berlin

Graphs for Modeling Networked Interactions

Mehran Mesbahi¹ and Magnus Egerstedt²

¹University of Washington, Seattle, WA, USA

²Georgia Institute of Technology, Atlanta, GA, USA

Abstract

Graphs constitute natural models for networks of interacting agents. This entry introduces graph theoretic formalisms that facilitate analysis and synthesis of coordinated control algorithms over networks.

Keywords

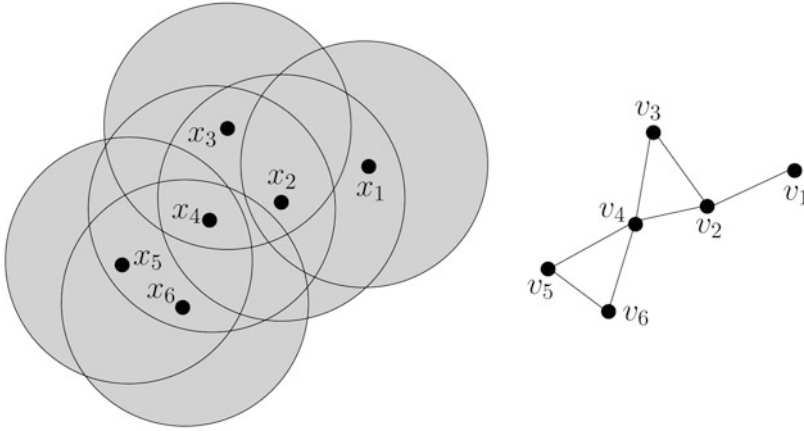
Distributed control · Graph theory · Multi-agent networks

Introduction

Distributed and networked systems are characterized by a set of dynamical units (agents, actors, nodes) that share information with each other in order to achieve a global performance objective using locally available information. Information can typically be shared if agents are within communication or sensing range of each other. It is useful to abstract away the particulars of the underlying information-exchange mechanism and simply say that an link exists between two nodes if they can share information. Such an abstraction is naturally represented in terms of a graph.

A graph is a combinatorial object defined by two constructs: vertices (or nodes) and edges (or links) connecting pairs of distinct vertices. The set of N vertices specified by $V = \{v_1, \dots, v_N\}$ corresponds to the agents, and an edge between vertices v_i and v_j is represented by (v_i, v_j) ; the set of all edges constitutes the edge set E . The *graph* G is thus the pair $G = (V, E)$ and the interpretation is that an edge $(v_i, v_j) \in E$ if information can flow from vertex i to vertex j . If the information exchange is sensor based, and if there is a state x_i associated with vertex i , e.g., its position, then this information is typically relative, i.e., the states are measured relative to each other and the information obtained along the edge is $x_j - x_i$. If, on the other hand, the information is communicated, then the full state information x_i can be transmitted along the edge (v_i, v_j) . The graph abstraction for a network of agents with sensor-based information exchange is illustrated in Fig. 1.

It is often useful to differentiate between scenarios where the information exchange is bidirectional – if agent i can get information from agent j , then agent j can get information from agent i – and when it is not. In the language of graph theory, an undirected graph is one where



Graphs for Modeling Networked Interactions, Fig. 1
A network of agents equipped with omnidirectional range sensors can be viewed as a graph (undirected in this case),

with nodes corresponding to the agents and edges to their pairwise interactions, which are enabled whenever the agents are within a certain distance from each other

$(v_i, v_j) \in E$ implies that $(v_j, v_i) \in E$, while a directed graph is one where such an implication may not hold.

cardinality of the set of edges incident on vertex i . Moreover,

$$a_{ij} = \begin{cases} 1 & \text{if } (v_j, v_i) \in E \\ 0 & \text{otherwise.} \end{cases}$$

Graph-Based Coordination Models

Graphs provide structural insights into how different coordination algorithms behave over a network. A coordination algorithm, or protocol, is an update rule that describes how the node states should evolve over time. To understand such protocols, one needs to connect the interaction dynamics to the underlying graph structure. This connection is facilitated through the common intersection of linear system theory and graph theory, namely, the broad discipline of *algebraic graph theory*, by first associating matrices with graphs. For undirected graphs, the following matrices play a key role:

Degree matrix : $\Delta = \text{Diag}(\deg(v_1), \dots, \deg(v_N))$,

Adjacency matrix : $A = [a_{ij}]$,

where **Diag** denotes a diagonal matrix whose diagonal consists of its argument and $\deg(v_i)$ is the degree of vertex i in the graph, i.e., the

As an example of how these matrices come into play, the so-called *consensus* protocol over scalar states can be compactly written on ensemble form as

$$\dot{x} = -Lx, \quad (1)$$

where $x = [x_1, \dots, x_N]^T$ and L is the *graph Laplacian*:

$$L = \Delta - A.$$

A useful matrix for directed networks is the incidence matrix, obtained by associating an index with each edge in E . We say that $v_i = \text{tail}(e_j)$ if edge e_j starts at node v_i and $v_i = \text{head}(e_j)$ if e_j ends up at v_i , leading to the

Incidence matrix : $D = [d_{ij}]$,

where

$$d_{ij} = \begin{cases} 1 & \text{if } v_i = \text{head}(e_j) \\ -1 & \text{if } v_i = \text{tail}(e_j) \\ 0 & \text{otherwise.} \end{cases}$$

It now follows that for undirected networks, the Laplacian has an equivalent representation as

$$L = DD^T,$$

where D is the incidence matrix associated with an arbitrary orientation (assignment of directions to the edges) of the undirected graph. This in turn implies that for undirected networks, L is a positive semi-definite matrix and that all of its eigenvalues are nonnegative.

If the network is directed, one has to pay attention to the direction in which information is flowing, using the *in-degree* and *out-degree* of the vertices. The out-degree of vertex i is the number of directed edges that originate at i , and similarly the in-degree of node i is the number of directed edges that terminate at node i . A directed graph is *balanced* if the out-degree is equal to the in-degree at every vertex in the graph. And, the graph Laplacian for directed graphs is obtained by only counting information flowing in the correct direction, i.e., if $L = [\ell_{ij}]$, then ℓ_{ii} is the in-degree of vertex i and $\ell_{ij} = -1$ if $i \neq j$ and $(v_j, v_i) \in E$.

As a final note, for both directed and undirected networks, it is possible to associate weights with the edges, $w : E \rightarrow \Upsilon$ where Υ is a set of nonnegative reals or more generally a field, in which case the Laplacian's diagonal elements are the sum of the weights of edges incident to node i and the off-diagonal elements are $-w(v_j, v_i)$ when $(v_j, v_i) \in E$.

Applications

Graph-based coordination has been used in a number of application domains, such as multi-agent robotics, mobile sensor and communication networks, formation control, and biological systems. One way in which the consensus protocol can be generalized is by defining an edge-tension energy $E_{ij}(\|x_i - x_j\|)$ along each edge in the graph, which gives the total energy in the network as

$$E(x) = \sum_{i=1}^N \sum_{j \in N_i} E_{ij}(\|x_i - x_j\|).$$

If the agents update their states in such a way as to reduce the total energy in the system according to a gradient descent scheme, the update law becomes

$$\dot{x} = -\frac{\partial E(x)}{\partial x_i} \Rightarrow \dot{E}(x) = -\left\| \frac{\partial E(x)}{\partial x} \right\|_2^2,$$

which is nonpositive, i.e., the total energy is reduced (or at least not increased) in the network. For undirected networks, the ensemble version of this protocol assumes the form

$$\dot{x} = -L_w(x)x,$$

where the weighted graph Laplacian is

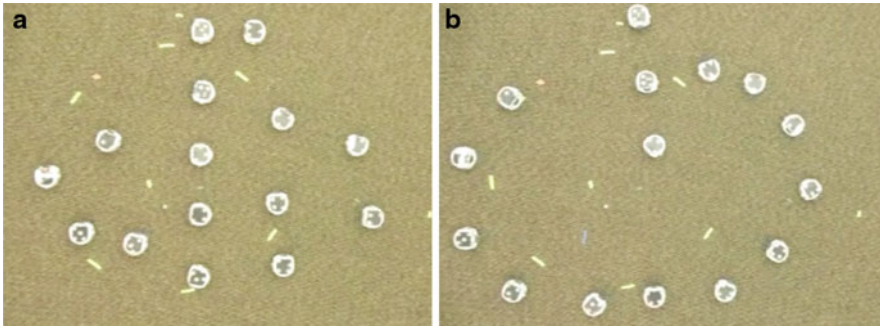
$$L_w(x) = DW(x)D^T,$$

with the weight matrix $W(x) = \text{Diag}(w_1(x), \dots, w_M(x))$. Here M is the total number of edges in the network, and $w_k(x)$ is the weight that corresponds to the k th edge, given an arbitrary ordering of the edges consistent with the incident matrix D . This energy interpretation allows for the synthesis of coordination laws for multi-agent networks with desirable properties, such as $E_{ij}(\|x_i - x_j\|) = (\|x_i - x_j\| - d_{ij})^2$ for making the agents reach the desired interagent distances d_{ij} , as shown in Fig. 2. Other applications where these types of constructions have been used include collision avoidance and connectivity maintenance.

Another set of applications involve graphs as the backbone for mapping inputs to outputs. In this case, group coordination facilitated by information diffusion on the graph has to be balanced with stability and performance criteria for such systems. For example, when an augmented version of the model (1) is subject to a forcing term as

$$\dot{x} = -Lx - D\mu - \nabla f, \quad \dot{\mu} = D^T x,$$

it leads to a (continuous) primal-dual algorithm for minimizing a sum of differentiable functions on $\Re^n$,



Graphs for Modeling Networked Interactions, Fig. 2 Fifteen mobile robots are forming the letter “G” by executing a weighted version of the consensus protocol. (a) Formation control ($t = 0$). (b) Formation control ($t = 5$)

$$J = \sum_{i=1}^n f_i(x),$$

a canonical model in distributed optimization; in this case, ∇f represents the vector of partial derivatives of f_i with respect to the i th coordinate of its argument, for $i = 1, 2, \dots, n$.

One can also consider inputs to the graph by the human operator or adversarial agents; in this case, system theoretic features of the resulting input-output system, such as its controllability, observability, and performance measure, assume an elegant graph theoretic character.

Summary and Future Directions

A number of issues pertaining to graph-based distributed control remain to be resolved. These include how heterogeneous networks, i.e., networks comprising of agents with different capabilities, can be designed, understood, and controlled. A variation to this theme is networks of networks, i.e., networks that are loosely coupled together and that must coordinate at a higher level of abstraction. Other key issues concern how human operators should interact with networked control systems and how network can be endowed with a distributed learning capability.

Cross-References

- [Averaging Algorithms and Consensus](#)
- [Distributed Optimization](#)

- [Dynamic Graphs, Connectivity of](#)
- [Flocking in Networked Systems](#)
- [Networked Systems](#)
- [Optimal Deployment and Spatial Coverage](#)
- [Oscillator Synchronization](#)
- [Vehicular Chains](#)

Recommended Reading

There are a number of research manuscripts and textbooks that explore the role of network structure on the system theoretic aspects of networked dynamic systems and its many ramifications. Some of these references are listed below.

Bibliography

- Bai H, Arcak M, Wen J (2011) Cooperative control design: a systematic, passivity-based approach. Springer, Berlin
- Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach Learn* 1:1–122
- Bullo F (2018) Lectures on network systems. CreateSpace, PlatformScotts Valley
- Fagnani F, Frasca P (2018) Introduction to averaging dynamics over networks. Springer, Berlin
- Fuhrmann PA, Helmke U (2015) The mathematics of networks of linear systems. Springer, Berlin
- Mesbahi M, Egerstedt M (2010) Graph theoretic methods in multiagent networks. Princeton University Press, Princeton
- Ren W, Beard R (2008) Distributed consensus in multi-vehicle cooperative control. Springer, Berlin

H₂ Optimal Control

Ben M. Chen

Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Shatin/Hong Kong, China

Abstract

An optimization-based approach to linear feedback control system design uses the H_2 norm, or energy of the impulse response, to quantify closed-loop performance. In this entry, an overview of state-space methods for solving H_2 optimal control problems via Riccati equations and matrix inequalities is presented in a continuous-time setting. Both regular and singular problems are considered. Connections to so-called LQR and LQG control problems are also described.

Keywords

Feedback control · H_2 control · Linear matrix inequalities · Linear systems · Riccati equations · State-space methods

Introduction

Modern multivariable control theory based on state-space models is able to handle multi-feedback-loop designs, with the added benefit

that design methods derived from it are amenable to computer implementation. Indeed, over the last five decades, a number of multivariable analysis and design have been developed using the state-space description of systems. Of these design tools, H_2 optimal control problems involve minimizing the H_2 norm of the closed-loop transfer function from exogenous disturbance signals to a pertinent controlled output signals of a given plane by appropriate use of an internally stabilizing feedback controller. It was not until the 1990s that a complete solution to the general H_2 optimal control problem began to emerge. To elaborate on this, let us concentrate our discussion on H_2 optimal control for a continuous-time system Σ expressed in the following state-space form:

$$\dot{x} = Ax + Bu + Ew \quad (1)$$

$$y = C_1x + D_{11}u + D_{1w} \quad (2)$$

$$z = C_2x + D_{2u} + D_{2w} \quad (3)$$

where x is the state variable; u is the control input; w is the exogenous disturbance input; y is the measurement output; and z is the controlled output. The system Σ is typically an augmented or generalized plant model including weighting functions that reflect design requirements. The H_2 optimal control problem is to find an appropriate control law, relating the control input u to the measured output y , such that when it is applied to the given plant in Eqs. (1), (2) and (3), the resulting closed-loop system is internally

stable and the H_2 norm of the resulting closed-loop transfer matrix from the disturbance input w to the controlled output z , denoted by $T_{zw}(s)$, is minimized. For a stable transfer matrix $T_{zw}(s)$, the H_2 norm is defined as

$$\|T_{zw}\|_2 = \left(\frac{1}{2\pi} \operatorname{trace} \left[\int_{-\infty}^{\infty} T_{zw}(j\omega) T_{zw}^H(j\omega) d\omega \right] \right)^{\frac{1}{2}} \quad (4)$$

where T_{zw}^H is the conjugate transpose of T_{zw} . Note that the H_2 norm is equal to the energy of the impulse response associated with $T_{zw}(s)$ and this is finite only if the direct feedthrough term of the transfer matrix is zero.

It is standard to make the following assumptions on the problem data: $D_{11} = 0$; $D_{22} = 0$; (A, B) is stabilizable; (A, C_1) is detectable. The last two assumptions are necessary for the existence of an internally stabilizing control law. The first assumption can be made without loss of generality via a constant loop transformation. Finally, either the assumption $D_{22} = 0$ can be achieved by a pre-static feedback law or the problem does not yield a solution that has finite H_2 closed-loop norm.

There are two main groups into which all H_2 optimal control problems can be divided. The first group, referred to as regular H_2 optimal control problems, consists of those problems for which the given plant satisfies two additional assumptions:

1. the subsystem from the control input to the controlled output, i.e., (A, B, C_2, D_2) , has no invariant zeros on the imaginary axis, and its direct feedthrough matrix, D_2 , is injective (i.e., it is tall and of full rank); and
2. the subsystem from the exogenous disturbance to the measurement output, i.e., (A, E, C_1, D_1) , has no invariant zeros on the imaginary axis, and its direct feedthrough matrix, D_1 , is surjective (i.e., it is fat and of full rank).

Assumption 1 implies that (A, B, C_2, D_2) is left invertible with no infinite zero, and Assumption 2 implies that (A, E, C_1, D_1) is right invertible with no infinite zero. The second, referred to as

singular H_2 optimal control problems, consists of those which are not regular.

Most of the research in the literature was expended on regular problems. Also, most of the available textbooks and review articles, see, for example, Anderson and Moore (1989), Bryson and Ho (1975), Fleming and Rishel (1975), Kailath (1974), Kwakernaak and Sivan (1972), Lewis (1986), and Zhou et al. (1996), to name a few, cover predominantly only a subset of regular problems. The singular H_2 control problem with state feedback was studied in Geerts (1989) and Willems et al. (1986). Using different classes of state and measurement feedback control laws, Stoorvogel et al. (1993) studied the general H_2 optimal control problems for the first time. In particular, necessary and sufficient conditions are provided therein for the existence of a solution in the case of state feedback control and in the case of measurement feedback control. Following this, Trentelman and Stoorvogel (1995) explored necessary and sufficient conditions for the existence of an H_2 optimal controller within the context of discrete-time and sampled-data systems. At the same time, Chen et al. (1993) and Chen et al. (1994a) provided a thorough treatment of the H_2 optimal control problem with state feedback controllers. This includes a parameterization and construction of the set of all H_2 optimal controllers and the associated sets of H_2 optimal fixed modes and H_2 optimal fixed decoupling zeros. Also, they provided a computationally feasible design algorithm for selecting an H_2 optimal state feedback controller that places the closed-loop poles at desired locations whenever possible. Furthermore, Chen and Saberi (1993) and Chen et al. (1996) developed the necessary and sufficient conditions for the uniqueness of an H_2 optimal controller. Interested readers are referred to the textbook (Saberi et al. 1995) for a detailed treatment of H_2 optimal control problems in their full generality.

Regular Case

Solving regular H_2 optimal control problems is relatively straightforward. In the case that all of

the state variables of the given plant are available for feedback, i.e., $y = x$, and Assumption 1 holds, the corresponding H_2 optimal control problem can be solved in terms of the unique positive semi-definite stabilizing solution $P \geq 0$ of the following algebraic Riccati equation:

$$A^T P + P A + C_2^T C_2 - (P B + C_2^T D_2)(D_2^T D_2)^{-1} (D_2^T C_2 + B^T P) = 0 \quad (5)$$

The H_2 optimal state feedback law is given by

$$u = Fx = -(D_2^T D_2)^{-1} (D_2^T C_2 + B^T P) x \quad (6)$$

and the resulting closed-loop transfer matrix from w to z , $T_{zw}(s)$, has the following property:

$$\|T_{zw}\|_2 = \sqrt{\text{trace}(E^T P E)} \quad (7)$$

Note that the H_2 optimal state feedback control law is generally nonunique. A trivial example is the case when $E = 0$, whereby every stabilizing control law is an optimal solution. It is also interesting to note that the closed-loop system comprising the given plant with $y = x$ and the state feedback control law of Eq. (6) has poles at all the stable invariant zeros and all the mirror images of the unstable invariant zeros of (A, B, C_2, D_2) together with some other fixed locations in the left half complex plane. More detailed results about the optimal fixed modes and fixed decoupling zeros for general H_2 optimal control can be found in Chen et al. (1993).

It can be shown that the well-known linear quadratic regulation (LQR) problem can be reformulated as a regular H_2 optimal control problem. For a given plant

$$\dot{x} = Ax + Bu, \quad x(0) = X_0 \quad (8)$$

with (A, B) being stabilizable, the LQR problem is to find a control law $u = Fx$ such that the following performance index is minimized:

$$J = \int_0^\infty (x^T Q_\star x + u^T R_\star u) dt, \quad (9)$$

where $R_\star > 0$ and $Q_\star \geq 0$ with $(A, Q_\star^{\frac{1}{2}})$ being detectable. The LQR problem is equivalent to finding a static state feedback H_2 optimal control law for the following auxiliary plant Σ_{LQR} :

$$\dot{x} = Ax + Bu + X_0 w \quad (10)$$

$$y = x \quad (11)$$

$$z = \begin{pmatrix} 0 \\ Q_\star^{\frac{1}{2}} \end{pmatrix} x + \begin{pmatrix} R_\star^{\frac{1}{2}} \\ 0 \end{pmatrix} u \quad (12)$$

For the measurement feedback case with both Assumptions 1 and 2 being satisfied, the corresponding H_2 optimal control problem can be solved by finding a positive semi-definite stabilizing solution $P \geq 0$ for the Riccati equation given in Eq. (5) and a positive semi-definite stabilizing solution $Q \geq 0$ for the following Riccati equation:

$$Q A^T + A Q + E E^T - (Q C_1^T + E D_1^T) (D_1 D_1^T)^{-1} (D_1 E^T + C_1 Q) = 0 \quad (13)$$

The H_2 optimal measurement feedback law is given by

$$\dot{v} = (A + B F + K C_1) v - K y, \quad u = F x \quad (14)$$

where F is as given in Eq. (6) and

$$K = -(Q C_1^T + E D_1^T) (D_1 D_1^T)^{-1} \quad (15)$$

In fact, such an optimal control law is unique, and the resulting closed-loop transfer matrix from w to z , $T_{zw}(s)$, has the following property:

$$\|T_{zw}\|_2 = \left\{ \text{trace}(E^T P E) + \text{trace} \left[\left(A^T P + P A + C_2^T C_2 \right) Q \right] \right\}^{\frac{1}{2}} \quad (16)$$

Similarly, consider the standard LQG problem for the following system

$$\dot{x} = Ax + Bu + G_\star d \quad (17)$$

$$y = Cx + N_\star n, \quad N_\star > 0 \quad (18)$$

$$z = \begin{pmatrix} H_\star x \\ R_\star u \end{pmatrix}, \quad R_\star > 0, \quad w = \begin{pmatrix} d \\ n \end{pmatrix} \quad (19)$$

where x is the state, u is the control, d and n are white noises with identity covariance, and y is the measurement output. It is assumed that (A, B) is stabilizable and (A, C) is detectable. The control objective is to design an appropriate control law that minimizes the expectation of $|z|^2$. Such an LQG problem can be solved via the H_2 optimal control problem for the following auxiliary system Σ_{LQG} , see Doyle (1983):

$$\dot{x} = Ax + Bu + [G_\star \ 0]w \quad (20)$$

$$y = Cx + [0 \ N_\star]w \quad (21)$$

$$z = \begin{pmatrix} H_\star \\ 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ R_\star \end{pmatrix} u \quad (22)$$

H_2 optimal control problem for discrete-time systems can be solved in a similar way via the corresponding discrete-time algebraic Riccati equations. It is worth noting that many works can be found in the literature that deal with solutions to discrete-time algebraic Riccati equations related to optimal control problems; see, for example, Kucera (1972), Pappas et al. (1980), and Silverman (1976), to name a few. It is proven in Chen et al. (1994b) that solutions to the discrete- and continuous-time algebraic Riccati equations for optimal control problems can be unified. More specifically, the solution to a discrete-time Riccati equation can be done through solving an equivalent continuous-time one and vice versa.

Singular Case

As in the previous section, only the key procedure in solving the singular H_2 -optimization problem for continuous-time systems is addressed. For the singular problem, it is generally not possible to obtain an optimal solution, except for some situations when the given plant satisfies certain geometric constraints, see, e.g., Chen et al. (1993)

and Stoorvogel et al. (1993). It is more feasible to find a suboptimal control law for the singular problem, i.e., to find an appropriate control law such that the H_2 norm of the resulting closed-loop transfer matrix from w to z can be made arbitrarily close to the best possible performance. The procedure given below is to transform the original problem into an H_2 almost disturbance decoupling problem; see Stoorvogel (1992) and Stoorvogel et al. (1993).

Consider the given plant in Eqs. (1), (2) and (3) with Assumption 1 and/or Assumption 2 not satisfied. First, find the largest solution $P \geq 0$ for the following linear matrix inequality

$$F(P) = \begin{pmatrix} A^T P + P A + C_2^T C_2 & P B + C_2^T D_2 \\ B^T P + D_2^T C_2 & D_2^T D_2 \end{pmatrix} \geq 0 \quad (23)$$

and find the largest solution $Q \geq 0$ for

$$G(Q) = \begin{pmatrix} A Q + Q A^T + E E^T & Q C_1^T + E D_1^T \\ C_1 Q + D_1 E^T & D_1 D_1^T \end{pmatrix} \geq 0 \quad (24)$$

Note that by decomposing the quadruples (A, B, C_2, D_2) and (A, E, C_1, D_1) into various subsystems in accordance with their structural properties, solutions to the above linear matrix inequalities can be obtained by solving a Riccati equation similar to those in Eq. (5) or Eq. (13) for the regular case. In fact, for the regular problem, the largest solution $P \geq 0$ for Eq. (23) and the stabilizing solution $P \geq 0$ for Eq. (5) are identical. Similarly, the largest solution $Q \geq 0$ for Eq. (24) and the stabilizing solution $Q \geq 0$ for Eq. (13) are also the same. Interested readers are referred to Stoorvogel et al. (1993) for more details or to Chen et al. (2004) for a more systematic treatment on the structural decomposition of linear systems and its connection to the solutions of the linear matrix inequalities.

It can be shown that the best achievable H_2 norm of the closed-loop transfer matrix from w to z , i.e., the best possible performance over all internally stabilizing control laws, is given by

$$\gamma_2^* = \left\{ \text{trace}(E^T P E) + \text{trace} \left[\left(A^T P + P A + C_2^T C_2 \right) Q \right] \right\}^{\frac{1}{2}} \quad (25)$$

Next, partition

$$F(P) = \begin{pmatrix} C_P^T \\ D_P^T \end{pmatrix} (C_P \ D_P) \quad \text{and} \quad G(Q) = \begin{pmatrix} E_Q \\ D_Q \end{pmatrix} (E_Q^T \ D_Q^T) \quad (26)$$

where $[C_P \ D_P]$ and $[E_Q^T \ D_Q^T]$ are of maximal rank, and then define an auxiliary system Σ_{PQ} :

$$\dot{x}_{PQ} = A x_{PQ} + B u + E_Q w_{PQ} \quad (27)$$

$$y = C_1 x_{PQ} + D_Q w_{PQ} \quad (28)$$

$$z_{PQ} = C_P x_{PQ} + D_P u \quad (29)$$

It can be shown that the quadruple (A, B, C_P, D_P) is right invertible and has no invariant zeros in the open right-half complex plane and the quadruple (A, E_Q, C_1, D_Q) is left invertible and has no invariant zeros in the open right-half complex plane. It can also be shown that there exists an appropriate control law such that when it is applied to Σ_{PQ} , the resulting closed-loop system is internally stable and the H_2 norm of the closed-loop transfer matrix from w_{PQ} to z_{PQ} can be made arbitrarily small. Equivalently, H_2 almost disturbance decoupling problem for Σ_{PQ} is solvable.

More importantly, it can further be shown that if an appropriate control law that solves the H_2 almost disturbance decoupling problem for Σ_{PQ} , then it solves the H_2 suboptimal problem for Σ . As such, the solution to the singular H_2 control problem for Σ can be done by finding a solution to the H_2 almost disturbance decoupling problem for Σ_{PQ} . There are vast results available in the literature dealing with disturbance decoupling problems. More detailed treatments can be found in Saberi et al. (1995).

Conclusion

This entry considers the basic solutions to H_2 optimal control problems for continuous-time systems. Both the regular problem and the general singular problem are presented. Readers interested in more details are referred to Saberi et al. (1995) and the references therein for the complete treatment of H_2 optimal control problems and to Chapter 10 of Chen et al. (2004) for the unification and differentiation of H_2 control, H_∞ control, and disturbance decoupling control problems. H_2 optimal control is a mature area and has a long history. Possible future research includes issues on how to effectively utilize the theory in solving real-life problems.

Cross-References

- [H-Infinity Control](#)
- [Linear Quadratic Optimal Control](#)
- [Optimal Control via Factorization and Model Matching](#)
- [Stochastic Linear-Quadratic Control](#)

Bibliography

- Anderson BDO, Moore JB (1989) Optimal control: linear quadratic methods. Prentice Hall, Englewood Cliffs
- Bryson AE, Ho YC (1975) Applied optimal control, optimization, estimation, and control. Wiley, New York
- Chen BM, Saberi A (1993) Necessary and sufficient conditions under which an H_2 -optimal control problem has a unique solution. Int J Control 58:337–348
- Chen BM, Saberi A, Sannuti P, Shamash Y (1993) Construction and parameterization of all static and dynamic H_2 -optimal state feedback solutions, optimal fixed modes and fixed decoupling zeros. IEEE Trans Autom Control 38:248–261
- Chen BM, Saberi A, Shamash Y, Sannuti (1994a) Construction and parameterization of all static and dynamic H_2 -optimal state feedback solutions for discrete time systems. Automatica 30:1617–1624
- Chen BM, Saberi A, Shamash Y (1994b) A non-recursive method for solving the general discrete time algebraic Riccati equation related to the H_∞ control problem. Int J Robust Nonlinear Control 4:503–519
- Chen BM, Saberi A, Shamash Y (1996) Necessary and sufficient conditions under which a discrete time H_2 -optimal control problem has a unique solution. J Control Theory Appl 13:745–753

- Chen BM, Lin Z, Shamash Y (2004) Linear systems theory: a structural decomposition approach. Birkhäuser, Boston
- Doyle JC (1983) Synthesis of robust controller and filters. In: 22nd IEEE Conference on Decision and Control, San Antonio
- Fleming WH, Rishel RW (1975) Deterministic and stochastic optimal control. Springer, New York
- Geerts T (1989) All optimal controls for the singular linear quadratic problem without stability: a new interpretation of the optimal cost. *Linear Algebra Appl* 122:65–104
- Kailath T (1974) A view of three decades of linear filtering theory. *IEEE Trans Inf Theory* 20:146–180
- Kucera V (1972) The discrete Riccati equation of optimal control. *Kybernetika* 8:430–447
- Kwakernaak H, Sivan R (1972) Linear optimal control systems. Wiley, New York
- Lewis FL (1986) Optimal control. Wiley, New York
- Pappas T, Laub AJ, Sandell NR (1980) On the numerical solution of the discrete-time algebraic Riccati equation. *IEEE Trans Autom Control* AC-25:631–641
- Saberi A, Sannuti P, Chen BM (1995) H_2 optimal control. Prentice Hall, London
- Silverman L (1976) Discrete Riccati equations: alternative algorithms, asymptotic properties, and system theory interpretations. *Control Dyn Syst* 12:313–386
- Stoorvogel AA (1992) The singular H_2 control problem. *Automatica* 28:627–631
- Stoorvogel AA, Saberi A, Chen BM (1993) Full and reduced order observer based controller design for H_2 -optimization. *Int J Control* 58:803–834
- Trentelman HL, Stoorvogel AA (1995) Sampled-data and discrete-time H_2 optimal control. *SIAM J Control Optim* 33:834–862
- Willems JC, Kitapci A, Silverman LM (1986) Singular optimal control: a geometric approach. *SIAM J Control Optim* 24:323–337
- Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River

H-Infinity Control

Keith Glover
Department of Engineering, University of
Cambridge, Cambridge, UK

Abstract

The area of robust control, where the performance of a feedback system is designed to be robust to uncertainty in the plant being controlled, has received much attention since

the 1980s. System analysis and controller synthesis based on the H-infinity norm has been central to progress in this area. This article outlines how the control law that minimizes the H-infinity norm of the closed-loop system can be derived. Connections to other problems, such as game theory and risk-sensitive control, are discussed and finally appropriate problem formulations to produce “good” controllers using this methodology are outlined.

Keywords

Loop-shaping · Robust control · Robust stability

Introduction

The $\mathcal{H}_\infty$ -norm probably first entered the study of robust control with the observations made by Zames (1981) in the considering optimal sensitivity. The so-called $\mathcal{H}_\infty$ methods were subsequently developed and are now routinely available to control engineers. In this entry we consider the $\mathcal{H}_\infty$ methods for control, and for simplicity of exposition, we will restrict our attention to linear, time-invariant, finite dimensional, continuous-time systems. Such systems can be represented by their transfer function matrix, $G(s)$, which will then be a rational function of s . Although the Hardy Space, $\mathcal{H}_\infty$, also includes nonrational functions, a rational $G(s)$ is in $\mathcal{H}_\infty$ if and only if it is proper and all its poles are in the open left half plane, in which case the $\mathcal{H}_\infty$ -norm is defined as:

$$\|G(s)\|_\infty = \sup_{\text{Res} > 0} \sigma_{\max}(G(s)) = \sup_{-\infty < \omega < \infty} \sigma_{\max}(G(j\omega))$$

(where $\sigma_{\max}$ denotes the largest singular value). Hence for a single input/single output system with transfer function, $g(s)$, its $\mathcal{H}_\infty$ -norm, $\|g(s)\|_\infty$ gives the maximum value of $|g(j\omega)|$ and hence the maximum amplification of sinusoidal signals by a system with this transfer function. In the multi-input/multi-output case a similar result holds regarding the system

amplification of a vector of sinusoids. There is now a good collection of graduate level textbooks that cover the area in some detail from a variety of approaches, and these are listed in the Recommended Reading section and the references in this article are generally to these texts rather than to the original journal papers.

Consider a system with transfer function, $G(s)$, input vector, $u(t) \in \mathcal{L}_2(0, \infty)$ and an output vector, $y(t)$, whose Laplace transforms are given by $\bar{u}(s)$ and $\bar{y}(s)$. Such a system will have a state space realization,

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

giving $G(s) = D + C(sI - A)^{-1}B$, which we also denote

$$G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right],$$

and hence $\bar{y}(s) = G(s)\bar{u}(s)$ if $x(0) = 0$.

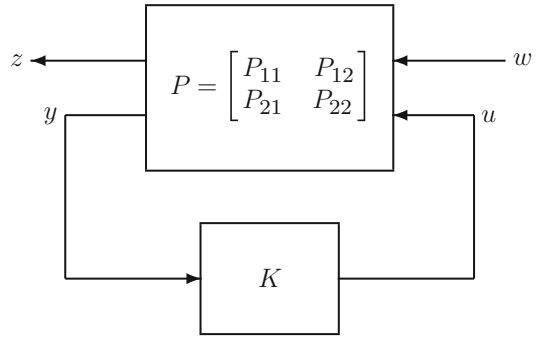
There are two main reasons for using the $\mathcal{H}_\infty$ -norm. Firstly in representing the system gain for input signals $u(t) \in \mathcal{L}_2(0, \infty)$ or equivalently $\bar{u}(j\omega) \in \mathcal{L}_2(-\infty, \infty)$, with corresponding norm $\|u\|_2^2 = \int_0^\infty u(t)^* u(t) dt$ (where x^* denotes the conjugate transpose of the vector x (or a matrix)). With these input and output spaces the induced norm of the system is easily shown to be the $\mathcal{H}_\infty$ -norm of $G(s)$, and in particular,

$$\|y\|_2 \leq \|G(s)\|_\infty \|u\|_2$$

Hence in a control context the $\mathcal{H}_\infty$ -norm can give a measure of the gain, for example, from disturbances to the resulting errors. In the interconnection of systems, the property that $\|P(s)Q(s)\|_\infty \leq \|P(s)\|_\infty \|Q(s)\|_\infty$ is often useful.

The second reason for using the $\mathcal{H}_\infty$ -norm is in representing uncertainty in the plant being controlled, e.g., the nominal plant is $P_o(s)$ but the actual plant is $P(s) = P_o(s) + \Delta(s)$ where $\|\Delta(s)\|_\infty \leq \delta$.

A typical control design problem is given in Fig. 1, i.e.,



H-Infinity Control, Fig. 1 Lower linear fractional transformation: feedback system

$$\begin{bmatrix} \bar{z} \\ \bar{y} \end{bmatrix} = P \begin{bmatrix} \bar{w} \\ \bar{u} \end{bmatrix} = \begin{bmatrix} P_{11}\bar{w} + P_{12}\bar{u} \\ P_{21}\bar{w} + P_{22}\bar{u} \end{bmatrix}$$

$$\bar{u} = K\bar{y}$$

$$\Rightarrow \bar{y} = (I - P_{22}K)^{-1}P_{21}\bar{w},$$

$$\bar{u} = K(I - P_{22}K)^{-1}P_{21}\bar{w}$$

$$\bar{z} = \left(P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} \right) \bar{w}$$

$$=: \mathcal{F}_l(P, K)\bar{w} =: T_{z \leftarrow w}\bar{w}$$

where $\mathcal{F}_l(P, K)$ denotes the lower Linear Fractional Transformation (LFT) with connection around the lower terminals of P as in Fig. 1.

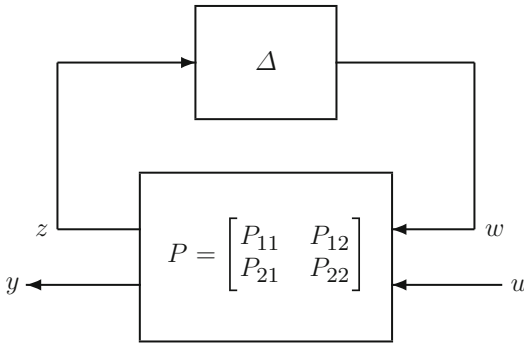
The standard $\mathcal{H}_\infty$ -control synthesis problem is to find a controller with transfer function, K , that

stabilizes the closed-loop system in Fig. 1 and minimizes $\|\mathcal{F}_l(P, K)\|_\infty$.

That is, the controller is designed to minimize the worst-case effect of the disturbance w on the output/error signal z as measured by the $\mathcal{L}_2$ norm of the signals. This article will describe the solution to this problem.

Robust Stability

Before we describe the solution to the synthesis problem, consider the problem of the robust stability of an uncertain plant with a feedback controller. Suppose the plant is given by the upper



H-Infinity Control, Fig. 2 Upper linear fractional transformation

LFT, $\mathcal{F}_u(P, \Delta)$ with $\|\Delta\|_\infty \leq 1/\gamma$ as illustrated in Fig. 2,

$$\bar{y} = \mathcal{F}_u(P, \Delta)\bar{u}, \quad (1)$$

$$\text{where } \mathcal{F}_u(P, K) := P_{22} + P_{21}\Delta(I - P_{11}\Delta)^{-1}P_{12} \quad (2)$$

The *small gain theorem* then states that the feedback system of Fig. 3 will be stable for all such Δ if the feedback connection of P_{22} and K is stable and $\|\mathcal{F}_l(P, K)\|_\infty < \gamma$. This robust stability result is valid if P and Δ are both stable; more care is required when either or both are unstable but with such care a similar result is true.

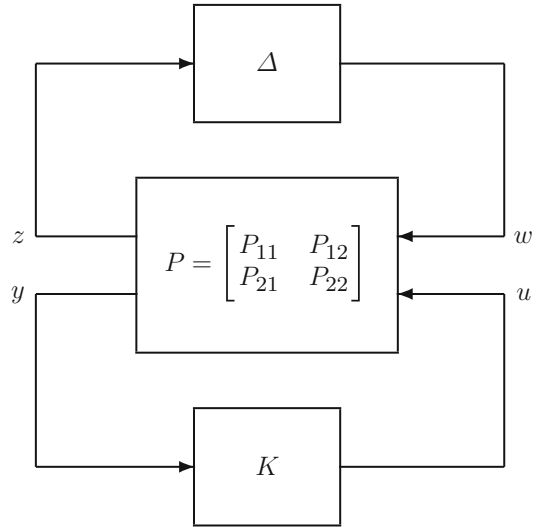
Let us consider a couple of examples. First suppose that the uncertainty is represented as output multiplicative uncertainty,

$$P_\Delta = (I + W_1\Delta W_2)P_o = \mathcal{F}_u\left(\begin{bmatrix} 0 & W_2P_o \\ W_1 & P_o \end{bmatrix}, \Delta\right)$$

with robust stability test given by

$$\begin{aligned} & \|\mathcal{F}_l\left(\begin{bmatrix} 0 & W_2P_o \\ W_1 & P_o \end{bmatrix}, K\right)\|_\infty \\ &= \|W_2P_oK(I - P_oK)^{-1}W_1\|_\infty < \gamma \end{aligned}$$

As a second example consider the plants $P_\Delta = (\tilde{M} + \Delta_M)^{-1}(\tilde{N} + \Delta_N)$, with $\Delta = \begin{bmatrix} \Delta_N & \Delta_M \end{bmatrix}$ and $\|\Delta\|_\infty \leq 1/\gamma$. Here $P_o = \tilde{M}^{-1}\tilde{N}$ is a left coprime factorization of the nominal plant and the plants P_Δ are represented by perturbations



H-Infinity Control, Fig. 3 Feedback system with plant uncertainty

to these coprime factors. In this case $P_\Delta = \mathcal{F}_u(P, \Delta)$, where

$$P = \begin{bmatrix} \begin{bmatrix} 0 \\ -\tilde{M}^{-1} \end{bmatrix} \\ \tilde{M}^{-1} \end{bmatrix} \begin{bmatrix} I \\ -\tilde{M}^{-1}\tilde{N} \end{bmatrix}$$

and the robust stability test will be

$$\|\mathcal{F}_l(P, K)\|_\infty = \left\| \begin{bmatrix} K \\ -I \end{bmatrix} (I - P_oK)^{-1}\tilde{M}^{-1} \right\|_\infty < \gamma$$

This is related to plant perturbations in the gap metric (see Vinnicombe 2001). It is therefore observed that the robust stability test for these useful representations of uncertain plants is given by an $\mathcal{H}_\infty$ -norm test just as in the controller synthesis problem.

Derivation of the $\mathcal{H}_\infty$ -Control Law

In this section we present a solution to the $\mathcal{H}_\infty$ -control problem and give some interpretations of the solution. The approach presented is as in by Doyle et al. (1989); see also Zhou et al. (1996). We will make some simplifying structural assumptions to make the formulae less complex

and will *not* state the required assumptions on rank, stabilizability, and detectability. Let the system in Fig. 1 be described by the equations:

$$\dot{x}(t) = Ax(t) + B_1w(t) + B_2u(t) \quad (3)$$

$$z(t) = C_1x(t) + D_{12}u(t) \quad (4)$$

$$y(t) = C_2x(t) + D_{21}w(t) \quad (5)$$

i.e., in Fig. 1

$$P = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{array} \right]$$

where we also assume, with little loss of generality, that $D_{12}^*D_{12} = I$, $D_{21}D_{21}^* = I$, $D_{12}^*C_1 = 0$ and $B_1D_{21}^* = 0$. Since we wish to have $\|T_{z \leftarrow w}\|_\infty < \gamma$, we need to find u such that

$$\|z\|_2^2 - \gamma^2\|w\|_2^2 < 0 \text{ for all } w \neq 0 \in \mathcal{L}_2(0, \infty).$$

We could consider w to be an adversary trying to make this expression positive, while u has to ensure that it always remains negative in spite of the malicious intentions of w , as in a noncooperative game. Suppose that there exists a solution, X_∞ , to the Algebraic Riccati Equation (ARE),

$$A^*X_\infty + X_\infty A + C_1^*C_1 + X_\infty(\gamma^{-2}B_1B_1^* - B_2B_2^*)X_\infty = 0 \quad (6)$$

with $X_\infty \geq 0$ and $A + (\gamma^{-2}B_1B_1^* - B_2B_2^*)X_\infty$ a stable “A-matrix.” A simple substitution then gives that

$$\begin{aligned} \frac{d}{dt}(x(t)^*X_\infty x(t)) &= -z^*z + \gamma^2w^*w \\ &\quad + v^*v - \gamma^2r^*r \end{aligned}$$

where

$$v := u + B_2^*X_\infty x, \quad r := w - \gamma^{-2}B_1^*X_\infty x.$$

Now let $x(0) = 0$ and assuming stability so that $x(\infty) = 0$, then integrating from 0 to ∞ gives

$$\|z\|_2^2 - \gamma^2\|w\|_2^2 = \|v\|_2^2 - \gamma^2\|r\|_2^2 \quad (7)$$

If the state is available to u , then the control law $u = -B_2^*X_\infty x$ gives $v = 0$ and $\|z\|_2^2 - \gamma^2\|w\|_2^2 < 0$ for all $w \neq 0$. It can be shown that (6) has a solution if there exists a controller such that $\|F_l(P, K)\|_\infty < \gamma$. In addition since transposing a system does not change its $\mathcal{H}_\infty$ -norm, the following dual ARE will also have a solution, $Y_\infty \geq 0$,

$$\begin{aligned} AY_\infty + Y_\infty A^* + B_1B_1^* \\ + Y_\infty(\gamma^{-2}C_1^*C_1 - C_2^*C_2)Y_\infty = 0 \end{aligned} \quad (8)$$

To obtain a solution to the output feedback case, note that (7) implies that $\|z\|_2^2 < \gamma^2\|w\|_2^2$ if and only if $\|v\|_2^2 < \gamma^2\|r\|_2^2$ and $\bar{v} = F_l(P_{\text{tmp}}, K)\bar{r}$ where

$$\begin{bmatrix} \bar{v} \\ \bar{y} \end{bmatrix} = P_{\text{tmp}} \begin{bmatrix} \bar{r} \\ \bar{u} \end{bmatrix},$$

and

$$P_{\text{tmp}} = \left[\begin{array}{c|cc} A + \gamma^{-2}B_1B_1^*X_\infty & B_1 & B_2 \\ \hline B_2^*X_\infty & 0 & I \\ C_2 & D_{21} & 0 \end{array} \right]$$

The special structure of this problem enables a solution to be derived in much the same way as the dual of the state feedback problem. The corresponding ARE will have a solution $Y_{\text{tmp}} = (I - \gamma^{-2}Y_\infty X_\infty)^{-1}Y_\infty \geq 0$ if and only if the spectral radius $\rho(Y_\infty X_\infty) < \gamma^2$.

The above outline, supported by significant technical detail and assumptions, will therefore demonstrate that there exists a stabilizing controller, $K(s)$, such that the system described by (3–5) satisfies $\|T_{z \leftarrow w}\|_\infty < \gamma$ if and only if there exist stabilizing solutions to the AREs in (6) and (8) such that

$$X_\infty \geq 0, \quad Y_\infty \geq 0, \quad \rho(Y_\infty X_\infty) < \gamma^2 \quad (9)$$

The state equations for the resulting controller can be written as

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + B_1\hat{w}_{\text{worst}} + B_2u + Z_\infty L_\infty (C_2\hat{x} - y) \\ u &= F_\infty\hat{x}, \quad \hat{w}_{\text{worst}} = \gamma^{-2}B_1^*X_\infty\hat{x} \\ F_\infty &:= -B_2^*X_\infty, \quad L_\infty := -Y_\infty C_2^*, \\ Z_\infty &:= (I - \gamma^{-2}Y_\infty X_\infty)^{-1}\end{aligned}$$

giving feedback from a state estimator in the presence of an estimate of the worst-case disturbance.

As $\gamma \rightarrow \infty$ the standard LQG controller is obtained with state feedback of a state estimate obtained from a Kalman filter. In contrast to the LQG problem, the controller depends on the value of γ , and if this is chosen to be too small, then one of the conditions in (9) will be violated. In order to determine the minimum achievable value of γ , a bisection search over γ can be performed checking (9) for each candidate value of γ .

In the limit as $\gamma \rightarrow \gamma_{\text{opt}}$ (its minimum value), a variety of situations can arise and the formulae given here may become ill-conditioned. Typically achieving γ_{opt} is more of an interesting and sometimes challenging mathematical exercise rather than a control system requirement.

This control problem does not have a unique solution, and all solutions can be characterized by an LFT form such as $K = \mathcal{F}_l(M, Q)$ where $Q \in \mathcal{H}_\infty$ with $\|Q\|_\infty < 1$, the present solution is sometimes referred to as the ‘‘central solution’’ obtained with $Q = 0$.

Relations for Other Solution Methods and Problem Formulations

The $\mathcal{H}_\infty$ -control problem has been shown to be related to an extraordinarily wide variety of mathematical techniques and to other problem areas, and investigations of these connections have been most fruitful. Earlier approaches (see Francis 1988) firstly used the characterization of all stabilizing controllers of Youla et al. (see Vidyasagar 1985) which shows that all stable closed-loop systems can be written as

$$\mathcal{F}_l(P, K) = T_1 + T_2 Q T_3, \quad \text{where } Q \in \mathcal{H}_\infty$$

and then solved the model matching problem $\inf_{Q \in \mathcal{H}_\infty} \|T_1 + T_2 Q T_3\|_\infty$. This model matching problem is related to interpolation theory and resulted in a productive interaction with the operator theory. One solution method reduces this problem to J-spectral factorisation problems (where $J = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$) and generates state-space solutions to these problems (Kimura 1997).

The derivation above clearly demonstrates relations to noncooperative differential games, and this is fully developed in Başar and Bernhard (1995) and Green and Limebeer (1995).

The model matching problem is clearly a convex optimization problem. The solution of linear matrix inequalities can give effective methods for solving certain convex optimization problems (e.g., calculating the $\mathcal{H}_\infty$ norm using the bounded real lemma) and can be exploited in the $\mathcal{H}_\infty$ -control problem. See Boyd and Barratt (1991) for a variety of results on convex optimization and control and Dullerud and Paganini (2000) for this approach in robust control.

As noted above there is a family of solutions to the $\mathcal{H}_\infty$ -control problem. The central solution in fact minimizes the entropy integral given by

$$\begin{aligned}I(T_{z \leftarrow w}; \gamma) &:= -\frac{\gamma^2}{2\pi} \int_{-\infty}^{\infty} \ln \\ &\quad \left| \det(I - \gamma^{-2} T_{z \leftarrow w}(j\omega)^* T_{z \leftarrow w}(j\omega)) \right| d\omega\end{aligned}\quad (10)$$

It can be seen that this criterion will penalize the singular values of $T_{z \leftarrow w}(j\omega)$ from being close to γ for a large range of frequencies.

One of the more surprising connections is with the risk-sensitive stochastic control problem (Whittle 1990) where w is assumed to be Gaussian white noise and it is desired to minimize

$$J_T(\gamma) := \frac{\gamma^2}{T} \ln \mathbf{E} \left\{ e^{\frac{1}{2}\gamma^{-2} V_T} \right\} \quad (11)$$

$$\text{where } V_T := \int_{-T}^T z(t)^* z(t) dt \quad (12)$$

The situation with $\gamma^2 > 0$ corresponds to the risk averse controller since large values of V_T

are heavily penalized by the exponential function. It can be shown that if $\|T_{z \leftarrow w}\|_\infty < \gamma$, then

$$\lim_{T \rightarrow \infty} J_T(\gamma) = I(T_{z \leftarrow w}; \gamma)$$

and hence the central controller minimizes both the entropy integral and the risk-sensitive cost function. When γ is chosen to be too small, Whittle refers to the controller having a “neurotic breakdown” because the cost will be infinite for all possible control laws! If in (11) we set $\gamma^2 = -\theta^{-1}$, then the entropy minimizing controller will have $\theta < 0$ and will be risk-averse. The risk neutral controller is when $\theta \rightarrow 0$, $\gamma \rightarrow \infty$ and gives the standard LQG case. If $\theta > 0$, then the controller will be risk-seeking, believing that large variance will be in its favor.

Controller Design with $\mathcal{H}_\infty$ Optimization

The above solutions to the $\mathcal{H}_\infty$ mathematical problem do not give guidance on how to set up a problem to give a “good” control system design. The problem formulation typically involves identifying frequency-dependent weighting matrices to characterize the disturbances, w , and the relative importance of the errors, z (see Skogestad and Postlethwaite 1996). The choice of weights should also incorporate system uncertainty to obtain a robust controller.

One approach that combines both closed-loop system gain and system uncertainty is called $\mathcal{H}_\infty$ *loop-shaping* where the desired closed-loop behavior is determined by the design of the loop-shape using pre- and post-compensators and the system uncertainty is represented in the gap metric (see Vinnicombe 2001). This makes classical criteria such as low frequency tracking error, bandwidth, and high-frequency roll-off all easily incorporated. In this framework the performance and robustness measures are very well matched to each other. Such an approach has been successfully exploited in a number of practical examples (e.g., Hyde (1995) for flight control taken through to successful flight tests). Standard

control design software packages now routinely have $\mathcal{H}_\infty$ -control design modules.

Summary and Future Directions

We have outlined the derivation of $\mathcal{H}_\infty$ controllers with straightforward assumptions that nevertheless exhibit most of the features of linear time-invariant systems without such assumptions and for which routine design software is now available. Connections to a surprisingly large range of other problems are also discussed.

Generalizations to more general cases such as time-varying and nonlinear systems, where the norm is interpreted as the induced norm of the system in $\mathcal{L}_2$, can be derived although the computational aspects are no longer routine. For the problems of robust control, there are necessarily continuing efforts to match the mathematical representation of system uncertainty and system performance to the physical system requirements and to have such representations amenable to analysis and computation.

Cross-References

- ▶ [Fundamental Limitation of Feedback Control](#)
- ▶ [H₂ Optimal Control](#)
- ▶ [Linear Quadratic Optimal Control](#)
- ▶ [LMI Approach to Robust Control](#)
- ▶ [Robust \$\mathcal{H}_2\$ Performance in Feedback Control](#)
- ▶ [Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems](#)

Bibliography

- Dullerud GE, Paganini F (2000) A course in robust control theory: a convex approach. Springer, New York
- Green M, Limebeer D (1995) Linear robust control. Prentice Hall, Englewood Cliffs
- Başar T, Bernhard P (1995) H^∞ -optimal control and related minimax design problems, 2nd edn. Birkhäuser, Boston
- Boyd SP, Barratt CH (1991) Linear controller design: limits of performance. Prentice Hall, Englewood Cliffs

- Doyle JC, Glover K, Khargonekar PP, Francis BA (1989) State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems. *IEEE Trans Autom Control* 34(8):831–847
- Francis BA (1988) A course in $\mathcal{H}_\infty$ control theory. Lecture notes in control and information sciences, vol 88. Springer, Berlin, Heidelberg
- Hyde RA (1995) $\mathcal{H}_\infty$ aerospace control design: a VSTOL flight application. Springer, London
- Kimura H (1997) Chain-scattering approach to $\mathcal{H}_\infty$ -control. Birkhäuser, Basel
- Skogestad S, Postlethwaite I (1996) Multivariable feedback control: analysis and design. Wiley, Chichester
- Vidyasagar M (1985) Control system synthesis: a factorization approach. MIT, Cambridge
- Vinnicombe G (2001) Uncertainty and feedback: $\mathcal{H}_\infty$ loop-shaping and the ν -gap metric. Imperial College Press, London
- Whittle P (1990) Risk-sensitive optimal control. Wiley, Chichester
- Zames G (1981) Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Trans Automat Control* 26:301–320
- Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River, New Jersey

History of Adaptive Control

Karl Åström

Department of Automatic Control, Lund University, Lund, Sweden

Abstract

This entry gives an overview of the development of adaptive control, starting with the early efforts in flight and process control. Two popular schemes, the model reference adaptive controller and the self-tuning regulator, are described with a thumbnail overview of theory and applications. There is currently a resurgence in adaptive flight control as well as in other applications. Some reflections on future development are also given.

Keywords

Adaptive control · Auto-tuning · Flight control · History · Model reference adaptive control · Process control · Robustness · Self-tuning regulators · Stability

Introduction

In everyday language, *to adapt* means to change a behavior to conform to new circumstances, for example, when the pupil area changes to accommodate variations in ambient light. The distinction between adaptation and conventional feedback is subtle because feedback also attempts to reduce the effects of disturbances and plant uncertainty. Typical examples are *adaptive optics* and *adaptive machine tool control* which are conventional feedback systems, with controllers having constant parameters. In this entry we take the pragmatic attitude that an adaptive controller is a controller that can modify its *behavior* in response to changes in the dynamics of the process and the character of the disturbances, by adjusting the controller parameters.

Adaptive control has had a colorful history with many ups and downs and intense debates in the research community. It emerged in the 1950s stimulated by attempts to design autopilots for supersonic aircrafts. Autopilots based on constant-gain, linear feedback worked well in one operating condition but not over the whole flight envelope. In process control there was also a need for automatic tuning of simple controllers.

Much research in the 1950s and early 1960s contributed to conceptual understanding of adaptive control. Bellman showed that *dynamic programming* could capture many aspects of adaptation (Bellman 1961). Feldbaum introduced the notion of *dual control*, meaning that control should be probing as well as directing; the controller should thus inject test signals to obtain better information. Tsypkin showed that schemes for *learning and adaptation* could be captured in a common framework (Tsypkin 1971).

Gabor's work on adaptive filtering (Gabor et al. 1959) inspired Widrow to develop an analogue neural network (Adaline) for adaptive control (Widrow 1962). Widrow's adaptation mechanism was inspired by Hebbian learning in biological systems (Hebb 1949).

There are adaptive control problems in economics and operations research. In these fields the problems are often called *decision making*

under uncertainty. A simple idea, called the *certainty equivalence principle* proposed by Simon (1956), is to neglect uncertainty and treat estimates as if they are true. Certainty equivalence was commonly used in early work on adaptive control.

A period of intense research and ample funding ended dramatically in 1967 with a crash of the rocket powered X15-3 using Honeywell's MH-96 self-oscillating adaptive controller. The self-oscillating adaptive control system has, however, been successfully used in several missiles.

Research in adaptive control resurged in the 1970s, when the two schemes the model reference adaptive control (MRAC) and the self-tuning regulator (STR) emerged together with successful applications. The research was influenced by stability theory and advances in the field of system identification. There was an intensive period of research from the late 1970s through the 1990s. The insight and understanding of stability, convergence, and robustness increased. Recently there has been renewed interest because of flight control (Hovakimyan and Cao 2010; Lavretsky and Wise 2013) and other applications; there is, for example a need for adaptation in autonomous systems.

The Brave Era

Supersonic flight posed new challenges for flight control. Eager to obtain results, there was a very short path from idea to flight test with very little theoretical analysis in between. A number of research projects were sponsored by the US air force. Adaptive flight control systems were developed by General Electric, Honeywell, MIT, and other groups. The systems are documented in the Self-Adaptive Flight Control Systems Symposium held at the Wright Air Development Center in 1959 (Gregory 1959) and the book (Mishkin and Braun 1961).

Whitaker of the MIT team proposed the model reference adaptive controller system which is based on the idea of specifying the performance of a servo system by a reference. Honeywell proposed a self-oscillating adaptive system (SOAS)

which attempted to keep a given gain margin by bringing the system to self-oscillation. The system was flight-tested on several aircrafts. It experienced a disaster in a test on the X-15. Combined with the success of gain scheduling based on air data sensors, the interest in adaptive flight control diminished significantly.

There was also interest of adaptation for process control. Foxboro patented an adaptive process controller with a pneumatic adaptation mechanism in 1950 (Foxboro 1950). DuPont had joint studies with IBM aimed at computerized process control. Kalman worked for a short time at the Engineering Research Laboratory at DuPont, where he started work that led to a paper (Kalman 1958), which is the inspiration of the self-tuning regulator. The abstract of this entry has the statement, *This paper examines the problem of building a machine which adjusts itself automatically to control an arbitrary dynamic process*, which clearly captures the dream of early adaptive control.

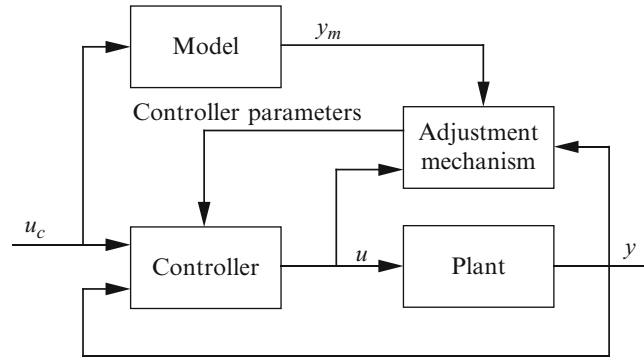
Draper and Li investigated the problem of operating aircraft engines optimally, and they developed a self-optimizing controller that would drive the system towards optimal working conditions. The system was successfully flight-tested (Draper and Li 1966) and initiated the field of *extremal control*.

Many of the ideas that emerged in the brave era inspired future research in adaptive control. The MRAC, the STR, and extremal control are typical examples.

Model Reference Adaptive Control (MRAC)

The MRAC was one idea from the early work on flight control that had a significant impact on adaptive control. A block diagram of a system with model reference adaptive control is shown in Fig. 1. The system has an ordinary feedback loop with a controller, having adjustable parameters, and the process. There is also a reference model which gives the ideal response y_m to the command signal y_m and a mechanism for adjusting the controller parameters θ . The parameter

History of Adaptive Control, Fig. 1 Block diagram of a feedback system with a model reference adaptive controller (MRAC)



adjustment is based on the process output y , the control signal u , and the output y_m of the reference model. Whitaker proposed the following rule for adjusting the parameters:

$$\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta}, \quad (1)$$

where $e = y - y_m$ and $\partial e / \partial \theta$ is the sensitivity derivative. Efficient ways to compute the sensitivity derivative were already available in sensitivity theory. The adaptation law (1) became known as the *MIT rule*.

Experiments and simulations of the model reference adaptive systems indicated that there could be problems with instability, in particular if the adaptation gain γ in Eq. (1) is large. This observation inspired much theoretical research. The goal was to replace the MIT rule by other parameter adjustment rules with guaranteed stability; the models used were non linear continuous time differential equations. The papers Butchart and Shackcloth (1965) and Parks (1966) demonstrated that control laws could be obtained using Lyapunov theory. When all state variables are measured, the adaptation laws obtained were similar to the MIT rule (1), but the sensitivity function was replaced by linear combinations of states and control variables. The problem was more difficult for systems that only permitted output feedback. Lyapunov theory could still be used if the process transfer function was strictly positive real, establishing a connection with Popov's hyper-stability theory (Landau 1979). The assumption of a positive real process is a severe restriction because such

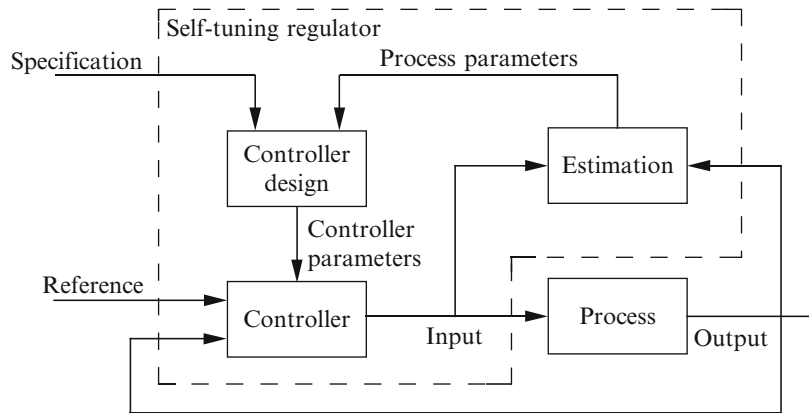
systems can be successfully controlled by high-gain feedback. The difficulty was finally resolved by using a scheme called *error augmentation* (Monopoli 1974; Morse 1980).

There was much research, and by the late 1980s, there was a relatively complete theory for MRAC and a large body of literature (Egardt 1979; Goodwin and Sin 1984; Anderson et al. 1986; Kumar and Varaiya 1986; Åström and Wittenmark 1989; Narendra and Annaswamy 1989; Sastry and Bodson 1989). The problem of flight control was, however, solved by using gain scheduling based on air data sensors and not by adaptive control (Stein 1980). The MRAC was also extended to nonlinear systems using *backstepping* (Krstić et al. 1993); Lyapunov stability and passivity were essential ingredients in developing the algorithm and analyzing its stability.

The Self-Tuning Regulator

The self-tuning regulator was inspired by steady-state regulation in process control. The mathematical setting was discrete time stochastic systems. A block diagram of a system with a self-tuning regulator is shown in Fig. 2. The system has an ordinary feedback loop with a controller and the process. There is an external loop for adjusting the controller parameters based on real-time parameter estimation and control design. There are many ways to estimate the process parameters and many ways to do the control design. Simple schemes do not take parameter uncertainty into account when computing the

History of Adaptive Control, Fig. 2 Block diagram of a feedback system with a self-tuning regulator (STR)



controller parameters invoking the *certainty equivalence principle*.

Single-input, single-output stochastic systems can be modeled by

$$\begin{aligned} y(t) + a_1 y(t-h) + \dots + a_n y(t-nh) = \\ b_1 u(t-h) + \dots + b_n u(t-nh) + \\ c_1 w(t-h) + \dots + c_n w(t-nh) + e(t), \end{aligned} \quad (2)$$

where u is the control signal, y the process output, w a measured disturbance, and e a stochastic disturbance. Furthermore, h is the sampling period and a_k , b_k and c_k , are the parameters. Parameter estimation is typically done using least squares, and a control design that minimized the variance of the variations was well suited for regulation. A surprising result was that if the estimates converge, the limiting controller is a minimum variance controller even if the disturbance e is colored noise (Åström and Wittenmark 1973). Convergence conditions for the self-tuning regulator were given in Goodwin et al. (1980), and a very detailed analysis was presented in Guo and Chen (1991).

The problem of output feedback does not appear for the model (2) because the sequence of past inputs and outputs $y(t-h), \dots, y(t-nh), u(t-h), \dots, u(t-nh)$ is indeed a state, albeit not a minimal state representation. The continuous analogue would be to use derivatives of states and inputs which is not feasible because of measurement noise. The selection of the sampling period is however important.

Early industrial experience indicated that the ability of the STR to adapt feedforward gains was particularly useful, because feedforward control requires good models.

Insight from system identification showed that *excitation* is required to obtain good estimates. In the absence of excitation, a phenomenon of *bursting* could be observed. There could be epochs with small control actions due to insufficient excitation. The estimated parameters then drifted towards values close to or beyond the stability boundary generating large control actions. Good parameter estimates were then obtained and the system quickly recovered stability. The behavior then repeated in an irregular fashion. There are two ways to deal with the problem. One possibility is to detect when there is poor excitation and stop adaptation (Hägglund and Åström 2000). The other is to inject perturbations when there is poor excitation in the spirit of dual control.

Robustness and Unification

The model reference adaptive control and the self-tuning regulator originate from different application domains, flight control and process control. The differences are amplified because they are typically presented in different frameworks, continuous time for MRAC and discrete time for the STR. The schemes are, however, not too different. For a given process model and given design criterion the process model can often be re-parameterized in terms of controller

parameters, and the STR is then equivalent to an MRAC. Similarly there are indirect MRAC where the process parameters are estimated (Egardt 1979).

A fundamental assumption made in the early analyses of model reference adaptive controllers was that the process model used for analysis had the same structure as the real process. Rohrs at MIT, which showed that systems with guaranteed convergence could be very sensitive to unmodeled dynamics, generated a good deal of research to explore robustness to unmodeled dynamics. Averaging theory, which is based on the observation that there are two loops in an adaptive system, a fast ordinary feedback and a slow parameter adjustment loop, turned out to be a key tool for understanding the behavior of adaptive systems. A large body of theory was generated and many books were written (Sastry and Bodson 1989; Ioannou and Sun 1995).

The theory resulted in several improvements of the adaptive algorithms. In the MIT rule (1) and similar adaptation laws derived from Lyapunov theory, the rate of change of the adaptation rate is a multiplication of the error e with other signals in the system. The adaptation rate may then become very large when signals are large. The analysis of robustness showed that there were advantages in avoiding large adaptation rates by *normalizing* the signals. The stability analysis also required that parameter estimates had to be bounded. To achieve this, parameters were *projected* on regions given by prior parameter bounds. The projection did, however, require prior process knowledge. The improved insight obtained from the robustness analysis is well described in the books Goodwin and Sin (1984), Egardt (1979), Åström and Wittenmark (1989), Narendra and Annaswamy (1989), Sastry and Bodson (1989), Anderson et al. (1986), and Ioannou and Sun (1995).

Applications

There were severe practical difficulties in implementing the early adaptive controllers using the analogue technology available in the

brave era. Kalman used a hybrid computer when he attempted to implement his controller. There were dramatic improvements when mini- and microcomputers appeared in the 1970s. Since computers were still slow at the time, it was natural that most experiments were executed in process control or ship steering which are slow processes. Advances in computing eliminated the technological barriers rapidly.

Self-oscillating adaptive controllers are used in several missiles. In piloted aircrafts there were complaints about the perturbation signals that were always exciting the system.

Self-tuning regulators have been used industrially since the early 1970s. Adaptive autopilots for ship steering were developed at the same time. They outperformed conventional autopilots based on PID control, because disturbances generated by waves were estimated and compensated for. These autopilots are still on the market (Northrop Grumman 2005). Asea (now ABB) developed a small distributed control system, Novatune, which had blocks for self-tuning regulators based on least-squares estimation, and minimum variance control. The company First Control, formed by members of the Novatune team, has delivered SCADA systems with adaptive control since 1985. The controllers are used for high-performance process control systems for pulp mills, paper machines, rolling mills, and pilot plants for chemical process control. The adaptive controllers are based on recursive estimation of a transfer function model and a control law based on pole placement. The controller also admits feedforward. The algorithm is provided with extensive safety logic, parameters are projected, and adaptation is interrupted when variations in measured signals and control signals are too small.

The most common industrial uses of adaptive techniques are automatic tuning of PID controllers. The techniques are used both in single loop controllers and in DCS systems. Many different techniques are used, pattern recognition as well as parameter estimation. The relay auto-tuning has proven very useful and has been shown to be very robust because it provides proper excitation of the process automatically. Some of

the systems use automatic tuning to automatically generate gain schedules, and they also have adaptation of feedback and feedforward gains (Åström and Hägglund 2005).

Summary and Future Directions

Adaptive control has had turbulent history with alternating periods of optimism and pessimism. This history is reflected in the conferences. When the IEEE Conference on Decision and Control started in 1962, it included a Symposium on Adaptive Processes, which was discontinued after the 20th CDC in 1981. There were two IFAC symposia on the Theory of Self-Adaptive Control Systems, the first in Rome in 1962 and the second in Teddington in 1965 (Hammond 1966). The symposia were discontinued but reappeared when the Theory Committee of IFAC created a working group on adaptive control chaired by Prof. Landau in 1981. The group brought the communities of control and signal processing together, and a workshop on Adaptation and Learning in Signal Processing and Control (ALCOSP) was created. The first symposium was held in San Francisco in 1983 and the 11th in Caen in 2013.

Adaptive control can give significant benefits, it can deliver good performance over wide operating ranges, and commissioning of controllers can be simplified. Automatic tuning of PID controllers is now widely used in the process industry. Auto-tuning of more general controller is clearly of interest. Regulation performance is often characterized by the Harris index which compares actual performance with minimum variance control. Evaluation can be dispensed with by applying a self-tuning regulator.

There are adaptive controllers that have been in operation for more than 30 years, for example, in ship steering and rolling mills. There is a variety of products that use scheduling, MRAC, and STR in different ways. Automatic tuning is widely used; virtually all new single loop controllers have some form of automatic tuning. Automatic tuning is also used to build gain schedules semiautomatically. The techniques appear

in tuning devices, in single loop controllers, in distributed systems for process control, and in controllers for special applications. There are strong similarities between adaptive filtering and adaptive control. Noise cancellation and adaptive equalization are widely spread uses of adaptation. The signal processing applications are a little easier to analyze because the systems do not have a feedback controller. New adaptive schemes are appearing. The $\mathcal{L}_1$ adaptive controller is one example. It inherits features of both the STR and the MRAC. The *model-free controller* by Fliess and Join (2013) is another example. It is similar to a continuous time version of the self-tuning regulator.

There is renewed interest in adaptive control in the aerospace industry, both for aircrafts and missiles (Lavretsky and Wise 2013). Good results in flight tests have been reported both using MRAC and the recently developed $\mathcal{L}_1$ adaptive controller (Hovakimyan and Cao 2010).

Adaptive control is a rich field, and to understand it well, it is necessary to know a wide range of techniques: nonlinear, stochastic, and sampled data systems, stability, robust control, and system identification.

In the early development of adaptive control, there was a dream of the universal adaptive controller that could be applied to any process with very little prior process knowledge. The insight gained by the robustness analysis shows that knowledge of bounds on the parameters is essential to ensure robustness. With the knowledge available today, adaptive controllers can be designed for particular applications. Design of proper safety nets is an important practical issue. One useful approach is to start with a basic constant-gain controller and provide adaptation as an add-on. This approach also simplifies design of supervision and safety networks.

There are still many unsolved research problems. Methods to determine the achievable adaptation rates are not known. Finding ways to provide proper excitation is another problem. The dual control formulation is very attractive because it automatically generates proper excitation when it is needed. The computations required to solve the Bellman equations are

prohibitive, except in very simple cases. The self-oscillating adaptive system, which has been successfully applied to missiles, does provide excitation. The success of the relay auto-tuner for simple controllers indicates that it may be called in to provide excitation of adaptive controllers. Adaptive control can be an important component of the emerging autonomous system. One may expect that the current upswing in systems biology may provide more inspiration because many biological clearly have adaptive capabilities.

Cross-References

- [Adaptive Control: Overview](#)
- [Autotuning](#)
- [Extremum Seeking Control](#)
- [Model Reference Adaptive Control](#)
- [PID Control](#)

Bibliography

- Anderson BDO, Bitmead RR, Johnson CR, Kokotović PV, Kosut RL, Mareels I, Praly L, Riedle B (1986) Stability of adaptive systems. MIT, Cambridge
- Åström KJ, Hägglund T (2005) Advanced PID control. ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park
- Åström KJ, Wittenmark B (1973) On self-tuning regulators. *Automatica* 9:185–199
- Åström KJ, Wittenmark B (1989) Adaptive control. Addison-Wesley, Reading. Second 1994 edition reprinted by Dover 2006 edition
- Bellman R (1961) Adaptive control processes—a guided tour. Princeton University Press, Princeton
- Butchart RL, Shackcloth B (1965) Synthesis of model reference adaptive control systems by Lyapunov's second method. In: Proceedings 1965 IFAC symposium on adaptive control, Teddington
- Draper CS, Li YT (1966) Principles of optimizing control systems and an application to the internal combustion engine. In: Oldenburger R (ed) Optimal and self-optimizing control. MIT, Cambridge
- Egardt B (1979) Stability of adaptive controllers. Springer, Berlin
- Fliess M, Join C (2013) Model-free control.
- Foxboro (1950) Control system with automatic response adjustment. US Patent 2,517,081
- Gabor D, Wilby WPL, Woodcock R (1959) A universal non-linear filter, predictor and simulator which optimizes itself by a learning process. *Proc Inst Electron Eng* 108(Part B):1061–, 1959
- Goodwin GC, Sin KS (eds) (1984) Adaptive filtering prediction and control. Prentice Hall, Englewood Cliffs
- Goodwin GC, Ramadge PJ, Caines PE (1980) Discrete-time multivariable adaptive control. *IEEE Trans Autom Control* AC-25:449–456
- Gregory PC (ed) (1959) Proceedings of the self adaptive flight control symposium. Wright Air Development Center, Wright-Patterson Air Force Base, Ohio
- Guo L, Chen HF (1991) The Åström-Wittenmark's self-tuning regulator revisited and ELS-based adaptive trackers. *IEEE Trans Autom Control* 30(7):802–812
- Hägglund T, Åström KJ (2000) Supervision of adaptive control algorithms. *Automatica* 36:1171–1180
- Hammond PH (ed) (1966) Theory of self-adaptive control systems. In: Proceedings of the second IFAC symposium on the theory of self-adaptive control systems, 14–17 Sept, National Physical Laboratory, Teddington. Plenum, New York
- Hebb DO (1949) The organization of behavior. Wiley, New York
- Hovakimyan N, Chengyu Cao (2010) $\mathcal{L}_1$ adaptive control theory. SIAM, Philadelphia
- Ioannou PA, Sun J (1995) Stable and robust adaptive control. Prentice-Hall, Englewood Cliffs
- Kalman RE (1958) Design of a self-optimizing control system. *Trans ASME* 80:468–478
- Krstić M, Kanellakopoulos I, Kokotović PV (1993) Non-linear and adaptive control design. Prentice Hall, Englewood Cliffs
- Kumar PR, Varaiya PP (1986) Stochastic systems: estimation, identification and adaptive control. Prentice-Hall, Englewood Cliffs
- Landau ID (1979) Adaptive control—the model reference approach. Marcel Dekker, New York
- Lavretsky E, Wise KA (2013) Robust and adaptive control with aerospace applications. Springer, London
- Mishkin E, Braun L (1961) Adaptive control systems. McGraw-Hill, New York
- Monopoli RV (1974) Model reference adaptive control with an augmented error signal. *IEEE Trans Autom Control* AC-19:474–484
- Morse AS (1980) Global stability of parameter-adaptive control systems. *IEEE Trans Autom Control* AC-25:433–439
- Narendra KS, Annaswamy AM (1989) Stable adaptive systems. Prentice Hall, Englewood Cliffs
- Northrop Grumman (2005) SteerMaster. <http://www.srhmar.com/brochures/as/SPERRY%20SteerMaster%20Control%20System.pdf>
- Parks PC (1966) Lyapunov redesign of model reference adaptive control systems. *IEEE Trans Autom Control* AC-11:362–367
- Sastry S, Bodson M (1989) Adaptive control: stability, convergence and robustness. Prentice-Hall, New Jersey
- Simon HA (1956) Dynamic programming under uncertainty with a quadratic criterion function. *Econometrica* 24:74–81

- Stein G (1980) Adaptive flight control: a pragmatic view. In: Narendra KS, Monopoli RV (eds) Applications of adaptive control. Academic, New York
- Tsytkin YaZ (1971) Adaptation and learning in automatic systems. Academic, New York
- Widrow B (1962) Generalization and information storage in network of Adaline neurons. In: Yovits et al. (ed) Self-organizing systems. Spartan Books, Washington

Human Decision-Making in Multi-agent Systems

Ming Cao

Faculty of Science and Engineering, University of Groningen, Groningen, The Netherlands

Abstract

In order to avoid suboptimal collective behaviors and resolve social dilemmas, researchers have tried to understand how humans make decisions when interacting with other humans or smart machines and carried out theoretical and experimental studies aimed at influencing decision-making dynamics in large populations. We identify the key challenges and open issues in the related research, list a few popular models with the corresponding results, and point out future research directions.

Keywords

Decision making · Multi-agent systems

Introduction

More and more large-scale social, economic, biological, and technological complex systems have been analyzed using models of complex networks of interacting autonomous agents. Researchers are not only interested in using agent-based models to gain insight into how large dynamical networks evolve over time but also keen to introduce control actions into such networks to influence (and/or simulate) the collective behaviors of large populations. This includes

social and economic networks, distributed smart energy grids, intelligent transportation systems, as well as mobile robot and smart sensor networks. However, in practice, such large numbers of interacting autonomous agents making decisions, in particular when humans are involved as participating agents, can result in highly complex, sometimes surprising, and often suboptimal, collective behaviors. It is for this very reason that human decision-making in large multi-agent networks has become a central topic for several research disciplines including economics, sociology, biology, psychology, philosophy, neuroscience, computer science, artificial intelligence, mathematics, robotics, electrical engineering, civil engineering, and last but not least systems and control.

One example to illustrate the complexity is the famous Braess paradox, observed more and more often recently in transportation, communication, and other types of flow networks, in which adding new links can actually worsen the performances of a network and vice versa (Steinberg and Zangwill 1983; Gisches and Rapoport 2012), when each agent decides to optimize its route based on its own local information. Another well-known example is the tragedy of the commons (Ostrom 2008), in which individuals use an excess of a certain shared-resource to maximize their own short-term benefit, leading to the depletion of the resource at a great long-term cost to the group. These considerations impact on a range of emerging engineering applications, e.g., smart transportation and autonomous robots, as well as many of the most pressing current societal concerns, including strategies for energy conservation, improving recycling programs, and reducing carbon emissions.

In order to resolve these types of behavioral and social dilemmas, researchers have tried to gain insight into how humans make decisions when interacting with other autonomous agents, smart machines, or humans and carried out theoretical and experimental studies aimed at influencing decision-making dynamics in large populations in the long run. In fact, control theorists and engineers have been working at the forefront, and a collection of related works

have appeared in the literature. For example, *Proceedings of the IEEE* published in 2012 a special issue on the topic of the decision-making interactions between humans and smart machines (Baillieul et al. 2012). After that, a range of new results have been reported in the literature. In what follows we first identify a few key challenges and difficulties of a number of open issues in the related research, then list a few of the most popular models and the corresponding results, and in the end point out some future research directions.

Challenges and Difficulties

Significantly, when smart machines gain increasing autonomy, empowered by recent breakthroughs in data-driven cognitive learning technologies, they interact with humans in the joint decision-making processes. Humans and smart machines, collectively as networks of autonomous agents, give rise to evolutionary dynamics (Sandholm 2010) that cannot be easily modeled, analyzed, and/or controlled using current systems and control theory that has been effective thus far for engineering practices over the past decades. To highlight the evolutionary nature of the collective decision dynamics, four features can be distinguished:

- (i) Learning and adaption: Autonomous agents may learn and adapt in response to the collaboration or competition pressure from their local peers or an external changing environment.
- (ii) Noise and stochastic effects: Random noise and stochastic deviation are unavoidable in both agents' decisions and the interactions among them.
- (iii) Heterogeneity: Agents differ in their perception capabilities; the agents' interaction patterns co-evolve with the agents' dynamics both in space and time.
- (iv) Availability for control: Some agents may not be available to be controlled directly, and even for those who are, the control is usually in the form of incentives that may only take effect

in the long run or “nudge” the agent in the desired direction.

Therefore, there are still many open problems from the viewpoint of systems and control in developing a general framework for studying human decision-making in multi-agent systems; more generally, there has been an urgent need to develop new theoretical foundations together with computational and experimental tools to tackle the emerging challenging control problems associated with these evolutionary dynamics for networked autonomous agents.

Standard Models and Related Results

Models for human decision-making processes are numerous, and we list a few that are popular and becoming standard especially in the context of multi-agent systems.

Diffusion Model

The standard diffusion model was first proposed by cognitive neural scientists in the 1970s for simple sequential two-choice decision tasks and has been developed into the broader framework known as “evidence accumulation models” (Ratcliff 1978; Ratcliff et al. 2016). The internal process for a human to make a decision is taken to be a process of accumulating evidence from some external stimulus, and once the accumulated evidence is strong enough, a decision is made. More specifically, a one-dimensional drift-diffusion process can be described by the following stochastic differential equation:

$$dz = \alpha dt + \sigma dW, \quad z(0) = 0, \quad (1)$$

where z is the accumulated evidence in favor of a choice of interest, α is the drift rate representing the signal intensity of the stimulus acting on z , and σdW is a Wiener process with the standard deviation σ , called the diffusion rate, representing the effect of white noise. Roughly speaking, when $z(t)$ grows to reach a certain boundary level $\bar{z} > 0$ at time $\bar{t} > 0$, a decision is made at time $\bar{t}$ in favor of the choice that z corresponds

to; otherwise, when another accumulator of some other choice reaches its boundary level first, then a decision favoring that choice is made. Such descriptions match the experimental data recording neural activities when human subjects make sequential decisions in lab environments (Ratcliff et al. 2016). Neural scientists have also used this model to look into the decision time and thus study the speed-accuracy trade-off in decision-making. So far, the model has been successfully applied to a range of cognitive decision-making tasks and used in clinical research (Ratcliff et al. 2016). Researchers from systems and control have looked into the convergence properties of this model (Cao et al. 2010; Woodruff et al. 2012) and used it to predict group decision-making dynamics (Stewart et al. 2012). New applications to robotic systems have also been reported (Reverdy et al. 2015).

Bayesian Model

The Bayesian model, or more generally Bayesian decision theory, is built upon the concepts of Bayesian statistics (Bernardo 1994); when making decisions, humans constantly update their estimates of the beliefs or preference values of different options using new observed inputs through Bayesian inference. It is particularly suitable for multi-alternative, multi-attribute decision-making where observations on different alternatives are dependent (Broder and Schiffer 2004; Evans et al. 2019). Let $p(A_i|x(0), x(1), \dots, x(t))$ denote the posterior belief that alternative A_i is preferred given observations $x(j)$, $0 \leq j \leq t$ up to time $t \geq 0$. Then a Bayesian decision refers to the action at time t of choosing that alternative A_i among all i that maximizes the posterior probability p just given. Note that in some cases, the Bayesian model and diffusion model are closely related (Bitzer et al. 2014). The Bayesian model has found broad applications in dealing with neural and behavior data (Broder and Schiffer 2004; Evans et al. 2019).

Threshold Model

The threshold model is a classic model in sociology to study collective behavior (Granovetter

1978). It stipulates that an individual in a large population will engage in one of several possible behaviors only after a sufficiently large proportion of her surrounding individuals or the population at large have done so. Let the binary state $z_i(t)$ denote the decision of agent i in a large population at time t , $t = 0, 1, 2, \dots$, and $\mathcal{N}_i(t)$ the set of other individuals whose decisions can be observed by agent i at time t . Then the linear threshold model dictates that

$$z_i(t+1) = \begin{cases} 1 & \text{if } \sum_{j \in \mathcal{N}_i(t)} z_j(t) \geq \Theta_i |\mathcal{N}_i(t)| \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where $0 < \Theta_i < 1$ is called the *threshold* of agent i and $|\cdot|$ returns the size of the corresponding set. The model or its variants have been widely used to study propagation of beliefs and cascading of social norm violations in social networks (Centola et al. 2016; Mas and Opp 2016). Control theorists have looked into the convergence of dynamics of large populations of individuals whose behaviors follow the threshold model (Ramazi et al. 2016; Rossi et al. 2019).

Evolutionary Game Models

Game theory has been linked to decision-making processes from the day of its birth, and there are several classic textbooks covering the topic, e.g., Myerson (1991). As a branch of game theory, evolutionary game theory is relevant to decision-making in large multi-agent networks especially considering the evolution of proportions of populations making certain decisions over time (Sandholm 2010). It is worth mentioning that under certain conditions, the threshold model turns out to be equivalent to some evolutionary game models (Ramazi et al. 2016). Learning strategies, such as imitation and best response, can be naturally built into different evolutionary game models; the same also holds for network structures and stochastic effects. The notions of *evolutionarily stable strategies* and *stochastically stable strategies* play prominent roles in a variety of studies (Sandholm 2010). There have been reviews on the analysis and

control of evolutionary game dynamics, and we refer the interested reader to Riehl et al. (2018).

Summary and Future Directions

Human decision-making in multi-agent systems remains a fascinating and challenging topic even after various models have been proposed, tested, and compared both theoretically and experimentally in the past few decades. There are several research directions that keep gaining momentum across different disciplines:

- (i) Using dynamical system models to help match human decision-making behavioral data with neural activities in the brain and provides a physiological foundation for decision theories: Although we have listed the diffusion model and Bayesian model and mentioned a few related works, the gap between behavioral and neural findings is still significant, and there is still no widely accepted unified framework that can accommodate both.
- (ii) Better embedding of human decision theory into the fast developing field of network science so that the network effects in terms of information flow and adaptive learning can be better utilized to understand how a human makes decisions within a group of autonomous agents: Threshold models and evolutionary game models are developments in this direction.
- (iii) Influencing and even controlling the decision-making processes: This is still a new area that requires many more novel ideas and accompanying theoretical and empirical analysis. Behavioral and social dilemmas are common in practice, and difficult to prevent for the betterment of society. Systems and control theory can play a major rule in looking for innovative decision policies and intervention rules (Riehl et al. 2018) to steer a network of autonomous agents away from suboptimal collective behaviors and toward behaviors desirable for society.

Cross-References

- ▶ [Controlling Collective Behavior in Complex Systems](#)
- ▶ [Cyber-physical-Human Systems](#)
- ▶ [Dynamical Social Networks](#)
- ▶ [Evolutionary Games](#)
- ▶ [Learning in Games](#)
- ▶ [Stochastic Games and Learning](#)

Bibliography

- Baillieul J, Leonard NE, Morgansen KA (2012) Interaction dynamics: the interface of humans and smart machines. *Proc IEEE* 100:567–570
- Bernardo JM (1994) Bayesian theory. Wiley, Chichester
- Bitzer S, Park H, Blankenburg F, Kiebel SJ (2014) Perceptual decision making: drift-diffusion model is equivalent to a Bayesian model. *Front Hum Neurosci* (102) 8:1–17
- Broder A, Schiffer S (2004) Bayesian strategy assessment in multi-attribute decision-making. *J Behav Decis Mak* 16:193–213
- Cao M, Stewart A, Leonard NE (2010) Convergence in human decision-making dynamics. *Syst Control Lett* 59:87–97
- Centola D, Willer R, Macy M (2016) The emperor's dilemma: a computational model of self-enforcing norms. *Am J Sociol* 110(4):1009–1040
- Evans NJ, Holmes WR, Trueblood JS (2019) Response-time data provide critical constraints on dynamic models of multi-alternative, multi-attribute choice. *Psychon Bull Rev* 26:193–213
- Gisches EJ, Rapoport A (2012) Degrading network capacity may improve performance: private versus public monitoring in the Braess paradox. *Theor Decis* 73: 901–933
- Granovetter M (1978) Threshold models of collective behavior. *Am J Sociol* 83:1420–1443
- Mas M, Opp K-D (2016) When is ignorance bliss? Disclosing true information and cascades of norm violation in networks. *Soc. Networks* 47: 116–129
- Myerson RB (1991) Game theory: analysis of conflict. Harvard University Press, Cambridge/London
- Ostrom E (2008) Tragedy of the commons. The new palgrave dictionary of economics. Palgrave Macmillan, UK, pp 3573–3576
- Ramazi P, Riehl J, Cao M (2016) Networks of conforming or nonconforming individuals tend to reach satisfactory decisions. *Proc Natl Acad Sci* 113(46):12985–12990
- Ratcliff R (1978) A theory of memory retrieval. *Psychol Rev* 85:59–108

- Ratcliff R, Smith PL, Brown SD, McKoon G (2016) Diffusion decision model: current issues and history. *Trends Cogn Sci* 20(4):260–281
- Reverdy P, Lihan BD, Koditschek DE (2015) A drift-diffusion model for robotic obstacle avoidance. In: *Proceedings of the 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, number 15666827. IEEE
- Riehl J, Ramazi P, Cao M (2018) A survey on the analysis and control of evolutionary matrix games. *Ann Rev Control* 45:87–106
- Rossi W, Como G, Fagnai F (2019) Threshold models of cascades in large-scale networks. *IEEE Trans Netw Sci Eng* 6:158–172
- Sandholm WH (2010) *Population games and evolutionary dynamics*. MIT Press, Cambridge
- Steinberg R, Zangwill WI (1983) The prevalence of Braess' paradox. *Transp Sci* 17:301–318
- Stewart A, Cao M, Nedic A, Tomlin D, Leonard AE (2012) Towards human-robot teams: model-based analysis of human decision making in two-alternative choice tasks with social feedback. *Proc IEEE* 100(3):751–775
- Woodruff C, Vu L, Morgansen KA, Tomlin D (2012) Deterministic modeling and evaluation of decision-making dynamics in sequential two-alternative forced choice tasks. *Proc IEEE* 100(3):734–750

Human-Building Interaction (HBI)

Burcin Becerik-Gerber
Sonny Astani Department of Civil and
Environmental Engineering, Viterbi School of
Engineering, University of Southern California,
Los Angeles, CA, USA

Abstract

This entry defines and provides an overview of an emerging field, namely, human-building interaction, and identifies some of the key multidisciplinary research directions for this new field. It also provides examples of human-building interaction applications for providing personalized thermal comfort-driven systems, applications for allowing for adaptive light switch and blind control, and a framework for enabling social interactions between buildings and their occupants.

Keywords

Building automation · Intelligent built environments · Energy and comfort management · Trust in automation · Human in the loop control

Definition of Human-Building Interaction

Human-building interaction (HBI), an *emerging field*, represents the dynamic interplay of embodied human and building intelligence; it studies how humans interact, perceive, and navigate through the physical, spatial, and social aspects of the built environment and the artifacts (objects) within it. The field of HBI not only includes buildings but also other built environments people occupy and use, such as public spaces, trains, etc. A new field at the intersection of building science, human computer interaction (HCI), social science, and computer science, HBI entails reciprocal actions/influences of humans and buildings on each other and as a result the change of both building and user behavior. For example, user interactions with building elements, such as windows, fans, heaters, or blinds, would impact both the comfort of users (e.g., changes in lighting levels) and the building performance (e.g., increased solar gain causing cooling system to activate) (Langevin et al. 2016).

Built environments are a collection of artifacts (e.g., spaces, sensors, appliances, lighting, computers, windows, etc.); however, they also entail spatiotemporal experiences. Therefore, one important differentiator of HBI from HCI is the fact that the users are immersed in the physical environment they are interacting with, causing these interactions to have immediate user comfort, user satisfaction, and user acceptability implications. These interactions are also influenced by human behavior (D'Oca et al. 2018; Langevin et al. 2015) (user habits, preferences, requirements) as well as building behavior (usability, aesthetical or architectural quality, indoor environmental quality, and performance, such as safety, sustainability, and maintainability).

HBI Research Directions

Emergence of this new research area requires the collaboration of professionals from multiple disciplines, such as computer scientists, interaction designers, architects, civil engineers, mechanical engineers, and social scientists, to name a few. A summary of essential research directions for the future of the HBI field is provided here.

Intelligent Augmentation. With the unparalleled surge of technological advancements, our spaces (e.g., workplace) will change dramatically in the near future. Considering the variety in building spaces (e.g., offices, schools, post offices, warehouses, homes) and the amount of time we spend in these environments, a natural question is how will our spaces change as a result of intelligent environments? As part of HBI, our built environments should take into account both human-centered and building-centered needs and be “aware” of fatigue, stress levels, psychological states, emotions, and moods of users. Through different modes of HBIs, users can offload some of the tasks to the building, reducing cognitive workload of users and also empowering human users and augmenting their performance where and when needed.

Shared Autonomy. Symbiotic relationships between users and buildings will give birth to new forms of artificial intelligence (AI), which will give buildings an unprecedented autonomy and agency. This will require new kinds of caring relationships and trust between buildings and their users. There are several research questions, for example, how does trust change based on the type of tasks users perform in buildings; how does level of intelligence effect trust between buildings and their users; how should AI be designed for buildings that is easily understandable by building users who might not have any background in AI or similar fields? Moreover, when buildings become intelligent agents that can work for and with human users, the desired level of autonomy (e.g., who does what task and when) should be properly defined and also learned on a continuous basis (Ahmadi-Karvigh et al. 2017).

Mutual Adaptation. Buildings should be capable of continuously sensing the changes in the

environment, different states and time, enabling context-aware sensing of human intent. Based on this acquired knowledge, both the building and the user could adapt to achieve shared goals, for example, energy efficiency by allowing the building to change set points (Ghahramani et al. 2014).

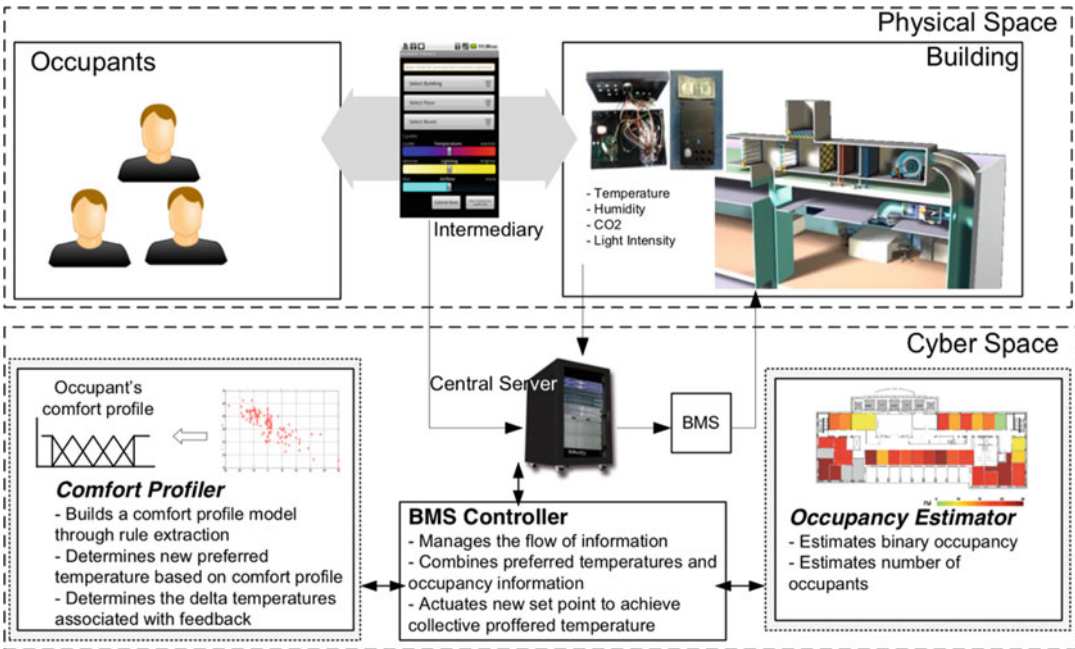
Security and Privacy. With the amount of data collected and stored both about the building and the user and the possibilities to actuate, buildings become hackable. What policies should govern the development and oversight of intelligent environments? What ethical and security measures must be anticipated and addressed? Who should govern the new codes and standards?

New Interfaces. Users and buildings create considerable amounts of data; how to mine and use this information for remembering and reusing of events and activities within a space becomes an interesting question. Moreover, there will exist a need for buildings and users to communicate. How to better design buildings to provide novel interaction opportunities to bi-directionally communicate with their users? What interaction types influence human perception of space and how? How does the style of interaction impact how we perceive our environments? How to enable a building’s lifelong interaction with humans? How to adapt interfaces/modalities based on tasks and users?

Education. We need to rethink our curriculum to be able to educate future engineers who can address the complexities of HBIs and broaden our engineering objectives to include human objectives, incorporating the complexity and dynamism of built environments rather than treating them as unmanageable. In addition, we need to incorporate more computational thinking in our traditional engineering curricula. At the same time, we need to educate building users and building managers to ensure sustainable technology adoption.

HBI Applications and Examples

HBI Framework for Personalized Thermal Comfort-Driven Systems. Jazizadeh et al. (2014a) developed an HBI framework for



Human-Building Interaction (HBI), Fig. 1 Human-building interaction framework proposed in Jazizadeh et al. (2014a)

integrating building user's thermal comfort preferences into the HVAC control loop. To do so, they have developed an HBI interface, using participatory sensing via smartphones, and collected occupants' comfort perception indices (i.e., comfort votes provided by users and mapped to a numerical value) and ambient temperature data through a sensor network. They computed comfort profiles for each user using a fuzzy rule-based descriptive and predictive model. Implementation of the personalized comfort profiles in the HVAC system was tested through actuation via a proportional controller algorithm in a building. The study proved that the algorithm was capable of keeping the zone temperatures within the range of user's thermal comfort preferences. In a follow-up study (Jazizadeh et al. 2014b), the researchers further tested the framework and demonstrated comfort improvements along with a 39% reduction in daily average airflow when the occupant comfort preferences were used (Fig. 1).

HBI Framework for Adaptive Light Switch and Blind Control. Gunay et al. (2017) analyzed the light switch and blind use behavior of occupants in private offices and developed an

adaptive control algorithm for lighting and blinds. Using the observed behavioral data (use of light switches and blinds), the algorithm learns the occupants' illuminance preferences, uses this information to determine photosensor setpoints, and switches off lighting and opens the blinds. The researchers tested the controller in private offices and shared office spaces and found that the solution could substantially reduce the lighting loads (around 25%) in office buildings without compromising the occupant's visual comfort. In another study, also focusing on personalized daylighting control for optimizing visual satisfaction and lighting energy use (Xiong et al. 2019), the researchers developed a personalized shading control framework. This optimization framework used personalized visual satisfaction utility functions and model-predicted lighting energy. Instead of assigning arbitrary weights to objectives, the researchers allowed users to make the final decisions in real time using a slider that balanced personal visual satisfaction and lighting energy use.

HBI Framework for Enabling Social Interactions Between Buildings and Their Users. Khashe et al. (2017) investigated the influence of direct



Human-Building Interaction (HBI), Fig. 2 Female and male avatar representing the building or building manager and communicate with the building occupants (Khashe et al. 2017)

communication between buildings and their users and tested the effectiveness of communicator styles (i.e., avatar, voice, and text) and communicator gender (i.e., female and male) and personification (communicator being either a building or building manager) on building occupant's compliance with pro-environmental requests. While the users were completing an office task (reading a passage), a request was presented to the user delivered through a combination of style, gender, and persona. An example of the request was *"If I open the blinds for you to have natural light, would you please dim or turn off the artificial lights?"* They found that the avatar was more effective than the voice and the voice was more effective than the text requests. Users complied more when the communicator was a female agent and when they thought the communicator was the building manager. In a follow-up study (Khashe et al. 2018), the researchers tested the effectiveness of social dialogue instead of a single request. They again tested two personas, building manager and building, however, this time only with female avatar communicator. The avatar started with a small talk, introduced herself (either as the building or building manager), told the user her name and asked the user's name, and so on, and then followed up with the request. They found that social dialogue persuaded the users to perform more pro-environmental actions. However, differently than the findings in their previous study, when the agent got engaged in a social

dialogue with the users, they complied more with the requests if they thought the building was communicating with them. This two-tier study is one of the examples of personification of buildings through digital artifacts with the aim of building a relationship between an inanimate object (i.e., a building) and its users for achieving a shared goal (i.e., pro-environmentalism) (Fig. 2).

Cross-References

- [Building Comfort and Environmental Control](#)
- [Building Control Systems](#)
- [Building Energy Management System](#)
- [Cyber-Physical-Human Systems](#)

Bibliography

- Ahmadi-Karvigh S, Becerik-Gerber B, Soibelman L (2017) One size does not fit all: understanding user preferences for building automation systems. *J Energ Buildings* 145:163–173
- D'Oca S, Pisello AL, De Simone M, Bartherlmes V, Hong T, Corinati S (2018) Human-building interaction at work: findings from an interdisciplinary cross-country survey in Italy. *J Building Environ* 132: 147–159
- Ghahramani A, Jazizadeh F, Becerik-Gerber B (2014) A knowledge based approach for selecting energy-aware and comfort-driven HVAC temperature set points. *J Energ Buildings* 85:536–548
- Gunay B, O'Brien W, Beausoleil-Morrison I, Gilani S (2017) Development and implementation of an adaptive lighting and blinds control algorithm. *J Building Environ* 113:185–199

- Jazizadeh F, Ghahramani A, Becerik-Gerber B, Kichkaylo T, Orosz M (2014a) Human-building interaction framework for personalized thermal comfort driven systems in office buildings. *ASCE J Comput Civil Eng* 28(1):2–16. Special issue: Computational approaches to understand and reduce energy consumption in the built environment
- Jazizadeh F, Ghahramani A, Becerik-Gerber B, Kichkaylo T, Orosz M (2014b) User-led decentralized thermal comfort driven HVAC operations for improved efficiency in office buildings. *J Energ Buildings* 70:398–410
- Khashe S, Lucas G, Becerik-Gerber B, Gratch J (2017) Buildings with persona: towards effective building-occupant communication. *J Comput Hum Behav* 75:607–618
- Khashe S, Lucas G, Becerik-Gerber B, Gratch J (2018) Establishing social dialogue between buildings and their users. *Int J Comput Hum Interact*. <https://doi.org/10.1080/10447318.2018.1555346>
- Langevin J, Gurian P, Wen J (2015) Tracking the human-building interaction: a longitudinal field study of occupant behavior in air-conditioned offices. *J Environ Psychol* 42:94–115
- Langevin J, Gurian P, Wen J (2016) Quantifying the human–building interaction: considering the active, adaptive occupant in building performance simulation. *J Energ Buildings* 117:372–386
- Xiong J, Tzempelikos A, Bilionis I, Karava P (2019) A personalized daylighting control approach to dynamically optimize visual satisfaction and lighting energy use. *J Energ Buildings* 193:111–126

Hybrid Dynamical Systems, Feedback Control of

Ricardo G. Sanfelice
Department of Electrical and Computer
Engineering, University of California, Santa
Cruz, CA, USA

Abstract

The control of systems with hybrid dynamics requires algorithms capable of dealing with

the intricate combination of continuous and discrete behavior, which typically emerges from the presence of continuous processes, switching devices, and logic for control. Several analysis and design techniques have been proposed for the control of nonlinear continuous-time plants, but little is known about controlling plants that feature truly hybrid behavior. This short entry focuses on recent advances in the design of feedback control algorithms for hybrid dynamical systems. The focus is on hybrid feedback controllers that are systematically designed employing Lyapunov-based methods. The control design techniques summarized in this entry include control Lyapunov function-based control, passivity-based control, trajectory tracking control, safety, and temporal logic.

Keywords

Feedback control · Hybrid control · Hybrid systems · Asymptotic stability

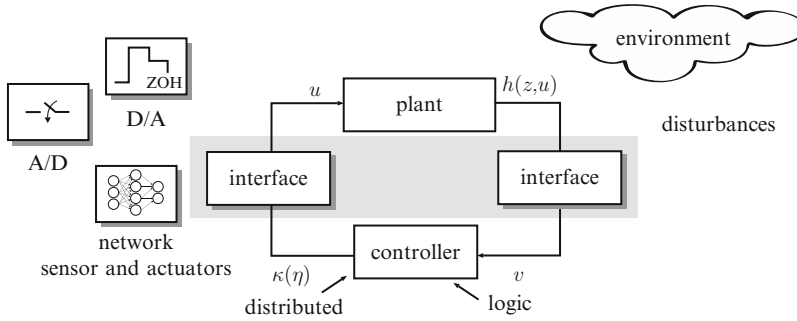
Introduction

A *hybrid control system* is a *feedback system* whose variables may flow and, at times, jump. Such a hybrid behavior can be present in one or more of the subsystems of the feedback system: in the system to control, i.e., *the plant*; in the algorithm used for control, i.e., *the controller*; or in the subsystems needed to interconnect the plant and the controller, i.e., *the interfaces/signal conditioners*. Figure 1 depicts a feedback system in closed-loop configuration with such subsystems under the presence of environmental disturbances. Due to its hybrid dynamics, a hybrid control system is a particular type of *hybrid dynamical system*.

Motivation

Hybrid dynamical systems are ubiquitous in science and engineering as they permit capturing the complex and intertwined continuous/discrete

This research has been partially supported by the National Science Foundation under CAREER Grant no. ECS-1150306, ECS-1710621, and CNS-1544396; by the Air Force Office of Scientific Research under Grant no. FA9550-12-1-0366, FA9550-16-1-0015, FA9550-19-1-0053, and FA9550-19-1-0169; and by CITRIS and the Banatao Institute at the University of California.



Hybrid Dynamical Systems, Feedback Control of,
Fig. 1 A hybrid control system: a feedback system with a plant, controller, and interfaces/signal conditioners

(along with environmental disturbances) as subsystems featuring variables that flow and, at times, jump

behavior of a myriad of systems with variables that flow and jump. The recent popularity of feedback systems combining physical and software components demands tools for stability analysis and control design that can systematically handle such a complex combination. To avoid the issues due to approximating the dynamics of a system, in numerous settings, it is mandatory to keep the system dynamics as pure as possible and to be able to design feedback controllers that can cope with flow and jump behavior in the system.

Modeling Hybrid Control Systems

In this entry, hybrid control systems are represented in the framework of *hybrid equations/inclusions* for the study of hybrid dynamical systems. Within this framework, the continuous dynamics of the system are modeled using a differential equation/inclusion, while the discrete dynamics are captured by a difference equation/inclusion. A solution to such a system can *flow* over nontrivial intervals of time and *jump* at certain time instants. The conditions determining whether a solution to a hybrid system should flow or jump are captured by subsets of the state space and input space of the hybrid control system. In this way, a *plant* with hybrid dynamics can be modeled by the hybrid inclusion (This hybrid inclusion captures the dynamics of (constrained or unconstrained)

continuous-time systems when $D_P = \emptyset$ and G_P is arbitrary. Similarly, it captures the dynamics of (constrained or unconstrained) discrete-time systems when $C_P = \emptyset$ and F_P is arbitrary. Note that while the output inclusion does not explicitly include a constraint on (z, u) , the output map is only evaluated along solutions.)

$$\mathcal{H}_P : \begin{cases} \dot{z} \in F_P(z, u) & (z, u) \in C_P \\ z^+ \in G_P(z, u) & (z, u) \in D_P \\ y = h(z, u) \end{cases} \quad (1)$$

where z is the *state* of the plant and takes values from the Euclidean space $\mathbb{R}^{n_P}$, u is the *input* and takes values from $\mathbb{R}^{m_P}$, y is the *output* and takes values from the output space $\mathbb{R}^{r_P}$, and (C_P, F_P, D_P, G_P, h) is the *data* of the hybrid system. The set C_P is the *flow set*, the set-valued map F_P is the *flow map*, the set D_P is the *jump set*, the set-valued map G_P is the *jump map*, and the single-valued map h is the *output map*. In (1), $\dot{z}$ denotes time derivative and z^+ denotes an instantaneous change in z .

Example 1 (controlled bouncing ball) Consider the juggling system consisting of a ball moving vertically and bouncing on a fixed horizontal surface. The surface, located at the origin of the line of motion, is equipped with a mechanical actuator that controls the speed of the ball resulting after impacts. From a physical viewpoint, control authority may be obtained varying the viscoelastic properties of the surface and, in turn,

the coefficient of restitution of the surface. The position and the velocity of the ball are denoted as z_1 and z_2 , respectively. Between bounces, the free motion of the ball is given by

$$\dot{z}_1 = z_2, \quad \dot{z}_2 = -\gamma \quad (2)$$

where $\gamma > 0$ is the gravity constant. The conditions at which impacts occur are modeled as

$$z_1 = 0 \text{ and } z_2 \leq 0 \quad (3)$$

while the new value of the state variables after each impact is described by the difference equations

$$z_1^+ = z_1, \quad z_2^+ = u \quad (4)$$

where u , which is larger than or equal to zero, is the controlled velocity after impacts, capturing the effect of the mechanism installed on the horizontal surface. In this way, the data (C_P, F_P, D_P, G_P, h) of the bouncing ball model is defined as follows:

$$\begin{aligned} F_P(z, u) &:= \begin{bmatrix} z_2 \\ -\gamma \end{bmatrix} & \forall (z, u) \in C_P &:= \left\{ (z, u) \in \mathbb{R}^2 \times \mathbb{R} : z_1 \geq 0 \right\} \\ G_P(z, u) &:= \begin{bmatrix} z_1 \\ u \end{bmatrix} & \forall (z, u) \in D_P &:= \left\{ (z, u) \in \mathbb{R}^2 \times \mathbb{R} : z_1 = 0, z_2 \leq 0 \right\} \end{aligned}$$

and, when assuming that the state z is measured, the output map is $h(z, u) = z$. $\triangle$

Other examples of hybrid plants whose dynamics can be captured by $\mathcal{H}_P$ in (1) include walking robots, network control systems, and spiking neurons. Systems with different modes of operation can also be modeled as $\mathcal{H}_P$ and, as the following example illustrates, can be captured by the constrained differential equation (or inclusion) part of $\mathcal{H}_P$. Such systems are not necessarily hybrid – at least as the term *hybrid* is used in this short entry – since they can be modeled by a dynamical system with state that only evolves continuously.

Example 2 (Thermostat system) The evolution of the temperature of a room under the effect of a heater can be modeled by a differential equation with constraints on its input. The temperature of the room is denoted by z and takes values from $\mathbb{R}$. The input is given by the pair $u = (u_1, u_2)$, where u_1 denotes whether the heater is turned on ($u_1 = 1$) or turned off ($u_1 = 0$) – these are the constraints on the inputs – while u_2 denotes the temperature outside the room, which can assume any value in $\mathbb{R}$. With these definitions, the evolution of the temperature z is governed by

$$\dot{z} = -z + \begin{bmatrix} z_\Delta & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$(z, u) \in C_P = \left\{ (z, u) \in \mathbb{R} \times \mathbb{R}^2 : u_1 \in \{0, 1\} \right\} \quad (5)$$

where z_Δ is a positive constant representing the heater capacity. Note that C_P captures the constraint on the input u_1 , which restricts it to the values 0 and 1. $\triangle$

Given an input u , a *solution to a hybrid inclusion* is defined by a state trajectory ϕ that satisfies the inclusions. Both the input and the state trajectory are functions of $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N} := [0, \infty) \times \{0, 1, 2, \dots\}$, where t keeps track of the amount of flow while j counting the number of jumps of the solution. These functions are given by *hybrid arcs* and *hybrid inputs*, which are defined on *hybrid time domains*. More precisely, hybrid time domains are subsets E of $\mathbb{R}_{\geq 0} \times \mathbb{N}$ that, for each $(T', J') \in E$,

$$E \cap ([0, T'] \times \{0, 1, \dots, J'\})$$

can be written in the form

$$\bigcup_{j=0}^{J-1} ([t_j, t_{j+1}], j)$$

for some finite sequence of times $0 = t_0 \leq t_1 \leq t_2 \leq \dots \leq t_J$, $J \in \mathbb{N}$. A hybrid arc ϕ is a function on a hybrid time domain. (The set $E \cap ([0, T] \times \{0, 1, \dots, J\})$ defines a compact hybrid time domain since it is bounded and closed.) The hybrid time domain of ϕ is denoted by $\text{dom } \phi$. A hybrid arc is such that, for each $j \in \mathbb{N}$, $t \mapsto \phi(t, j)$ is locally absolutely continuous on intervals of flow $I^j := \{t : (t, j) \in \text{dom } \phi\}$ with nonzero Lebesgue measure. A hybrid input u is a function on a hybrid time domain that, for each $j \in \mathbb{N}$, $t \mapsto u(t, j)$ is Lebesgue measurable and locally essentially bounded on the interval I^j .

In this way, a solution to the plant $\mathcal{H}_P$ is given by a pair (ϕ, u) with $\text{dom } \phi = \text{dom } u$ ($= \text{dom}(\phi, u)$) satisfying

(S0) $(\phi(0, 0), u(0, 0)) \in \bar{C}_P$ or $(\phi(0, 0), u(0, 0)) \in D_P$, and $\text{dom } \phi = \text{dom } u$;

(S1) For each $j \in \mathbb{N}$ such that I^j has nonempty interior $\text{int}(I^j)$, we have

$$(\phi(t, j), u(t, j)) \in C_P \quad \text{for all } t \in \text{int}(I^j)$$

and

$$\frac{d}{dt} \phi(t, j) \in F_P(\phi(t, j), u(t, j))$$

for almost all $t \in I^j$

(S2) For each $(t, j) \in \text{dom}(\phi, u)$ such that $(t, j+1) \in \text{dom}(\phi, u)$, we have

$$(\phi(t, j), u(t, j)) \in D_P$$

and

$$\phi(t, j+1) \in G_P(\phi(t, j), u(t, j))$$

A solution pair (ϕ, u) to $\mathcal{H}$ is said to be *complete* if $\text{dom}(\phi, u)$ is unbounded and *maximal* if there does not exist another pair $(\phi, u)'$ such that (ϕ, u) is a truncation of $(\phi, u)'$ to some proper subset of $\text{dom}(\phi, u)'$. A solution pair (ϕ, u) to $\mathcal{H}$ is said to be *Zeno* if it is complete and the projection of $\text{dom}(\phi, u)$ onto $\mathbb{R}_{\geq 0}$ is bounded.

On decomposition of inputs and outputs: At times, it is convenient to define inputs $u_c \in \mathbb{R}^{m_{P,c}}$ and $u_d \in \mathbb{R}^{m_{P,d}}$ collecting every component of the input u that affect flows and that affect jumps, respectively (Some of the components of u can be used to define both u_c and u_d , that is, there could be inputs that affect both flows and jumps.). Similarly, one can define y_c and y_d as the components of y that are measured during flows and jumps, respectively.

To control the hybrid plant $\mathcal{H}_P$ in (1), control algorithms that can cope with the nonlinearities introduced by the flow and jump equations/inclusions are required. In general, feedback controllers designed using classical techniques from the continuous-time and discrete-time domain fall short. Due to this limitation, hybrid feedback controllers would be more suitable for the control of plants with hybrid dynamics. Then, following the hybrid plant model above, hybrid controllers for the plant $\mathcal{H}_P$ in (1) will be given by the hybrid inclusion:

$$\mathcal{H}_K : \begin{cases} \dot{\eta} \in F_K(\eta, v) & (\eta, v) \in C_K \\ \eta^+ \in G_K(\eta, v) & (\eta, v) \in D_K \\ \zeta = \kappa(\eta, v) \end{cases} \quad (6)$$

where η is the *state* of the controller and takes values from the Euclidean space $\mathbb{R}^{n_K}$, v is the *input* and takes values from $\mathbb{R}^{r_P}$, ζ is the *output* and takes values from the output space $\mathbb{R}^{m_P}$, and $(C_K, F_K, D_K, G_K, \kappa)$ is the *data* of the hybrid inclusion defining the hybrid controller.

The control of $\mathcal{H}_P$ via $\mathcal{H}_K$ defines an interconnection through the input/output assignment $u = \zeta$ and $v = y$; the system in Fig. 1 without interfaces represents this interconnection. The resulting closed-loop system is a hybrid dynamical system given in terms of a hybrid inclusion/equation with state $x = (z, \eta)$. We will denote such a closed-loop system by $\mathcal{H}$, with data denoted (C, F, D, G) , state $x \in \mathbb{R}^n$, and dynamics:

$$\mathcal{H} : \begin{cases} \dot{x} \in F(x) & x \in C \\ x^+ \in G(x) & x \in D \end{cases} \quad (7)$$

Its data can be constructed from the data (C_P, F_P, D_P, G_P, h) and $(C_K, F_K, D_K, G_K, \kappa)$ of each of the subsystems. Solutions to both $\mathcal{H}_K$ and $\mathcal{H}$ are understood following the notion introduced above for $\mathcal{H}_P$.

Definitions and Notions

For convenience, we use the equivalent notation $[x^\top \ y^\top]^\top$ and (x, y) for vectors x and y . Also, we denote by $\mathcal{K}_\infty$ the class of functions from $\mathbb{R}_{\geq 0}$ to $\mathbb{R}_{\geq 0}$ that are continuous, zero at zero, strictly increasing, and unbounded.

In general, the dynamics of hybrid inclusions have right-hand sides given by set-valued maps. Unlike functions or single-valued maps, set-valued maps may return a set when evaluated at a point. For instance, at points in C_P , the set-valued flow map F_P of the hybrid plant $\mathcal{H}_P$ might return more than one value, allowing for different values of the derivative of z . A particular continuity property of set-valued maps that will be needed later is lower semicontinuity. A set-valued map S from $\mathbb{R}^n$ to $\mathbb{R}^m$ is lower semicontinuous if for each $x \in \mathbb{R}^n$ one has that $\liminf_{x_i \rightarrow x} S(x_i) \supset S(x)$, where $\liminf_{x_i \rightarrow x} S(x_i) = \{z : \forall x_i \rightarrow x, \exists z_i \rightarrow z \text{ s.t. } z_i \in S(x_i)\}$ is the so-called inner limit of S .

A vast majority of control problems consist of designing a feedback algorithm that assures that a function of the solutions to the plant approach a desired set-point condition (*attractivity*) and, when close to it, the solutions remain nearby (*stability*). In some scenarios, the desired set-point condition is not necessarily an isolated point, but rather a set. The problem of designing a hybrid controller $\mathcal{H}_K$ for a hybrid plant $\mathcal{H}_P$ typically pertains to the stabilization of sets, in particular, due to the hybrid state of the controller including timers that persistently evolve within a bounded time interval and logic variables that take values from discrete sets. Denoting by $\mathcal{A}$ the set of points to stabilize for the closed-loop system $\mathcal{H}$ and $|\cdot|_{\mathcal{A}}$ as the distance to such set, the following property captures the typically desired properties outlined above. A closed set $\mathcal{A}$ is said to be

- (S) *Stable* if for each $\varepsilon > 0$ there exists $\delta > 0$ such that each maximal solution ϕ to $\mathcal{H}$ with $\phi(0, 0) = x_o$, $|x_o|_{\mathcal{A}} \leq \delta$ satisfies $|\phi(t, j)|_{\mathcal{A}} \leq \varepsilon$ for all $(t, j) \in \text{dom } \phi$;
- (pA) *Pre-attractive* if there exists $\mu > 0$ such that every maximal solution ϕ to $\mathcal{H}$ with $\phi(0, 0) = x_o$, $|x_o|_{\mathcal{A}} \leq \mu$ is bounded and if it is complete satisfies $\lim_{(t, j) \in \text{dom } \phi, t+j \rightarrow \infty} |\phi(t, j)|_{\mathcal{A}} = 0$;
- (FTpA) *Finite time pre-attractive* if there exists $\mu > 0$ such that every maximal solution ϕ to $\mathcal{H}$ with $\phi(0, 0) = x_o$, $|x_o|_{\mathcal{A}} \leq \mu$ is such that $|\phi(t, j)|_{\mathcal{A}} = 0$ for some $(t, j) \in \text{dom } \phi$;
- (pAS) *Pre-asymptotically stable* if it is stable and pre-attractive.

The basin of pre-attraction of a pre-asymptotically stable set $\mathcal{A}$ is the set of points from where the pre-attractivity property holds. The set $\mathcal{A}$ is said to be globally pre-asymptotically stable when the basin of pre-attraction is equal to the entire state space. Similarly, notions pertaining to finite-time stability and basin of attraction for finite-time stability can be defined. When every maximal solution is complete, then the prefix “pre” can be dropped since, in that case, the notions resemble those for continuous-time or discrete-time systems with solutions defined for all time.

At times, one is interested in asserting convergence when the state trajectory or the output remains in a set. For instance, a dynamical system (with assigned inputs) is said to be detectable when its output being held to zero implies that its state converges to the origin (or to a particular set of interest). A similar property can be defined for hybrid dynamical systems, for a general set K , which may not necessarily be the set of points at which the output is zero. For the closed-loop system $\mathcal{H}$, given sets $\mathcal{A}$ and K , the distance to $\mathcal{A}$ is said to be

- (D) *0-input detectable relative to K* if every complete solution ϕ to $\mathcal{H}$ is such that

$$\phi(t, j) \in K \quad \forall (t, j) \in \text{dom } \phi \quad \Rightarrow \quad \lim_{(t, j) \in \text{dom } \phi, t+j \rightarrow \infty} |\phi(t, j)|_{\mathcal{A}} = 0$$

Note that “ $\phi(t, j) \in K$ ” captures the “output being held to zero”-like property in the usual detectability notion.

In addition to stability, attractivity, and detectability, in this short note, we are interested in hybrid controllers that guarantee that a set K is forward invariant in the following sense:

- (FpI) *Forward pre-invariant* if each maximal solution ϕ to $\mathcal{H}$ with $\phi(0, 0) = x_o, x_o \in K$, satisfies $\phi(t, j) \in K$ for all $(t, j) \in \text{dom } \phi$.
- (FI) *Forward invariant* if each maximal solution ϕ to $\mathcal{H}$ with $\phi(0, 0) = x_o, x_o \in K$, is complete and satisfies $\phi(t, j) \in K$ for all $(t, j) \in \text{dom } \phi$.

Feedback Control Design for Hybrid Dynamical Systems

Several methods for the design of a hybrid controller $\mathcal{H}_K$ rendering a given set $\mathcal{A}$ such that the properties defined in section “Definitions and Notions” are given below. At the core of these methods are sufficient conditions in terms of Lyapunov-like functions guaranteeing properties such as asymptotic stability, invariance, and finite-time attractivity of a set. Some of the methods presented below exploit such sufficient conditions when applied to the closed-loop system $\mathcal{H}$, while others exploit the properties of the hybrid plant to design controllers with a particular structure.

CLF-Based Control Design

In simple terms, a control Lyapunov function (CLF) is a regular enough scalar function that decreases along solutions to the system for some values of the unassigned input. When such a function exists, it is very tempting to exploit its properties to construct an asymptotically stabilizing control law. Following the ideas from the literature of continuous-time and discrete-time nonlinear systems, we define control Lyapunov functions for hybrid plants $\mathcal{H}_P$ and present results on CLF-based

control design. For simplicity, as mentioned in the *input and output modeling remark* in section “Definitions and Notions”, we use inputs u_c and u_d instead of u . Also, for simplicity, we restrict the discussion to sets $\mathcal{A}$ that are compact as well as hybrid plants with F_P, G_P single-valued and such that $h(z, u) = z$. For notational convenience, we use Π to denote the “projection” of C_P and D_P onto $\mathbb{R}^{n_P}$, i.e., $\Pi(C_P) = \{z : \exists u_c \text{ s.t. } (z, u_c) \in C_P\}$ and $\Pi(D_P) = \{z : \exists u_d \text{ s.t. } (z, u_d) \in D_P\}$, and the set-valued maps $\Psi_c(z) = \{u_c : (z, u_c) \in C_P\}$ and $\Psi_d(z) = \{u_d : (z, u_d) \in D_P\}$.

Given a compact set $\mathcal{A}$, a continuously differentiable function $V : \mathbb{R}^{n_P} \rightarrow \mathbb{R}$ is a *control Lyapunov function for $\mathcal{H}_P$ with respect to $\mathcal{A}$* if there exist $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ and a continuous, positive definite function ρ such that

$$\alpha_1(|z|_{\mathcal{A}}) \leq V(z) \leq \alpha_2(|z|_{\mathcal{A}})$$

$$\forall z \in \mathbb{R}^{n_P}$$

$$\inf_{u_c \in \Psi_c(z)} \langle \nabla V(z), F_P(z, u_c) \rangle \leq -\rho(|z|_{\mathcal{A}})$$

$$\forall z \in \Pi(C_P) \quad (8)$$

$$\inf_{u_d \in \Psi_d(z)} V(G_P(z, u_d)) - V(z) \leq -\rho(|z|_{\mathcal{A}})$$

$$\forall z \in \Pi(D_P) \quad (9)$$

With the availability of a CLF, the set $\mathcal{A}$ can be asymptotically stabilized if it is possible to synthesize a controller $\mathcal{H}_K$ from inequalities (8) and (9). Such a synthesis is feasible, in particular, for the special case of $\mathcal{H}_K$ being a static state-feedback law $z \mapsto \kappa(z)$. Sufficient conditions guaranteeing the existence of such a controller as well as a particular state-feedback law with point-wise minimum norm are given next.

Given a compact set $\mathcal{A}$ and a control Lyapunov function V (with respect to $\mathcal{A}$), define, for each $r \geq 0$, the set $\mathcal{I}(r) := \{z \in \mathbb{R}^{n_P} : V(z) \geq r\}$. Moreover, for each (z, u_c) and $r \geq 0$, define the function

$$\Gamma_c(z, u_c, r) := \begin{cases} \langle \nabla V(z), F_P(z, u_c) \rangle + \frac{1}{2} \rho(|z|_{\mathcal{A}}) & \text{if } (z, u_c) \in C_P \cap (\mathcal{I}(r) \times \mathbb{R}^{m_{P,c}}), \\ -\infty & \text{otherwise} \end{cases}$$

and, for each (z, u_d) and $r \geq 0$, the function

$$\Gamma_d(z, u_d, r) := \begin{cases} V(G_P(z, u_d)) - V(z) + \frac{1}{2} \rho(|z|_{\mathcal{A}}) & \text{if } (z, u_d) \in D_P \cap (\mathcal{I}(r) \times \mathbb{R}^{m_{P,d}}), \\ -\infty & \text{otherwise} \end{cases}$$

It can be shown that the following conditions involving V and the data (C_P, F_P, D_P, G_P, h) of $\mathcal{H}_P$ guarantee that, for each $r > 0$, there exists a state-feedback law

$$z \mapsto \kappa(z) = (\kappa_c(z), \kappa_d(z))$$

with κ_c continuous on $\Pi(C_P) \cap \mathcal{I}(r)$ and κ_d continuous on $\Pi(D_P) \cap \mathcal{I}(r)$ rendering the compact set

$$\mathcal{A}_r := \{z \in \mathbb{R}^{n_P} : V(z) \leq r\}$$

pre-asymptotically stable for $\mathcal{H}_P$:

(CLF1) C_P and D_P are closed sets, and F_P and G_P are continuous;

(CLF2) The set-valued maps $\Psi_c(z) = \{u_c : (z, u_c) \in C_P\}$ and $\Psi_d(z) = \{u_d : (z, u_d) \in D_P\}$ are lower semicontinuous with convex values;

(CLF3) For every $r > 0$, we have that, for every $z \in \Pi(C_P) \cap \mathcal{I}(r)$, the function $u_c \mapsto \Gamma_c(z, u_c, r)$ is convex on $\Psi_c(z)$ and that, for every $z \in \Pi(D_P) \cap \mathcal{I}(r)$, the function $u_d \mapsto \Gamma_d(z, u_d, r)$ is convex on $\Psi_d(z)$;

In addition to guaranteeing the existence of a (continuous) state-feedback law practically pre-asymptotically stabilizing the set $\mathcal{A}$, these conditions also lead to the following natural definition of the feedback: for every $r > 0$, the state-feedback law pair

$$\kappa_c : \Pi(C_P) \rightarrow \mathbb{R}^{m_{P,c}}, \quad \kappa_d : \Pi(D_P) \rightarrow \mathbb{R}^{m_{P,d}}$$

can be defined on $\Pi(C_P)$ and $\Pi(D_P)$ as

$$\kappa_c(z) := \arg \min \{|u_c| : u_c \in \mathcal{T}_c(z)\} \quad \forall z \in \Pi(C_P) \cap \mathcal{I}(r)$$

$$\kappa_d(z) := \arg \min \{|u_d| : u_d \in \mathcal{T}_d(z)\} \quad \forall z \in \Pi(D_P) \cap \mathcal{I}(r)$$

where $\mathcal{T}_c(z) = \Psi_c(z) \cap \{u_c : \Gamma_c(z, u_c, V(z)) \leq 0\}$ and $\mathcal{T}_d(z) = \Psi_d(z) \cap \{u_d : \Gamma_d(z, u_d, V(z)) \leq 0\}$.

The stability property guaranteed by this feedback is also practical. Under further properties, similar results hold when the input u is not partitioned into u_c and u_d . To achieve asymptotic stability (or stabilizability) of $\mathcal{A}$ with a continuous state-feedback law, extra conditions are required to hold nearby the compact set, which for the case of stabilization of continuous-time systems are the so-called small control

properties. Furthermore, the continuity of the feedback law assures that the closed-loop system has closed flow and jump sets as well as continuous flow and jump maps, which, in turn, due to the compactness of $\mathcal{A}$, implies that the asymptotic stability property is robust. Robustness follows from results for hybrid systems without inputs.

Example 3 (controlled bouncing ball revisited) For the juggling system in Example 1, the feed-

back law $u = \kappa_d \equiv 0$ asymptotically stabilizes its origin – note that for this system, $u_d = u$. In fact, with this feedback, after the first impact (or jump), every solution remains at the origin.

Passivity-Based Control Design

Dissipativity and its special case, passivity, provide a useful physical interpretation of a feedback control system as they characterize the exchange of energy between the plant and its controller. For an open system, passivity (in its very pure form) is the property that the energy stored in the system is no larger than the energy it has absorbed over a period of time. The energy stored in a system is given by the difference between the initial and final energy over a period of time, where the energy function is typically called the storage function. Hence, conveniently, passivity can be expressed in terms of the derivative of a storage function (i.e., the rate of change of the internal energy) and the product between inputs and outputs (i.e., the power flow of the system). Under further observability conditions, this power inequality can be employed as a design tool by selecting a control law that makes the rate of change of the internal energy negative. This method is called *passivity-based control design*.

The passivity-based control design method can be employed in the design of a controller for a “passive” hybrid plant $\mathcal{H}_P$, in which energy might be dissipated during flows, jumps, or both. Passivity notions and a passivity-based control design method for hybrid plants are given next. Since the form of the output of the plant plays a key role in asserting a passivity property, and this property may not necessarily hold both during flows and jumps, as mentioned in the *input and output modeling remark* in section “[Definitions and Notions](#)”, we define outputs y_c and y_d , which, for simplicity, are assumed to be single-valued: $y_c = h_c(x)$ and $y_d = h_d(x)$. Moreover, we consider the case when the dimension of the space of the inputs u_c and u_d coincides with that of the outputs y_c and y_d , respectively, i.e., a “duality” of the output and input space.

Given a compact set $\mathcal{A}$ and functions h_c, h_d such that $h_c(\mathcal{A}) = h_d(\mathcal{A}) = 0$, a hybrid plant $\mathcal{H}_P$ for which there exists a continuously differ-

entiable function $V : \mathbb{R}^{n_P} \rightarrow \mathbb{R}_{\geq 0}$ satisfying for some functions $\omega_c : \mathbb{R}^{m_{P,c}} \times \mathbb{R}^{n_P} \rightarrow \mathbb{R}$ and $\omega_d : \mathbb{R}^{m_{P,c}} \times \mathbb{R}^{n_P} \rightarrow \mathbb{R}$

$$\langle \nabla V(z), F_P(z, u_c) \rangle \leq \omega_c(u_c, z) \forall (z, u_c) \in C \quad (10)$$

$$V(G_P(z, u_d)) - V(z) \leq \omega_d(u_d, z) \forall (z, u_d) \in D \quad (11)$$

is said to be *passive with respect to a compact set* $\mathcal{A}$ if

$$(u_c, z) \mapsto \omega_c(u_c, z) = u_c^\top y_c \quad (12)$$

$$(u_d, z) \mapsto \omega_d(u_d, z) = u_d^\top y_d \quad (13)$$

The function V is the so-called storage function. If (10) holds with ω_c as in (12), and (11) holds with $\omega_d \equiv 0$, then the system is called *flow-passive*, i.e., the power inequality holds only during flows. If (10) holds with $\omega_c \equiv 0$, and (11) holds with ω_d as in (13), then the system is called *jump-passive*, i.e., the energy of the system decreases only during jumps.

Under additional detectability properties, these passivity notions can be used to design static output feedback controllers. In fact, given a hybrid plant $\mathcal{H}_P = (C_P, F_P, D_P, G_P, h)$ satisfying

(PBC1) C_P and D_P are closed sets; F_P and G_P are continuous; and h_c and h_d are continuous;

and a compact set $\mathcal{A}$, it can be shown that if $\mathcal{H}_P$ is flow-passive with respect to $\mathcal{A}$ with a storage function V that is positive definite with respect to $\mathcal{A}$ and has compact sublevel sets, and if there exists a continuous function $\kappa_c : \mathbb{R}^{m_{P,c}} \rightarrow \mathbb{R}^{m_{P,c}}$, $y_c^\top \kappa_c(y_c) > 0$ for all $y_c \neq 0$, such that the resulting closed-loop system with $u_c = -\kappa_c(y_c)$ and $u_d \equiv 0$ has the following properties:

(PBC2) The distance to $\mathcal{A}$ is detectable relative to

$$\{z \in \Pi(C_P) \cup \Pi(D_P) \cup G_P(D_P) :$$

$$h_c(z)^\top \kappa_c(h_c(z)) = 0, (z, -\kappa_c(h_c(z))) \in C_P\};$$

(PBC3) Every complete solution ϕ is such that, for some $\delta > 0$ and some $J \in \mathbb{N}$, we have $t_{j+1} - t_j \geq \delta$ for all $j \geq J$;

then the output-feedback law

$$u_c = -\kappa_c(y_c), \quad u_d \equiv 0$$

renders $\mathcal{A}$ globally pre-asymptotically stable.

In a similar manner, an output-feedback law can be designed when, instead of being flow-passive, $\mathcal{H}_P$ is jump-passive with respect to $\mathcal{A}$. In this case, if the storage function V is positive definite with respect to $\mathcal{A}$ and has compact sublevel sets, and if there exists a continuous function $\kappa_d : \mathbb{R}^{m_{P,d}} \rightarrow \mathbb{R}^{m_{P,d}}$, $y_d^\top \kappa_d(y_d) > 0$ for all $y_d \neq 0$, such that the resulting closed-loop system with $u_c \equiv 0$ and $u_d = -\kappa_d(y_d)$ has the following properties:

(PBC4) The distance to $\mathcal{A}$ is detectable relative to

$$\{z \in \Pi(C_P) \cup \Pi(D_P) \cup G_P(D_P) : h_d(z)^\top \kappa_d(h_d(z)) = 0, (z, -\kappa_d(h_d(z))) \in D_P\};$$

(PBC5) Every complete solution ϕ is Zeno;

then the the output-feedback law

$$u_d = -\kappa_d(y_d), \quad u_c \equiv 0$$

renders $\mathcal{A}$ globally pre-asymptotically stable. Such a feedback design can be employed to globally asymptotically stabilize the controlled bouncing ball in Example 1.

Strict passivity notions can also be formulated for hybrid plants, including the special cases where the power inequalities hold only during flows or jumps. In particular, strict passivity and output strict passivity can be employed to assert asymptotic stability with zero inputs.

Tracking Control Design

While numerous control problems pertain to the stabilization of a set-point condition, at times, it is desired to stabilize the solutions to the plant to a time-varying trajectory. In this section, we consider the problem of designing a hybrid

controller $\mathcal{H}_K$ for a hybrid plant $\mathcal{H}_P$ to track a given reference trajectory r (a hybrid arc). The notion of tracking is introduced below. We propose sufficient conditions that general hybrid plants and controllers should satisfy to solve such a problem. For simplicity, we consider tracking of state trajectories and that the hybrid controller can measure both the state of the plant z and the reference trajectory r ; hence, $v = (z, r)$.

The particular approach used here consists of recasting the tracking control problem as a set stabilization problem for the closed-loop system $\mathcal{H}$. To do this, we embed the reference trajectory r into an augmented hybrid model for which it is possible to define a set capturing the condition that the plant tracks the given reference trajectory. This set is referred to as *the tracking set*. More precisely, given a reference $r : \text{dom } r \rightarrow \mathbb{R}^{n_P}$, we define the set $\mathcal{T}_r$ collecting all of the points (t, j) in the domain of r at which r jumps, that is, every point $(t_j^-, j) \in \text{dom } r$ such that $(t_j^-, j+1) \in \text{dom } r$. Then, the state of the closed loop $\mathcal{H}$ is augmented by the addition of states $\tau \in \mathbb{R}_{\geq 0}$ and $k \in \mathbb{N}$. The dynamics of the states τ and k are such that τ counts elapsed flow time, while k counts the number of jumps of $\mathcal{H}$; hence, during flows $\dot{\tau} = 1$ and $\dot{k} = 0$, while at jumps $\tau^+ = \tau$ and $k^+ = k + 1$. These new states are used to parameterize the given reference trajectory r , which is employed in the definition of the tracking set:

$$\mathcal{A} = \{(z, \eta, \tau, k) \in \mathbb{R}^{n_P} \times \mathbb{R}^{n_K} \times \mathbb{R}_{\geq 0} \times \mathbb{N} : z = r(\tau, k), \eta \in \Phi_K\} \quad (14)$$

This set is the target set to be stabilized for $\mathcal{H}$. The set $\Phi_K \subset \mathbb{R}^{n_K}$ in the definition of $\mathcal{A}$ is some closed set capturing the set of points asymptotically approached by the state of the controller η .

Using results to certify pre-asymptotic stability of closed sets for hybrid systems, sufficient conditions guaranteeing that a hybrid controller $\mathcal{H}_K$ stabilizes the tracking set $\mathcal{A}$ in (14) for the hybrid closed-loop system $\mathcal{H}$ can be formulated. In particular, with $\mathcal{H}$ having state $x = (z, \eta, \tau, k)$ and data

$$\begin{aligned}
C &= \{x : (z, \kappa_c(\eta, z, r(\tau, k))) \in C_P, \tau \in [t_k^r, t_{k+1}^r], (\eta, z, r(\tau, k)) \in C_K\} \\
F(z, \eta, \tau, k) &= (F_P(z, \kappa_c(\eta, z, r(\tau, k))), F_K(\eta, z, r(\tau, k)), 1, 0) \\
D &= \{x : (z, \kappa_c(\eta, z, r(\tau, k))) \in D_P, (\tau, k) \in \mathcal{T}_r\} \cup \\
&\quad \{x : \tau \in [t_k^r, t_{k+1}^r), (\eta, z, r(\tau, k)) \in D_K\} \\
G_1(z, \eta, \tau, k) &= (G_P(z, \kappa_c(\eta, z, r(\tau, k))), \eta, \tau, k + 1), \\
G_2(z, \eta, \tau, k) &= (z, G_K(\eta, z, r(\tau, k)), \tau, k)
\end{aligned}$$

given a complete reference trajectory $r : \text{dom } r \rightarrow \mathbb{R}^{n_P}$ and associated tracking set $\mathcal{A}$, a hybrid controller $\mathcal{H}_K$ with data $(C_K, F_K, D_K, G_K, \kappa)$ guaranteeing that

- (T1) The jumps of r and $\mathcal{H}_P$ occur simultaneously;
(T2) For some continuously differentiable function $V : \mathbb{R}^{n_P} \times \mathbb{R}^{n_K} \times \mathbb{R}_{\geq 0} \times \mathbb{N} \rightarrow \mathbb{R}$, functions $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$; and continuous, positive definite functions ρ_1, ρ_2, ρ_3 , the following hold:

- (T2a) For all $(z, \eta, \tau, k) \in C \cup D \cup G_1(D) \cup G_2(D)$

$$\begin{aligned}
\alpha_1(|(z, \eta, \tau, k)|_{\mathcal{A}}) &\leq V(z, \eta, \tau, k) \leq \\
&\alpha_2(|(z, \eta, \tau, k)|_{\mathcal{A}})
\end{aligned}$$

- (T2b) For all $(z, \eta, \tau, k) \in C$ and all $\zeta \in F(z, \eta, \tau, k)$,

$$\langle \nabla V(z, \eta, \tau, k), \zeta \rangle \leq -\rho_1(|(z, \eta, \tau, k)|_{\mathcal{A}})$$

- (T2c) For all $(z, \eta, \tau, k) \in D_1$ and all $\zeta \in G_1(z, \eta, \tau, k)$

$$V(\zeta) - V(z, \eta, \tau, k) \leq -\rho_2(|(z, \eta, \tau, k)|_{\mathcal{A}})$$

- (T2d) For all $(z, \eta, \tau, k) \in D_2$ and all $\zeta \in G_2(z, \eta, \tau, k)$

$$V(\zeta) - V(z, \eta, \tau, k) \leq -\rho_3(|(z, \eta, \tau, k)|_{\mathcal{A}})$$

renders $\mathcal{A}$ is globally pre-asymptotically stable for $\mathcal{H}$.

Note that condition (T1) imposes that the jumps of the plant and of the reference trajectory occur simultaneously. Though restrictive, at

times, this property can be enforced by proper design of the controller.

Forward Invariance-Based Control Design

As defined in section “[Definitions and Notions](#)”, a set K is forward invariant if every solution to the system from K stays in K . Also known as *flow-invariance*, *positively invariance*, *viability*, or just *invariance*, this property is very important in feedback control design. In fact, asymptotically stabilizing feedback laws induce forward invariance of the set $\mathcal{A}$ that is asymptotically stabilized. Forward invariance is also key to guarantee safety properties, since safety can be typically recast as forward invariance of the set that excludes every point (and, for robustness, a neighborhood of it) for which the system is considered to be unsafe. In this section, forward invariance (or, equivalently, safety) is guaranteed by infinitesimal conditions that involve functions of the state known as barrier functions.

Given a hybrid closed-loop system $\mathcal{H} = (C, F, D, G)$, a function $B : \mathbb{R}^n \rightarrow \mathbb{R}$ is a *barrier function candidate* defining a set $K \subset C \cup D$ if

$$K = \{x \in C \cup D : B(x) \leq 0\} \quad (15)$$

In some settings, the set K might be given a priori and then one would seek for a barrier function B such that (15) holds; namely, find a function B that is nonpositive at points in K only. In some other settings, one may generate the set K from the given sets C, D and the given function B .

A function candidate B is a *barrier function* if, in addition, its change along every solution ϕ to $\mathcal{H}$ that starts from K is such that

$$(t, j) \mapsto B(x(t, j))$$

is nonpositive. One way to guarantee such property is as follows. Given a hybrid system $\mathcal{H} = (C, F, D, G)$, suppose the barrier function candidate B defines a closed K as in (15). Furthermore, suppose B is continuously differentiable. Then, B is said to be a barrier function if, for some $\rho > 0$,

$$\begin{aligned} \langle \nabla B(x), \xi \rangle &\leq 0 \quad \forall x \in ((K + \rho\mathbb{B}) \setminus K) \cap C, \\ \forall \xi &\in F(x) \cap T_C(x) \end{aligned} \quad (16)$$

$$B(\xi) \leq 0 \quad \forall \xi \in G(D \cap K) \quad (17)$$

$$G(D \cap K) \subset C \cup D \quad (18)$$

The barrier function notion introduced above for $\mathcal{H}$ can be formulated for a hybrid plant $\mathcal{H}_P$ given as in (1). In such a setting, since the input to $\mathcal{H}_P$ is not yet assigned, the conditions in (16), (17), and (18) would depend on the input – similar to the conditions that control Lyapunov functions in section “CLF-Based Control Design” have to satisfy. Such an extension is illustrated in the next example for the bouncing ball system, which, since the input only affects the jumps, conditions (17) and (18) become

$$\begin{aligned} \forall z \in \Pi(D_P) \exists u_d \text{ such that } (z, u_d) \in D_P \text{ and} \\ \begin{cases} B(\xi) \leq 0 & \forall \xi \in G_P(z, u_d) \\ G_P(z, u_d) \subset \Pi(C_P) \cup \Pi(D_P) \end{cases} \end{aligned} \quad (19)$$

Example 4 (controlled bouncing ball revisited) Consider the problem of keeping the total energy of the juggling system in Example 1 less than or equal to a constant $V^* \geq 0$. The total energy of the hybrid plant therein is given by

$$V(z) = \gamma z_1 + \frac{1}{2} z_2^2$$

Then, the desired set K to render invariant is defined by the continuously differentiable barrier candidate

$$B(z) := V(z) - V^* \quad \forall z \in \mathbb{R}^2$$

In fact, for the case of a hybrid plant, the set K in (15) collects all points in $z \in \Pi(C_P) \cup \Pi(D_P)$

such that $V(z) \leq V^*$. It follows that

$$\langle \nabla B(z), F_P(z) \rangle = 0 \quad \forall z \in \Pi(C_P)$$

since, during flows, the total energy remains constant. At jumps, the following hold – recall that $u_d = u$: for each $z \in \Pi(D_P) = \{z \in \mathbb{R}^2 : z_1 = 0, z_2 \leq 0\}$,

$$G_P(z, u) = \begin{bmatrix} 0 \\ u \end{bmatrix}$$

and

$$\begin{aligned} B(G_P(z, u)) &= V(G_P(z, u)) - V^* \\ &= \gamma z_1 + \frac{1}{2} u^2 - V^* = \frac{1}{2} u^2 - V^* \end{aligned}$$

Hence, for B to be a barrier function, for each $z \in \Pi(D_P)$, we pick u to satisfy

$$|u| \leq \sqrt{2V^*}$$

In this way, any state feedback $z \mapsto \kappa_d(z)$ such that $|\kappa_d(z)| \leq \sqrt{2V^*}$ for each $z \in \Pi(D_P)$ leads to a hybrid closed-loop system with K forward invariant (It is easy to show that every maximal solution to such a closed loop is complete.). Note that assigning u to a feedback that is positive (when possible) leads to solutions that, after a jump, flow for some time.

Temporal Logic

Design specifications for control design typically include requirements that go beyond asymptotic stability properties, such as finite-time properties and safety constraints, some of which need to be satisfied at specific times rather than in the limit. A framework suitable for handling such specifications is linear temporal logic (LTL). LTL permits the formulation of desired properties such as *safety*, or equivalently, “something bad never happens,” and *liveness*, namely, “something good eventually happens” in finite time.

In LTL, a formula (or sentence) is given in terms of atomic propositions that are combined using Boolean and temporal operators. An atomic

proposition is a function of the state that, for each possible value of the state, is either true or false. More precisely, for $\mathcal{H}$ in (7), a proposition α is such that $\alpha(x)$ is either **True** (1 or $\top$) or **False** (0 or $\perp$). Boolean operators include the following: $\neg$ is the *negation* operator; $\vee$ is the *disjunction* operator; $\wedge$ is the *conjunction* operator; $\Rightarrow$ is the *implication* operator; and $\Leftrightarrow$ is the *equivalence* operator. A way to reason about a solution defined over hybrid time is needed to introduce temporal operators. This is defined by the semantics of LTL, as follows.

Given a solution ϕ to $\mathcal{H}$, a proposition α being **True** at $(t, j) \in \text{dom } \phi$ is denoted by

$$\phi(t, j) \models \alpha$$

If α is **False** at $(t, j) \in \text{dom } \phi$, then we write

$$\phi(t, j) \not\models \alpha$$

Similarly, given an LTL formula f , we say that it is satisfied by ϕ at (t, j) if

$$(\phi, (t, j)) \models f$$

while f not being satisfied at (t, j) is denoted by

$$(\phi, (t, j)) \not\models f$$

The temporal operators are defined as follows: with α and $\mathbf{b}$ being two atomic propositions

- $\bigcirc$ is the *next* operator: $(\phi, (t, j)) \models \bigcirc \alpha$ if and only if

$$(t, j+1) \in \text{dom } \phi \text{ and } (\phi, (t, j+1)) \models \alpha$$

- $\diamond$ is the *eventually* operator: there exists $(t', j') \in \text{dom } \phi$, $t' + j' \geq t + j$ such that $(\phi, (t', j')) \models \alpha$

- $\square$ is the *always* operator: $(\phi, (t, j)) \models \square \alpha$ if and only if, for each $t' + j' \geq t + j$, $(t', j') \in \text{dom } \phi$,

$$(\phi, (t', j')) \models \alpha$$

- $\mathcal{U}_s$ is the strong *until* operator: $(\phi, (t, j)) \models \alpha \mathcal{U}_s \mathbf{b}$ if and only if there exists $(t', j') \in \text{dom } \phi$, $t' + j' \geq t + j$, such that

$$(\phi, (t', j')) \models \mathbf{b}$$

and for all $(t'', j'') \in \text{dom } \phi$ such that $t + j \leq t'' + j'' < t' + j'$,

$$(\phi, (t'', j'')) \models \alpha$$

- $\mathcal{U}_w$ is the weak *until* operator: $(\phi, (t, j)) \models \alpha \mathcal{U}_w \mathbf{b}$ if and only if either

$$(\phi, (t', j')) \models \alpha$$

for all $(t', j') \in \text{dom } \phi$ such that $t' + j' \geq t + j$, or

$$(\phi, (t, j)) \models \alpha \mathcal{U}_s \mathbf{b}$$

Similar semantics apply to a formula f .

Example 5 (Thermostat system revisited) Consider the thermostat system in Example 2. Suppose that the goal is to keep the temperature within the range $[z_{\min}, z_{\max}]$ when the temperature starts in that region, and when it does not start from that range, steer it to that range in finite time and, after that, remain in that range for all time. For simplicity, suppose that the second input is constant and given by $u_2 \equiv z_{\text{out}}$, with $z_{\text{out}} \in (-\infty, z_{\max}]$ and $z_{\text{out}} + z_{\Delta} \in [z_{\min}, \infty)$. It can be shown that the following hybrid controller $\mathcal{H}_K$ accomplishes the state goal: with $\eta \in \{0, 1\}$, and with dynamics

$$\mathcal{H}_K : \begin{cases} \dot{\eta} \in F_K(\eta, v) := 0 & (\eta, v) \in C_K := (\{0\} \times C_{K,0}) \cup (\{1\} \times C_{K,1}) \\ \eta^+ \in G_K(\eta, v) := 1 - \eta & (\eta, v) \in D_K := (\{0\} \times D_{K,0}) \cup (\{1\} \times D_{K,1}) \\ \zeta = \kappa(\eta, v) := \eta \end{cases} \quad (20)$$

where

$$C_{K,0} := \{v : v \geq z_{\min}\},$$

$$C_{K,1} := \{v : v \leq z_{\max}\}$$

$$D_{K,0} := \{v : v \leq z_{\min}\},$$

$$D_{K,1} := \{v : v \geq z_{\max}\}$$

The input of the controller is assigned via $v = z$ and its output assigned u via $u = \zeta = \eta$. Furthermore, it can be shown that the hybrid closed-loop system satisfies the following LTL formulae: with

$$\begin{aligned} \alpha(z, \eta) &= 1 & \text{if } z \in [z_{\min}, z_{\max}], \\ \alpha(z, \eta) &= 0 & \text{if } z \notin [z_{\min}, z_{\max}], \end{aligned}$$

the following hold:

- $\Box a$ for every solution with initial temperature in $[z_{\min}, z_{\max}]$, regardless of the initial value of η .
- $\Diamond a$, $\Diamond \Box a$, and $\Box \Diamond a$ for every solution.

Sufficient conditions involving Lyapunov functions for finite-time attractivity and barrier functions can be employed to guarantee that certain formulas are satisfied. The following table provides pointers to such results (Table 1).

Summary and Future Directions

Advances over the last decade on modeling and robust stability of hybrid dynamical systems (without control inputs) have paved the road for the development of systematic methods for the design of control algorithms for hybrid plants. The results selected for this short expository entry, along with recent efforts on multi-mode/logic-based control, event-based control, and backstepping, which were not covered here, are scheduled to appear. Future research

Hybrid Dynamical Systems, Feedback Control of, Table 1 Sufficient conditions for LTL formulas involving temporal operators $\Box$, $\Diamond$, $\mathcal{U}_s$, and $\mathcal{U}_w$

f	Sufficient conditions in the literature
$\Box a$	Properties of the data of $\mathcal{H}$ – (Han and Sanfelice 2018, Sections 4.3 and 5.3)
$\Box a$	Forward invariance – Chai and Sanfelice (2019), Maghenem and Sanfelice (2018), and (Han and Sanfelice 2018, Section 5.1)
$\Diamond a$	Finite-time attractivity – Li and Sanfelice (2019) and (Han and Sanfelice 2018, Section 5.2)
$\alpha \mathcal{U}_s b$	Forward invariance and finite-time attractivity – (Han and Sanfelice 2018, Section 5.4)
$\alpha \mathcal{U}_w b$	Forward invariance or finite-time attractivity – (Han and Sanfelice 2018, Section 5.4)

directions include the development of more powerful tracking control design methods, state observers, and optimal controllers for hybrid plants.

Cross-References

- [Hybrid Model Predictive Control](#)
- [Hybrid Observers](#)
- [Modeling Hybrid Systems](#)
- [Simulation of Hybrid Dynamic Systems](#)
- [Stability Theory for Hybrid Dynamical Systems](#)

Bibliography

Set-Valued Dynamics and Variational Analysis

- Aubin J-P, Frankowska H (1990) Set-valued analysis. Birkhauser, Boston
- Rockafellar RT, Wets RJ-B (1998) Variational analysis. Springer, Berlin/Heidelberg

Modeling and Stability

- Branicky MS (2005) Introduction to hybrid systems. In: Hristu-Varsakelis, Dimitrios, Levine, William S. (Eds.) Handbook of networked and embedded control systems. Springer, Boston, pp 91–116

- Goebel R, Sanfelice RG, Teel AR (2012) Hybrid dynamical systems: modeling, stability, and robustness. Princeton University Press, Princeton
- Haddad WM, Chellaboina V, Nersisov SG (2006) Impulsive and hybrid dynamical systems: stability, dissipativity, and control. Princeton University Press, Princeton
- Lygeros J, Johansson KH, Simić SN, Zhang J, Sastry SS (2003) Dynamical properties of hybrid automata. 48(1):2–17
- van der Schaft A, Schumacher H (2000) An introduction to hybrid dynamical systems. Lecture notes in control and information sciences. Springer, Berlin

Control

- Biemond JJB, van de Wouw N, Heemels WPMH, Nijmeijer H (2013) Tracking control for hybrid systems with state-triggered jumps. *IEEE Trans Autom Control* 58(4):876–890
- Chai J, Sanfelice RG (2019) Forward invariance of sets for hybrid dynamical systems (part I). *IEEE Trans Autom Control* 64(6):2426–2441
- Forni F, Teel AR, Zaccarian L (2013) Follow the bouncing ball: global results on tracking and state estimation with impacts. *IEEE Trans Autom Control* 58(6):1470–1485
- Han H, Sanfelice RG (2018) Linear temporal logic for hybrid dynamical systems: characterizations and sufficient conditions. In: *Proceedings of the 6th analysis and design of hybrid systems*, vol 51, pp 97–102, and [ArXiv \(https://arxiv.org/abs/1807.02574\)](https://arxiv.org/abs/1807.02574)
- Li Y, Sanfelice RG (2019) Finite time stability of sets for hybrid dynamical systems. *Automatica* 100:200–211
- Lygeros J (2005) An overview of hybrid systems control. In: *Handbook of networked and embedded control systems*. Springer, pp 519–538
- Maghenem M, Sanfelice RG (2018) Barrier function certificates for invariance in hybrid inclusions. In: *Proceedings of the IEEE conference on decision and control*, Dec 2018, pp 759–764
- Naldi R, Sanfelice RG (2013) Passivity-based control for hybrid systems with applications to mechanical systems exhibiting impacts. *Automatica* 49(5):1104–1116
- Sanfelice RG (2013a) On the existence of control Lyapunov functions and state-feedback laws for hybrid systems. *IEEE Trans Autom Control* 58(12):3242–3248
- Sanfelice RG (2013b) Control of hybrid dynamical systems: an overview of recent advances. In: Daafouz J, Tarbouriech S, Sigalotti M (eds) *Hybrid systems with constraints*. Wiley, Hoboken, pp 146–177
- Sanfelice RG, Biemond JJB, van de Wouw N, Heemels WPMH (2013) An embedding approach for the design of state-feedback tracking controllers for references with jumps. *Int J Robust Nonlinear Control* 24(11):1585–1608

Hybrid Model Predictive Control

Ricardo G. Sanfelice
Department of Electrical and Computer Engineering, University of California, Santa Cruz, CA, USA

Abstract

Model predictive control is a powerful technique to solve problems with constraints and optimality requirements. Hybrid model predictive control further expands its capabilities by allowing for the combination of continuous-valued with discrete-valued states, and continuous dynamics with discrete dynamics or switching. This short article summarizes techniques available in the literature addressing different aspects of hybrid MPC, including those for discrete-time piecewise affine systems, discrete-time mixed logical dynamical systems, linear systems with periodic impulses, and hybrid dynamical systems.

Keywords

Model predictive control · Receding horizon control · Hybrid systems · Hybrid control · Optimization

Introduction

Model predictive control (MPC) ubiquitously incorporates mathematical models, constraints, and optimization schemes to solve control problems. Through the use of a mathematical model describing the evolution of the system to

This research has been partially supported by the National Science Foundation under Grant no. ECS-1710621 and Grant no. CNS-1544396, by the Air Force Office of Scientific Research under Grant no. FA9550-16-1-0015, Grant no. FA9550-19-1-0053, and Grant no. FA9550-19-1-0169, and by CITRIS and the Banatao Institute at the University of California.

control and the environment, MPC strategies determine the value of the control inputs to apply by solving an optimization problem over a finite time horizon. The system to control, which is known as the *plant*, is typically given by a state-space model with state, input, and outputs. Most MPC strategies **measure** the output of the plant and, using a mathematical model of the plant, **predict** the possible values of the state and the output in terms of free input variables to be determined. The prediction is performed over a *prediction horizon* which is defined as a finite time interval in the future. With such parameterized values of the state and input over the prediction horizon, MPC **solves** an optimization problem. To formulate the optimization control problem (OCP), a cost functional is to be defined. The OCP can also incorporate constraints, such as those on the state, the input, and the outputs of the plant. Then, MPC **selects** an input signal that solves the OCP and **applies** it to the plant for a finite amount of time, which is defined by the *control horizon*. After the input is applied over the control horizon, the process is repeated.

Over the past two decades, MPC has emerged as one of the preferred planning and control engines in robotics and control applications. Such grown interest in MPC has been propelled by recent advances in computing and optimization, which have led to significantly smaller computation times in the solution to certain optimization problems. However, the complexity of the dynamics, constraints, and control objectives in the MPC has increased so as to accommodate the specifics of the applications of interest. In particular, the dynamics of the system to control might substantially change over the region of operation, the constraints or the control objectives may abruptly change upon certain self-triggered or externally-triggered events, and the optimization problems may involve continuous-valued and discrete-valued variables. MPC problems with such features have been studied in the literature under the name *hybrid model predictive control* (Sanfelice 2018). Within a common framework and notation, this entry summarizes

several hybrid MPC approaches available in the literature.

Notation and Preliminaries

Notation

The set of nonnegative integers is denoted as $\mathbb{N} := \{0, 1, 2, \dots\}$ and the set of positive integers as $\mathbb{N}_{>0} := \{1, 2, \dots\}$. Given $N \in \mathbb{N}_{>0}$, we define $\mathbb{N}_{<N} := \{0, 1, 2, \dots, N-1\}$ and $\mathbb{N}_{\leq N} := \{0, 1, 2, \dots, N\}$. The set of real numbers is denoted $\mathbb{R}$ and, given $n \in \mathbb{N}_{>0}$, $\mathbb{R}^n$ is the n -dimensional Euclidean space. The set of nonnegative real numbers is $\mathbb{R}_{\geq 0} := [0, \infty)$ and the set of positive real numbers is denoted as $\mathbb{R}_{>0} := (0, \infty)$. Given $x \in \mathbb{R}^n$, $|x|$ denotes its Euclidean norm and given $p \in [1, \infty]$, $|x|_p$ denotes its p -norm.

Discrete-Time, Continuous-Time, and Hybrid Systems

A general nonlinear discrete-time system takes the form

$$x^+ = f(x, u) \quad (1)$$

$$y = h(x, u) \quad (2)$$

where x is the state, u the input, and y the output. The symbol $^+$ on x is to indicate, when a solution to the system is computed, the new value of the state after each discrete-time step. The parameter $k \in \mathbb{N}$ denotes discrete time. Given an input signal $\mathbb{N} \ni k \mapsto u(k)$, the solution to (1) is given by the function $\mathbb{N} \ni k \mapsto x(k)$ that satisfies

$$x(k+1) = f(x(k), u(k)) \quad \forall k \in \mathbb{N}$$

Similarly, the output associated to this solution is given by

$$y(k) = h(x(k), u(k)) \quad \forall k \in \mathbb{N}$$

A general nonlinear continuous-time system takes the form

$$\dot{x} = f(x, u) \quad (3)$$

$$y = h(x, u) \quad (4)$$

where x is the state, u the input, and y the output. The dot on x is to indicate the change of x with respect to ordinary time. The parameter t denotes continuous time. Given an input signal $\mathbb{R}_{\geq 0} \ni t \mapsto u(t)$, a solution to (3) is given by a function $\mathbb{R}_{\geq 0} \ni t \mapsto x(t)$ that is regular enough to satisfy

$$\frac{d}{dt}x(t) = f(x(t), u(t))$$

at least for almost all $t \in \mathbb{R}_{\geq 0}$. The output associated to this solution is given by

$$y(t) = h(x(t), u(t)) \quad \forall t \in \mathbb{R}_{\geq 0}$$

A hybrid dynamical system takes the form

$$\mathcal{H} : \begin{cases} \dot{x} = F(x, u) & (x, u) \in C \\ x^+ = G(x, u) & (x, u) \in D \end{cases} \quad (5)$$

where x is the state and u the input. The set C defines the set of points in the state and input space from which flows are possible according to the differential equation $\dot{x} = F(x, u)$. The function F is called the flow map. The set D defines the set of points in the state and input space from where jumps are possible according to the difference equation $x^+ = G(x, u)$. The function G is called the jump map. Similar to the model for continuous-time systems in (3) and (4), the dot on x is to indicate the change of x with respect to ordinary time. Ordinary time is denoted by the parameter t , which takes values from $\mathbb{R}_{\geq 0}$. As a difference to the model for discrete-time systems in (1) and (2), the symbol $^+$ on x denotes the new value of the state after a jump is triggered. Every time that a jump occurs, the discrete counter j is incremented by one. A solution to the hybrid system can flow or jump, according to the values of the state and of the input relative to C and D , respectively. Solutions to $\mathcal{H}$ are parameterized by t and j , and are defined on hybrid time domains. The hybrid time domain of a solution x is denoted $\text{dom } x$, which is a subset of $\mathbb{R}_{\geq 0} \times \mathbb{N}$ and has the following structure: there exists a real nondecreasing

sequence $\{t_j\}_{j=0}^J$ with $t_0 = 0$, and when J is finite, $t_{J+1} \in [t_J, \infty]$ such that $\text{dom } x = \bigcup_{j=0}^J I_j \times \{j\}$. Here, $I_j = [t_j, t_{j+1}]$ for all $j < J$. If J is finite, I_{J+1} can take the form $[t_J, t_{J+1}]$ (when $t_{J+1} < \infty$), or possibly $[t_J, t_{J+1})$. It is convenient to consider the solution and the input as a pair (x, u) defined on the same hybrid time domain, namely, $\text{dom}(x, u) = \text{dom } x = \text{dom } u$. Given an input signal $(t, j) \mapsto u(t, j)$, a function $(t, j) \mapsto x(t, j)$ is a solution to the hybrid system if, over intervals of flow, it is locally absolutely continuous and satisfies

$$\frac{d}{dt}x(t, j) = F(x(t, j), u(t, j))$$

when

$$(x(t, j), u(t, j)) \in C$$

and, at jump times, it satisfies

$$x(t, j+1) = G(x(t, j), u(t, j))$$

when

$$(x(t, j), u(t, j)) \in D$$

Hybrid Model Predictive Control Strategies

Most MPC strategies in the literature are for plants given in terms of discrete-time models of the form $x^+ = f(x, u)$, where x is the state and u is the input. In particular, the case of f being linear has been widely studied and documented ► [Model Predictive Control in Practice](#). The cases in which the function f is discontinuous, or when either x or u have continuous-valued and discrete-value components require special treatment. In the literature, such cases are referred to as *hybrid*: either the system or the MPC strategy are said to be hybrid. Another use of the term hybrid is when the control algorithm obtained from MPC is nonsmooth, e.g., when the MPC strategy selects a control law from a family or when the MPC strategy is implemented using sample and hold. Finally, the term hybrid has been also employed in the literature to indicate that the plant exhibits state jumps at certain events. In such settings,

the plant is modeled by the combination of continuous dynamics and discrete dynamics. In this section, after introducing basic definitions and notions, we present hybrid MPC strategies in the literature, starting from those that are for purely discrete-time systems (albeit with nonsmooth right-hand side) and ending at those for systems that are hybrid due to the combination of continuous and discrete dynamics.

MPC for Discrete-Time Piecewise Affine Systems

Piecewise affine (PWA) systems in discrete time take the form

$$x^+ = A_i x + B_i u + f_i \quad (6)$$

$$y = C_i x + D_i u \quad (7)$$

$$\text{subject to } x \in \Omega_i, u \in \mathcal{U}_i(x), i \in S \quad (8)$$

where x is the state, u the input, and y the output. The origin of (6), (7), and (8) is typically assumed to be an equilibrium state for zero input u . The set $S := \{1, 2, \dots, \bar{s}\}$ with $\bar{s} \in \mathbb{N}_{>0}$ is a finite set to index the different values of the matrices and constraints. For each $i \in S$, the constant matrices $(A_i, B_i, f_i, C_i, D_i)$ define the dynamics of x and the output y . The elements f_i of the collection is such that $f_i = 0$ for all $i \in S$ such that $0 \in \Omega_i$. For each $i \in S$, the state constraints are determined by Ω_i and the state-dependent input constraints by $\mathcal{U}_i(x)$. The collection $\{\Omega_i\}_{i=1}^{\bar{s}}$ is a collection of polyhedra such that

$$\bigcup_{i \in S} \Omega_i = \mathcal{X}$$

where $\mathcal{X} \subset \mathbb{R}^n$ is the region of operation of interest. Moreover, the polyhedra are typically such that their interiors do not intersect, namely,

$$\text{int}(\Omega_i) \cap \text{int}(\Omega_j) = \emptyset \quad \forall i, j \in S : i \neq j$$

For each $x \in \Omega_i$, the set $\mathcal{U}_i(x)$ defines the allowed values for the input u .

For this class of systems, an MPC strategy consists of

1. At the current state of the plant, solve the OCP over a discrete prediction horizon;
2. Apply the optimal control input over a discrete control horizon;
3. Repeat.

The OCP associated to this strategy is as follows: given

- the current state x_0 of (6), (7), and (8),
- a prediction horizon $N \in \mathbb{N}_{>0}$,
- a terminal constraint set $\mathcal{X}_f$,
- a stage cost $\mathcal{L}$, and
- a terminal cost $\mathcal{F}$

the MPC problem consists of minimizing the cost functional

$$\mathcal{J}(x, i, u) := \sum_{k=0}^{N-1} \mathcal{L}(x(k), i(k), u(k)) + \mathcal{F}(x(N))$$

over solutions $k \mapsto x(k)$, indices sequence $k \mapsto i(k)$, and inputs $k \mapsto u(k)$ subject to

$$x(0) = x_0 \quad (9)$$

$$x(N) \in \mathcal{X}_f \quad (10)$$

$$\forall k \in \mathbb{N}_{<N} : \quad x(k+1) = A_{i(k)}x(k) + B_{i(k)}u(k) + f_{i(k)} \quad (11)$$

$$\forall k \in \mathbb{N}_{\leq N} : \quad y(k) = C_{i(k)}x(k) + D_{i(k)}u(k) \quad (12)$$

$$x(k) \in \Omega_{i(k)} \quad (13)$$

$$u(k) \in \mathcal{U}_{i(k)}(x(k)) \quad (14)$$

$$i(k) \in S \quad (15)$$

A few observations about the OCP above are in order. The solution $k \mapsto x(k)$ is uniquely defined by x_0 and $k \mapsto (i(k), u(k))$. The condition in (9) imposes the initial condition for the solution $k \mapsto x(k)$, while (10) restricts the value of x at the end of the prediction horizon. The condition in (11) is to enforce that $k \mapsto x(k)$ is a solution for the input $k \mapsto u(k)$. Similarly, (12) imposes that an output is generated according to the dynamics of the PWA system in (6), (7), and (8). The conditions in (13) and (14) enforce the state and (state-dependent) input constraints associated with the PWA system. Finally, (15) forces the indexing signal $k \mapsto i(k)$ to belong to the finite set of possible indices S . Typical choices of the functions $\mathcal{L}$ and $\mathcal{F}$ in the cost functional $\mathcal{J}$ are

$$\begin{aligned}\mathcal{L}(x, i, u) &:= |Q_i x|_p + |R_i u|_p, \\ \mathcal{F}(x) &:= |P x|_p\end{aligned}$$

for some $p \in [1, \infty]$, where, for each $i \in S$, Q_i and R_i and P are matrices of appropriate dimensions.

Further Reading

For key properties of the MPC problem for PWA systems in this section, the reader is referred to Lazar et al. (2006). In particular, sufficient conditions for recursive feasibility and asymptotic stability of the origin of the PWA system appeared in Lazar et al. (2006, Theorem III.2). In that reference, the reader can also find insight on how to select the terminal cost and the terminal constraint set; see Lazar et al. (2006, Section IV and Section V). The same reference also provides insight on how to solve the OCP using off-the-shelf tools. Very importantly, when $p = 1$ or $p = \infty$, the OCP can be rewritten as a mixed integer linear program (MILP). Furthermore, when the stage and terminal costs are quadratic, the OCP can be rewritten as a mixed integer quadratic program (MIQP).

MPC for Discrete-Time Systems with Continuous and Discrete-Valued States

Mixed Logical Dynamical (MLD) systems are discrete-time systems involving states, inputs,

and outputs that have continuous-valued and discrete-valued components. MLD systems are given by

$$x^+ = Ax + B_1 u + B_2 \delta + B_3 z + B_4 \quad (16)$$

$$y = Cx + D_1 u + D_2 \delta + D_3 z + D_4 \quad (17)$$

$$\text{subject to } E_2 \delta + E_3 z \leq E_1 u + E_4 x + E_5 \quad (18)$$

where x is the state, u the input, y the output, z continuous-valued auxiliary variables, and δ discrete-valued auxiliary variables. The state, the input, and the output have continuous-valued and discrete-valued components; for example, certain components take values from Euclidean spaces while others from discrete sets, like $\{0, 1\}$. The partition of the state x is typically given by $x = (x_c, x_\ell)$, with x_c being the continuous-valued components and x_ℓ the discrete-valued components of x ; similarly for the input $u = (u_c, u_\ell)$ and for the output $y = (y_c, y_\ell)$. The matrices A , $\{B_i\}_{i=1}^3$, B_4 , C , $\{D_i\}_{i=1}^3$, and D_4 define the dynamics of x and the output y . The matrices $\{E_i\}_{i=1}^5$ are used in (18) to define constraints coupling the continuous-valued and discrete-valued states. The latter matrices can be properly defined to recast constraints into properties of discrete-valued states. For instance, when $x \in [-x_{\max}, x_{\max}]$ with $x_{\max} \geq 0$, the constraint $x \geq 0$ is equivalent to $\delta = 1$ (and, in turn, $x < 0$ is equivalent to $\delta = 0$) when, for some $\varepsilon > 0$, the following inequalities hold:

$$x_{\max} \delta \leq x + x_{\max}, \quad -(x_{\max} + \varepsilon) \delta \leq -x - \varepsilon \quad (19)$$

This constraint can be written as (18) with $E_1 = 0$, $E_2 = [x_{\max} \quad -(x_{\max} + \varepsilon)]^\top$, $E_3 = 0$, $E_4 = [1 \quad -1]^\top$, and $E_5 = [x_{\max} \quad \varepsilon]^\top$.

For this class of systems, the MPC strategy in section “[MPC for Discrete-Time Piecewise Affine Systems](#)” is typically employed, and the OCP associated to it is as follows: given

- the current state x_0 of (16), (17), and (18),
- a prediction horizon $N \in \mathbb{N}_{>0}$,

- a terminal constraint set $\mathcal{X}_f$,
- a stage cost $\mathcal{L}$, and
- a terminal cost $\mathcal{F}$

$$\mathcal{J}(x, z, \delta, u)$$

the MPC problem consists of minimizing the cost functional

over solutions $k \mapsto x(k)$, auxiliary variables $k \mapsto z(k)$ and $k \mapsto \delta(k)$, and inputs $k \mapsto u(k)$ subject to

$$x(0) = x_0 \quad (20)$$

$$x(N) \in \mathcal{X}_f \quad (21)$$

$$\forall k \in \mathbb{N}_{<N} : \quad x(k+1) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) + B_4 \quad (22)$$

$$\forall k \in \mathbb{N}_{\leq N} : \quad y(k) = Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) + D_4 \quad (23)$$

$$E_2\delta(k) + E_3z(k) \leq E_1u(k) + E_4x(k) + E_5 \quad (24)$$

H

A particular choice of the cost functional $\mathcal{J}$ used in the MLD literature is

$$\begin{aligned} \mathcal{J}(x, z, \delta, u) = & \sum_{k=0}^{N-1} \mathcal{L}(x(k), z(k), \delta(k), u(k)) \\ & + \mathcal{F}(x(N)) \end{aligned}$$

where the stage and terminal costs are given by

$$\begin{aligned} \mathcal{L}(x, z, \delta, u) := & |Qx|_p + |Q_z z|_p \\ & + |Q_\delta \delta|_p + |Ru|_p, \\ \mathcal{F}(x) := & |Px|_p \end{aligned}$$

for some $p \in [1, \infty]$ and constant matrices Q , R , Q_δ , and Q_z of appropriate dimension.

Further Reading

Chapter 18 of Borrelli et al. (2017) provides a detailed presentation of MPC for MLD systems as in (16), (17), and (18). Following the ideas in recasting state constraints using discrete-valued auxiliary states as in (19), the MPC problem for MLD systems is formulated therein as mixed integer (linear and quadratic) problems. Previous articles about MLD systems include Bemporad et al. (2000a,b, 2002), Lazar et al. (2006), and Borrelli et al. (2017). Mixed-integer optimization methods were also used in Karaman et al. (2008) to solve feasibility and optimization

problems for MLD systems with temporal logic specifications.

Periodic MPC for Continuous-Time Systems

MPC can be directly applied to plants given by continuous-time nonlinear systems modeled as in (3) and (4), without discretization of the dynamics. The price to pay is that the optimal control input has to be determined over a finite-time window of continuous time. An MPC strategy for such systems consists of:

1. At the current state of the plant, solve the OCP over a continuous-time prediction horizon;
2. Apply the optimal control input over a continuous-time control horizon;
3. Repeat.

The OCP to solve is as follows: given

- the current state x_0 of (3),
- a prediction horizon $T \in \mathbb{R}_{>0}$,
- a terminal constraint set $\mathcal{X}_f$,
- a stage cost $\mathcal{L}$, and
- a terminal cost $\mathcal{F}$

minimize the cost functional

$$\mathcal{J}(x, u) := \int_0^T \mathcal{L}(x(t), u(t))dt + \mathcal{F}(x(T))$$

over solutions $t \mapsto x(t)$ and inputs $t \mapsto u(t)$ subject to

$$x(0) = x_0 \quad (25)$$

$$x(T) \in \mathcal{X}_f \quad (26)$$

$$\forall t \in (0, T) : \frac{d}{dt}x(t) = f(x(t), u(t)) \quad (27)$$

$$\forall t \in [0, T] : y(t) = h(x(t), u(t)) \quad (28)$$

$$u(t) \in \mathcal{U} \quad (29)$$

Typically, the right-hand side f is assumed to be twice continuously differentiable to assure uniqueness of solutions. Furthermore, the zero state is typically assumed to be an equilibrium point for (3) with zero input. The input constraint set, which is denoted by $\mathcal{U}$, is commonly assumed to be compact and convex, and to have the property that the origin belongs to its interior.

Further Reading

One of the first articles on MPC for continuous-time systems is Mayne and Michalska (1990). A reference that is more closely related to the MPC algorithm presented in this section is Chen and Allgöwer (1998). The approach in this article is to select the terminal constraint set $\mathcal{X}_f$ as a forward invariant neighborhood of the origin. The invariance-inducing feedback law is taken to be linear. In Chen and Allgöwer (1998), a method to design the feedback, the terminal constraint set, and the terminal cost is provided. Due to their design approach leading to a cost functional that upper bounds the cost of the associated infinite horizon control problem, the MPC strategy in Chen and Allgöwer (1998) is called quasi-infinite horizon nonlinear MPC. While the work in Chen and Allgöwer (1998) allows for general piecewise continuous inputs as candidates for the optimal control, the particular case studied in Magni and Scattolini (2004) considers only piecewise-constant functions that change values periodically. The algorithm in the latter reference results in a type of sample-and-hold MPC strategy. Similar ideas were employed in Nešić

and Grüne (2006) for the purposes of finding a sampled version of a continuous-time controller via MPC.

MPC for Linear Systems with Periodic Impulses

Linear systems with periodic impulses on the state are given by

$$\dot{x}(t) = Ax(t) \quad \forall t \in (k\delta, (k+1)\delta] \quad (30)$$

$$x(t^+) = x(t) + Bu_k \quad \forall t = k\delta \quad (31)$$

for each $k \in \mathbb{N}$, where $t \mapsto x(t)$ is a left-continuous function and, for each impulse time, namely, each $t \in \{0, \delta, 2\delta, \dots\}$, $x(t^+)$ is the right limit of $x(t)$. Similarly to the MPC strategies presented in the earlier sections, an output can be attached to the model in (30) and (31). The constant $\delta > 0$ is the sampling period, and u_k is the constant input applied at the impulse time $t = k\delta$. As shown in the literature, the sequence of constant inputs $\{u_k\}_{k \in \mathbb{N}}$ can be determined using the following MPC strategy: for each $k \in \mathbb{N}$,

1. At the current state of the plant at $t = k\delta$, solve the OCP over a prediction horizon including $N - 1$ future impulse times;
2. Apply the first entry of the optimal input to the plant until the next impulse time.

The OCP to solve is as follows: given

- the current state x_0 of (30) and (31),
- a prediction horizon $N \in \mathbb{N}_{>0}$,
- a terminal constraint set $\mathcal{X}_f$, and
- a stage cost $\mathcal{L}$

minimize the cost functional

$$\mathcal{J}(x, u) = \sum_{k=0}^{N-1} \mathcal{L}(x(\tau_k), u(\tau_k))$$

over the value of the solutions and inputs at the impulse times, which are denoted as $x(k)$ and $u(k)$, respectively, subject to

$$x(0) = x_0 \quad (32)$$

$$x(N\delta) \in \mathcal{X}_f \quad (33)$$

$$\forall k \in \mathbb{N}_{<N} : \quad \dot{x}(t) = Ax(t) \quad \forall t \in (k\delta, (k+1)\delta] \quad (34)$$

$$x(t^+) = x(t) + Bu(t) \quad \forall t = k\delta \quad (35)$$

$$x(t) \in \mathcal{X} \quad \forall t \in [k\delta, (k+1)\delta] \quad (36)$$

$$u(\tau_k) \in \mathcal{U} \quad (37)$$

The cost functional given above is a particular choice used in the literature, which does not include a terminal cost. The set $\mathcal{X}$ denotes the state constraints and the set $\mathcal{U}$ the input constraints.

Further Reading

The formulation above is based on Sopasakis et al. (2015). The developments therein further exploit the periodicity of the setting and employ the fact that the solutions to the system in (30) and (31) can be directly obtained at the impulse times from the computation of the solutions to the discrete-time system $x^+ = \exp(A\delta)(x + Bu)$. In addition, Sopasakis et al. (2015) employs over approximation techniques with the goal of reducing the number of constraints associated with the OCP above. For this purpose, polytopic over-approximations of the continuous dynamics of the impulsive system are proposed therein so as to arrive at a convex quadratic program. It should be pointed out that the stability concept used in Sopasakis et al. (2015) is weak in the sense that closeness and convergence of the values of the solution are only required to occur at the impulse times. Due to the combination of features of impulsive systems and of sample-data systems, the MPC strategy in Sopasakis et al. (2015) is one of the MPC approaches found in the literature that is closest to hybrid dynamical systems, as introduced in the next section.

MPC for Hybrid Dynamical Systems

Hybrid dynamical systems are systems with state variables that are allowed to evolve continuously and, at times, jump. A mathematical model of a hybrid system, denoted $\mathcal{H}$, was given in (5). This model is general enough to capture the

main features of other hybrid system modeling formalisms. For the purposes of MPC, it should be noted that the flow set C and the jump set D in (5) can already accommodate state and input constraints. Furthermore, compared to the MPC strategies for discrete-time systems and for continuous-time systems introduced earlier, the fact that hybrid systems have solution pairs (x, u) defined on a hybrid time domain $\text{dom}(x, u)$, which is parameterized by ordinary time t and jump time j , requires a prediction horizon that includes both prediction of ordinary time and of jump times. One such prediction horizon is given by the set

$$\mathcal{T} := \{(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N} : \max\{t/\delta, j\} = \tau\} \quad (38)$$

for a positive integer τ and some positive constant δ . These parameters determine the “length” of $\mathcal{T}$ and the step size during flow for prediction. With this construction, the OCP to solve in an MPC strategy for hybrid systems will require that the terminal time, which now will be given by a pair (T, J) , satisfies $\max\{T/\delta, J\} = \tau$. A hybrid control horizon, parameterized by a positive integer $\tau_c < \tau$ and positive constant $\delta_c < \delta$, with the same structure as the prediction horizon is defined.

With such a horizon structure, an MPC strategy for a plant modeled as a hybrid dynamical system $\mathcal{H}$ is as follows:

- (1) At the current state of the plant, solve the OCP over the hybrid prediction horizon;
- (2) Apply the optimal hybrid input to the plant over the hybrid control horizon;
- (3) Repeat.

The OCP to solve as part of this MPC strategy is as follows: given

- the current state x_0 of (5),
- prediction horizon parameters $\tau \in \mathbb{N}_{>0}$ and $\delta > 0$,
- control horizon parameters $\tau_c \in \mathbb{N}_{<\tau} \setminus \{0\}$ and $\delta_c \in (0, \delta)$,
- a terminal constraint set $\mathcal{X}_f$,
- stage costs $\mathcal{L}_C$ and $\mathcal{L}_D$, and
- a terminal cost $\mathcal{F}$

the MPC problem consists of minimizing the cost functional

$$\begin{aligned} \mathcal{J}(x, u) := & \left(\sum_{j=0}^J \int_{t_j}^{t_{j+1}} \mathcal{L}_C(x(t, j), u(t, j)) dt \right) \\ & + \left(\sum_{j=0}^{J-1} \mathcal{L}_D(x(t_{j+1}, j), u(t_{j+1}, j)) \right) \\ & + \mathcal{F}(x(T, J)) \end{aligned}$$

over solution pairs $(t, j) \mapsto (x(t, j), u(t, j))$ with hybrid time domain that is a compact subset of $[0, T] \times \mathbb{N}_{\leq J}$, subject to

$$x(0, 0) = x_0 \quad (39)$$

$$(T, J) \in \mathcal{T} \quad (40)$$

$$x(T, J) \in \mathcal{X}_f \quad (41)$$

$$\begin{aligned} \forall j \in \mathbb{N} \text{ with } \text{int}(I_j) \neq \emptyset : \quad & \frac{d}{dt}x(t, j) = F(x(t, j), u(t, j)) \quad (x(t, j), u(t, j)) \in C \\ & \text{for almost all } t \in I_j, \quad (42) \end{aligned}$$

$$\forall (t, j) \in \text{dom } x \text{ s.t. } (t, j+1) \in \text{dom } x : \quad x(t, j+1) = G(x(t, j), u(t, j)) \quad (x(t, j), u(t, j)) \in D \quad (43)$$

where (T, J) denotes the terminal time of the solution pair (x, u) and $\{t_j\}_{j=0}^{J+1}$ is the sequence defining $\text{dom}(x, u)$, with $t_{J+1} = T$. The function $\mathcal{L}_C$ defines the cost of flowing on the flow set C and the function $\mathcal{L}_D$ defined the cost of jumping from the jump set.

Further Reading

For more details about the MPC strategy for hybrid dynamical systems presented in this section, the reader is referred to Altin et al. (2018) and Altin and Sanfelice (2019). The case when the hybrid dynamics are discretized is treated in Ojaghi et al. (2019). It should be noted that when the flow and jump sets overlap, the OCP may have multiple optimal inputs. Nonuniqueness of optimal inputs already appears in nonlinear MPC (see, e.g., in Grimm et al. 2007; Rawlings and Mayne 2009; Borrelli et al. 2017; Mayne and Michalska 1990; Chen and Allgöwer 1997; Magni and Scattolini 2004),

but the source of nonuniqueness of solutions for the case of hybrid dynamical systems is conceptually different. For these systems, conditions guaranteeing that the value function, which at every point x_0 is given by $\mathcal{J}^*(x_0) := \mathcal{J}(x_*, u_*)$ with (x_*, u_*) being minimizers of $\mathcal{J}$ from x_0 , implies an asymptotic stability property for the plant are given in Altin and Sanfelice (2019). The same reference also highlights key properties of the feasibility set, recursive feasibility, monotonicity properties of the cost functional, and regularity properties of the value function.

Cross-References

- [Model Predictive Control for Power Networks](#)
- [Model Predictive Control in Practice](#)
- [Modeling Hybrid Systems](#)
- [Robust Model Predictive Control](#)

- [Simulation of Hybrid Dynamic Systems](#)
- [Stability Theory for Hybrid Dynamical Systems](#)

Bibliography

- Altin B, Sanfelice RG (2019) Asymptotically stabilizing model predictive control for hybrid dynamical systems. In: Proceedings of the American Control Conference
- Altin B, Ojaghi P, Sanfelice RG (2018) A model predictive control framework for hybrid systems. In: NMPC, vol 51, pp 128–133, Aug 2018
- Bemporad A, Borrelli F, Morari M (2000a) Optimal controllers for hybrid systems: stability and piecewise linear explicit form. In: Proceedings of the 39th IEEE conference on decision and control, vol 2. IEEE, pp 1810–1815
- Bemporad A, Borrelli F, Morari M (2000b) Piecewise linear optimal controllers for hybrid systems. In: Proceedings of the 2000 American control conference, vol 2, pp 1190–1194. IEEE
- Bemporad A, Heemels WMHP, De Schutter B (2002) On hybrid systems and closed-loop MPC systems. IEEE Trans Autom Control 47(5):863–869
- Borrelli F, Bemporad A, Morari M (2017) Predictive control for linear and hybrid systems. Cambridge University Press, Cambridge/New York
- Chen H, Allgöwer F (1997) A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. In: European control conference. IEEE, pp 1421–1426
- Chen H, Allgöwer F (1998) A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica 34(10):1205–1217
- Grimm G, Messina MJ, Tuna SE, Teel AR (2007) Nominally robust model predictive control with state constraints. IEEE Trans Autom Control 52(10):1856–1870
- Karaman S, Sanfelice RG, Frazzoli E (2008) Optimal control of mixed logical dynamical systems with linear temporal logic specifications. In: Proceedings of 47th IEEE conference on decision and control, NULL, pp 2117–2122
- Lazar M, Heemels WMPH, Bemporad A (2006) Stabilizing model predictive control of hybrid systems. IEEE Trans. Autom. Control 51(11):1813–1818
- Magni L, Scattolini R (2004) Model predictive control of continuous-time nonlinear systems with piecewise constant control. IEEE Trans. Autom. Control 49(6):900–906
- Mayne DQ, Michalska H (1990) Receding horizon control of nonlinear systems. IEEE Trans. Autom. Control 35(7):814–824
- Nešić D, Grüne L (2006) A receding horizon control approach to sampled-data implementation of continuous-time controllers. Syst. Control Lett. 55(8):660–672
- Ojaghi P, Altin B, Sanfelice RG (2019) A model predictive control framework for asymptotic stabilization of

- discretized hybrid dynamical systems. In: To appear at the IEEE conference on decision and control
- Rawlings JB, Mayne DQ (2009) Model predictive control: theory and design. Nob Hill Pub., Madison
- Sanfelice RG (2018) Hybrid predictive control, 1 edn. Birkhäuser, Basel, pp 199–220
- Sopasakis P, Patrinos P, Sarimveis H, Bemporad A (2015) Model predictive control for linear impulsive systems. IEEE Trans Autom Control 60(8):2277–2282

Hybrid Observers

Daniele Carnevale

Dipartimento di Ing. Civile ed Ing. Informatica,
Università di Roma “Tor Vergata”, Roma, Italy

H

Abstract

In first part two hybrid observer designs for non-hybrid systems are presented. In the second part, recently results available in the literature related to the observability and observer design for different classes of hybrid systems are introduced.

Keywords

Hybrid systems · Observer design ·
Observability · Switching systems

Introduction

Observers' design, which are used to estimate the unmeasured plant state, has received a lot of attention since the late 1960s. One of the first leading contributions to clearly formalize the estimation problem and propose a solution in the linear case has been introduced by Luenberger (1966). The recipe to implement a Luenberger-type observer for linear time-invariant (LTI) continuous-time systems described by

$$\dot{x} = Ax + Bu, \quad y = Cx + Du, \quad (1)$$

with $x \in \mathbb{R}^n, u \in \mathbb{R}^p, y \in \mathbb{R}^m, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times p}, C \in \mathbb{R}^{m \times n}$, and $D \in \mathbb{R}^{m \times p}$ has three main ingredients: system data (A, B, C, D) , the correction term commonly referred to as *output injection*, and the *observability/detectability/determinability* conditions. A Luenberger-type observer for (1) is then a copy of the system dynamics with the output injection term $\mathbf{L}(y - \hat{y})$ as

$$\dot{\hat{x}} = A\hat{x} + Bu + \mathbf{L}(y - \hat{y}), \quad \hat{y} = C\hat{x} + Du, \quad (2)$$

with $\mathbf{L} \in \mathbb{R}^{n \times m}$ and where $\hat{x}$ is the estimated value of x . Certainly if $\hat{x}(0) = x(0)$ and the input to the plant $u(t)$ is known, then $\hat{x}(t) = x(t)$ for all $t \geq 0$. For all nontrivial cases, when $x(0) \neq \hat{x}(0)$, the estimation error defined as $e = x - \hat{x}$ satisfies the differential equation $\dot{e} = (A - LC)e$, with initial condition $e(0) = x(0) - \hat{x}(0)$. The time evolution of the estimation error $e(t)$ depends only on the properties of the matrix $A + LC$, i.e., if the pair (A, C) is observable (equivalent to determinability for continuous-time systems), there exists a unique $\mathbf{L}$ to assign arbitrarily the eigenvalues of $A + LC$ with negative real part to render the origin $e = 0$ of the estimation error system globally asymptotically stable (GAS). If the pair (A, C) is detectable, there exists $\mathbf{L}$ such that all the eigenvalues of $A + LC$ still have negative real parts ensuring that the origin $e = 0$ is GAS, although all the eigenvalues cannot be arbitrarily chosen. The observer (2) exploits only the injection term $\mathbf{L}$ in continuous time to render Hurwitz the matrix $A + LC$, but one may ask how profitable could be to allow jumps of the observer state designing an observer that has both continuous (flow map) and discrete (jump map) time dynamics, i.e., an *hybrid observer*. On the other way around, there are systems that are intrinsically described by hybrid dynamics and for which it seems natural to directly provide hybrid observers. Designs of hybrid observer is a relatively new area of research, and results are consolidated only for few classes of systems.

In section “[Continuous-Time Plants](#),” a hybrid redesign of the observer (2) is discussed first, and then a more general design for nonlinear systems

is introduced, whereas in section “[Systems with Flows and Jumps](#),” the recent results related to observability and observer designs for hybrid systems are discussed. Conclusions are given in section “[Summary and Future Directions](#).”

Hybrid Observers: Different Strategies

The community of researchers working on hybrid observer, which is a quite recent area and is the subject of growing interest, is wide, and a unique formal definition/notation has not been reached yet. This fact is strictly related to the large number of different hybrid system models that are currently adopted by researchers. To render as simple as possible this short presentation, we let the state $x(t)$ of a hybrid system be driven by the flow map (differential equation) when $t \neq t_j$ and by the jump map (difference equation) when $t = t_j$ is the jump time and $x(t)$ is right continuous, i.e., $\lim_{t \rightarrow t_j^+} x(t) = x(t_j)$.

Continuous-Time Plants

Linear Case

A simple strategy to improve convergence to zero of the estimation error for (1) has been proposed in Raff and Allgower (2008) and consists in resetting the observer state $\hat{x}$, at predetermined fixed jump times t_j , by means of the linear correction term $\mathbf{K}(t)(y(t) - C\hat{x}(t))$ as

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + \mathbf{L}(y(t) - C\hat{x}(t)), \quad (3a)$$

$$\hat{x}(t_j) = \hat{x}(t_j^-) + \mathbf{K}(t_j^-)(y(t_j^-) - C\hat{x}(t_j^-)), \quad (3b)$$

where $t_0 = 0$, $t_{j+1} - t_j = T > 0$, $j \in \mathbb{N}_{\geq 1}$ and T is a parameter that defines the interval times between resets and has to be chosen such that

$$\text{Im}(\lambda_p - \lambda_r)T \neq 2r\pi, \quad r \in \mathbb{Z} \setminus \{0\}, \quad (4)$$

for each pair (λ_p, λ_r) of complex eigenvalues of the matrix $A - LC$. This selection preserves the (continuous time or flow) observability of the system (1) when sampled at time instants t_j . Then, the estimation error $e(t)$ converges to zero in finite time, $t = nT$, if (1) is observable and the matrix $K(t) : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n \times m}$ is selected such as $K(t) = K_0$ if $t \leq t_n$ and $K(t) = 0$ otherwise where K_0 is such that $(I - K_0 C) \exp((A - LC)T)$ has all its eigenvalues at zero. It is important to note that the state reset (3b) yields a hybrid estimation error system given by

$$\dot{e}(t) = (A - LC)e(t), \quad (5a)$$

$$e(t_j) = (I - K(t_j^-)C)e(t_j^-). \quad (5b)$$

The stability property of the origin can be easily deduced by noting that

$$e(t_j) = \prod_{k=1}^j (I - K(t_k^-)C) \exp((A - LC)T)e(0),$$

and given that $(I - K_0 C) \exp((A - LC)T)$ is nilpotent, then $e(t_n) = e(nT) = 0$.

Along the same line, to improve convergence performances of continuous-time observer, in Prieur et al. (2012), it is proposed a methodology to limit the well-known *peaking phenomena* affecting the *high-gain observers* (Tornambé 1992; Khalil and Praly 2013) opportunely resetting its (augmented) state. Moreover, when the output of (1) is a nonlinear function of the state, $y = h(x)$, with $h(\cdot)$ not invertible (e.g., the saturation function), it would be possible to rewrite (1) as a hybrid system with linear flow map and augmented state designing a hybrid observer as in Carnevale and Astolfi (2009). In Possieri and Teel (2016), an introduction to structural properties for estimation and output feedback stabilization of linear hybrid systems is given.

Nonlinear Case

When the input of a continuous-time plant is piecewise-constant, the hybrid observer proposed in Moraal and Grizzle (1995), exploiting sampled measurements, can be successfully applied for a class of nonlinear continuous (or discrete-time) systems

$$\dot{x} = f(x(t), u(t)), \quad y(t) = h(x(t), u(t)), \quad (6)$$

with sufficiently smooth maps $f(\cdot, \cdot)$ and $h(\cdot, \cdot)$ and where

$$x(t_j) = F(x(t_{j-1}), u(t_{j-1})), \quad (7)$$

is the sample data (discrete-time) version of (6) with sampling time $T = t_{j-1} - t_j$. Then, it is possible to define a hybrid observer of the following type:

$$\dot{\hat{x}}(t) = f(\hat{x}(t), u(t)), \quad (8a)$$

$$\hat{x}(t_j) = \Gamma(y(t_j^-), \hat{x}(t_j^-), \xi(t_j^-)), \quad (8b)$$

where the reset map Γ and the dynamics of the new variable $\xi(t)$ have to be properly defined. The main idea in Moraal and Grizzle (1995) is that the Newton method, in continuous and discrete-time, can be used to estimate the value of ξ that renders zero the function

$$W_j^N(\xi) = Y_j^N - H(\xi, U_j^N), \quad (9)$$

where $U_j^N = [u'(t_{j-N+1}), \dots, u'(t_j)]'$ and $Y_j^N = [y'(t_{j-N+1}), \dots, y'(t_j)]'$ are the sampled input and output vectors, respectively, and $H : \mathbb{R}^n \times \mathbb{R}^{m \times N} \rightarrow \mathbb{R}^N$ maps the state $x(t_j)$ and the N-tuple of control inputs U_j^N into the output vector Y_j^N , i.e., $H(x(t_j), U_j^N) = Y_j^N$, and is defined as

$$H(x, U_j^N) \triangleq \begin{bmatrix} h(F^{-1}(F^{-1}(\dots, u(t_{j-N+1})), u(t_{j-N+1}))) \\ \vdots \\ h(F^{-1}(x, u(t_{j-1})), u(t_{j-1})) \\ h(x, u(t_j)) \end{bmatrix}, \quad (10)$$

where F^{-1} shortly represents the inverse of the map F such that $x(t_{j-1}) = F^{-1}(x(t_j), u(t_{j-1}))$.

The system (6) and (7) is said to be *N-observable*, for some $N \geq 1$ (the generic selection is $N = 2n + 1$), when $W_j^N(\xi) = 0$ hold only if $\xi = x(t_j)$, uniformly in U_j^N . Then, under certain technical assumptions (see Moraal and Grizzle 1995) related to the derivatives of f and h and the invertibility of the Jacobian matrix $J(x) = \partial H(x)/\partial x$, it is possible to select

$$\dot{\xi}(t) = kJ(\xi(t))^{-1} (Y_j^N - H(\xi(t), U_j^N)), \quad (11a)$$

$$\xi(t_j) = F(\xi(t_j^-), u(t_{j-1})), \quad (11b)$$

with a sufficiently high-gain $k > 0$ and the reset map $\Gamma(\cdot) = F(\xi(t_j^-), u(t_{j-1}))$. Note that (11a) is commonly referred to as Newton flow. This approach could be easily extended to other continuous-time minimization algorithms (normalized gradient, line search, etc.) changing the rhs of (11a) or even with discrete-time methods iterated at higher frequency within the sample time T , yielding faster convergence to zero of the estimation error.

The same approach can be used when a continuous-time observer for (6) is considered in place of (8a) and the Newton-based resets can be used to possibly improve the performances. The continuous and discrete-time Newton algorithm require the knowledge of the jump map F to define (7), i.e., the exact discrete-time model of (6), and the Jacobian matrix $J(x) = \partial H(x)/\partial x$. An approach that does not require such knowledge is proposed in Biyik and Arcak (2006), where continuous-time filters and secant method allow to estimate (numerically) the map F and the Jacobian matrix, or in Sassano et al. (2011) where an *extremum-seeking*-based technique is considered.

A different approach to estimate the state of a continuous-time plant, Pursued, for example, in Ahrens and Khalil (2009) and Liu (1997), exploits switching output injections, letting the correction term $l_\sigma(\cdot)$ to switch among opportune values selected by a suitable definition (often derived by a Lyapunov-based proof) of the switching signal $\sigma(t)$. These switching gains allow to improve observer performances and robustness against measurement noise and model uncertainties.

Systems with Flows and Jumps

The classical notion of observability does not hold for hybrid systems. As an example, consider the autonomous linear hybrid system described by $\dot{x}(t) = Ax(t)$ and $x(t_j) = Jx(t_j^-)$ with

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad J = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad (12)$$

and $C = [0, 1, 0]$. Evidently the flow is not observable in the classic sense given that $\mathcal{O}_{\text{flow}} = [C', (CA)', (CA^2)']'$ is not full rank and the flow-unobservable subspace is $\ker(\mathcal{O}_{\text{flow}}) \triangleq \{x \in \mathbb{R}^3 : x_2 = x_3 = 0\}$. Nevertheless, in the first flow time interval $\tau = t_1 - t_0$, it is possible to estimate (e.g., in finite time using the observability Gramian matrix) the initial conditions $(x_2(t_0), x_3(t_0))$. Then when the first jump take place at time t_1 , thanks to the structure of the jump map J that resets the value of $x_3(t_1)$ with the flow-unobservable $x_1(t_1^-)$, it is possible to estimate in the next flow time interval the value of $x_1(t_1^-)$ so that the initial condition $x(t_0)$ can be completely determined. The hybrid observability matrix in this case has the following expression

$$\mathcal{O}_{\text{hybrid}} = [\mathcal{O}'_{\text{flow}}, (\mathcal{O}_{\text{flow}} J e^{AT_1})', (\mathcal{O}_{\text{flow}} (J e^{AT_2})^2)']'$$

and is full rank for all $T_j = t_j - t_{j-1}$ satisfying (4). Note that from a practical point of view, in this case the time interval that allows to reconstruct the complete state is $[t_0, t_1 + \epsilon)$ since the observer needs at least an ϵ time of the new measurements (after the first jump) to evaluate the full state $[\mathcal{O}'_{\text{flow}}, (\mathcal{O}_{\text{flow}} J e^{AT_1})', (\mathcal{O}_{\text{flow}} (J e^{AT_2})^2)']'$. This simple example suggests that (impulsive) hybrid systems might have a reacher notion of observability than the classical ones. These properties have been studied also for mechanical systems subject to non-smooth impacts in Martinelli et al. (2004), where a high-gain-like observer design has been proposed assuming the knowledge of the impact times t_i , no *Zeno* phenomena (no finite accumulation point for t_j 's), and a minimum *dwell-time*, $t_{j+1} - t_j \geq \delta > 0$. With the aforementioned assumptions and considering the more general class of hybrid system described by

$$\begin{aligned}\dot{x}(t) &= f(x, u), \\ x(t_j) &= g(x(t_j^-), u(t_j^-)),\end{aligned}\quad (13)$$

with $y = h(x, u)$, a frequent choice is to consider the hybrid observer of the form

$$\dot{\hat{x}}(t) = f(\hat{x}, u) + \mathbf{l}(y, \mathbf{x}, \mathbf{u}), \quad (14a)$$

$$\hat{x}(t_j) = g(\hat{x}(t_j^-), u(t_j^-)) + \mathbf{m}(\hat{\mathbf{x}}(t_j^-), \mathbf{u}(t_j^-)), \quad (14b)$$

with $\mathbf{l}(\cdot)$ and $\mathbf{m}(\cdot)$ that are zero when $\hat{x} = x$ rendering flow and jump-invariant the manifold $\hat{x} = x$ relying only on the correction term $\mathbf{l}(\cdot)$ ($\mathbf{m} \equiv 0$) in a high-gain-like design during the flow. The correction during the flow has to recover, within the minimum dwell-time δ , the worst deterioration of the estimation error induced by the jumps (if any) and the transients such that $\|e(t_{j+1})\| < \|e(t_j^-)\|$ or $V(e(t_{j+1})) < V(e(t_j^-))$ if a Lyapunov function is available. This type of observer design, with $m = 0$ and the linear choice $\mathbf{l}(y, \hat{x}, u) = L(y - M\hat{x})$, has been proposed in

Heemels et al. (2011) for *linear complementarity systems* (LCS) in the presence of state jumps induced by impulsive input. Therein, solutions of LCS are characterized by means of piecewise Bohl distributions, and the specially defined *well-posedness* and *low-index* properties, which combined with passivity-based arguments, allow to design a global hybrid observer with exponential convergence. A separation principle to design an output feedback controller is also proposed.

An interesting approach is pursued in Forni et al. (2003) where global output tracking results on a class of linear hybrid systems subject to impacts is introduced. Therein, the key ingredient is the definition of a “mirrored” tracking reference (a change of coordinate) that depends on the sequence of different jumps between the desired trajectory (a virtual bouncing ball) and the plant (the controlled ball). Exploiting this (time-varying) change of coordinates and assuming that the impact times are known, it is possible to define an estimation error that is not discontinuous even when the tracked ball has a bounce (state jump) and the plant does not. A time regularization is included in the model embedding a minimum dwell-time among jumps. In this way, it is possible to design a linear hybrid observer represented by (14b) with a linear (mirrored) term $\mathbf{l}(\cdot)$ and $\mathbf{m}(\cdot) \equiv 0$, proving (by standard quadratic Lyapunov functions) that the origin of the estimation error system is GES. In this case, the standard observability condition for the couple (A, C) is required.

Switching Systems and Hybrid Automata

Switching systems and hybrid automata have been the subject of intense study of many researchers in the last two decades. For these class of systems, there is a neat separation $x = [z, q]'$ among purely discrete-time state q (*switching signal or system mode*) and rest of the state z that generically can both flow and jump. The observability of the entire system is often divided into the problem of determining the switching signal q first and then z . The switching signal can be divided into

two categories: arbitrary (*universal problem*) or specific (*existential problems*) switchings.

In Vidal et al. (2003), the observability of autonomous linear switched systems with no state jump, minimum dwell-time, and unknown switching signal is analyzed. Necessary and sufficient observability conditions based on rank tests and output discontinuities detection strategies are given. Along the same line, the results are extended in Babaali and Pappas (2005) to nonautonomous switched systems with non-Zeno solutions and without the minimum dwell-time requirement, providing state z and mode q observability characterized by linear-algebraic conditions.

Luenberger-type observers with two distinct gain matrices L_1 and L_2 are proposed in the case of bimodal piecewise linear systems in Juloski et al. (2007) (where state jumps are considered), whereas recently in Tanwani et al. (2013), algebraic observability conditions and observer design are proposed for switched linear systems admitting state jumps with known switching signal (although some asynchronism between the observer and the plant switches is allowed). Results related to the observability of hybrid automata, which include switching systems, can be found in Balluchi et al. (2002) and the related references. Therein the *location observer* estimates first the system *current location* q , processing system input and output assuming that it is *current-location observable*, a property that is related to the system *current-location observation tree*. This graph is iteratively explored at each new input to determine the node associated to the current value of $q(t)$. Then, a linear (switched) Luenberger-type observer for the estimation of the state z , assuming minimum dwell-time and observability of each pair (A_q, C_q) , is proposed. A local separation principle have been proposed in Teel (2010).

Summary and Future Directions

Observer design and observability properties of general hybrid systems is an active field of research, and a number of different results have

been proposed by researchers. Results are based on different notations and definitions for hybrid systems. Efforts to provide a unified approach, in many cases considering the general framework for hybrid systems proposed in Goebel et al. (2009), are pursued by the scientific community to improve consistency and cohesion of the general results. Observer designs, observability properties, and separation principle are still open challenges for the scientific community.

Cross-References

- [Hybrid Dynamical Systems, Feedback Control of](#)
- [Observer-Based Control](#)
- [Observers for Nonlinear Systems](#)
- [Observers in Linear Systems Theory](#)

Bibliography

- Ahrens JH, Khalil HK (2009) High-gain observers in the presence of measurement noise: a switched-gain approach. *Automatica* 45(5):936–943
- Babaali M, Pappas GJ (2005) Observability of switched linear systems in continuous time. In: Morari M, Thiele L (eds) *Hybrid systems: computation and control*. Volume 3414 of lecture notes in computer science. Springer, Berlin/Heidelberg, pp 103–117
- Balluchi A, Benvenuti L, Benedetto MDD, Vincentelli ALS (2002) Design of observers for hybrid systems. In: *Hybrid systems: computation and control*, vol 2289. Springer, Stanford
- Biyik E, Arcak M (2006) A hybrid redesign of Newton observers in the absence of an exact discrete-time model. *Syst Control Lett* 55(8):429–436
- Branicky MS (1998) Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans Autom Control* 43(5):475–482
- Carnevale D, Astolfi A (2009) Hybrid observer for global frequency estimation of saturated signals. *IEEE Trans Autom Control* 54(13):2461–2464
- Forni F, Teel A, Zaccarian L (2003) Follow the bouncing ball: global results on tracking and state estimation with impacts. *IEEE Trans Autom Control* 58(8):1470–1485
- Goebel R, Sanfelice R, Teel AR (2009) Hybrid dynamical systems. *IEEE Control Syst Mag* 29:28–93
- Heemels WPMH, Camlibel MK, Schumacher J, Brogliato B (2011) Observer-based control of linear complementarity systems. *Int J Robust Nonlinear Control* 21(13):1193–1218. Special issues on hybrid systems

- Juloski AL, Heemels WPMH, Weiland S (2007) Observer design for a class of piecewise linear systems. *Int J Robust Nonlinear Control* 17(15):1387–1404
- Khalil HK, Praly L (2013) High-gain observers in nonlinear feedback control. *Int J Robust Nonlinear Control* 24:993–1015
- Liu Y (1997) Switching observer design for uncertain nonlinear systems. *IEEE Trans Autom Control* 42(12):1699–1703
- Luenberger DG (1966) Observers for multivariable systems. *IEEE Trans Autom Control* 11: 190–197
- Martinelli F, Menini L, Tornambè A (2004) Observability, reconstructibility and observer design for linear mechanical systems unobservable in absence of impacts. *J Dyn Syst Meas Control* 125:549
- Moraal P, Grizzle J (1995) Observer design for nonlinear systems with discrete-time measurements. *IEEE Trans Autom Control* 40(3):395–404
- Possieri C, Teel A (2016) Structural properties of a class of linear hybrid systems and output feedback stabilization. *IEEE Trans Autom Control* 62(6):2704–2719
- Prieur C, Tarbouriech S, Zaccarian L (2012) Hybrid high-gain observers without peaking for planar nonlinear systems. In: 2012 IEEE 51st annual conference on decision and control (CDC), Maui, pp 6175–6180
- Raff T, Allgower F (2008) An observer that converges in finite time due to measurement-based state updates. In: Proceedings of the 17th IFAC world congress, COEX, South Korea, vol 17, pp 2693–2695
- Sassano M, Carnevale D, Astolfi A (2011) Extremum seeking-like observer for nonlinear systems. In: 18th IFAC world congress, Milano, vol 18, pp 1849–1854
- Tanwani A, Shim H, Liberzon D (2013) Observability for switched linear systems: characterization and observer design. *IEEE Trans Autom Control* 58(5): 891–904
- Teel A (2010) Observer-based hybrid feedback: a local separation principle. In: American control conference (ACC), 2010, Baltimore, pp 898–903
- Vidal R, Chiuso A, Soatto S, Sastry S (2003) Observability of linear hybrid systems. In: Maler O, Pnueli A (eds) Hybrid systems: computation and control. Volume 2623 of lecture notes in computer science. Springer, Berlin/Heidelberg, pp 526–539
- Tornambè A (1992) High-gain observers for nonlinear systems. *International Journal of Systems Science* 23(9): 1475–1489

Identification and Control of Cell Populations

Mustafa Khammash¹ and J. Lygeros²

¹Department of Biosystems Science and Engineering, Swiss Federal Institute of Technology at Zurich (ETHZ), Basel, Switzerland

²Automatic Control Laboratory, Swiss Federal Institute of Technology Zurich (ETHZ), Zurich, Switzerland

Abstract

We explore the problem of identification and control of living cell populations. We describe how de novo control systems can be interfaced with living cells and used to control their behavior. Using computer controlled light pulses in combination with a genetically encoded light-responsive module and a flow cytometer, we demonstrate how in silico feedback control can be configured to achieve precise and robust set point regulation of gene expression. We also outline how external control inputs can be used in experimental design to improve our understanding of the underlying biochemical processes.

Keywords

Extrinsic variability · Heterogeneous populations · Identification · Intrinsic

variability · Population control · Stochastic biochemical reactions

Introduction

Control systems, particularly those that employ feedback strategies, have been used successfully in engineered systems for centuries. But natural feedback circuits evolved in living organisms much earlier, as they were needed for regulating the internal milieu of the early cells. Owing to modern genetic methods, engineered feedback control systems can now be used to control in real-time biological systems, much like they control any other process. The challenges of controlling living organisms are unique. To be successful, suitable sensors must be used to measure the output of a single cell (or a sample of cells in a population), actuators are needed to affect control action at the cellular level, and a controller that connects the two should be suitably designed. As a model-based approach is needed for effective control, methods for identification of models of cellular dynamics are also needed. In this entry, we give a brief overview of the problem of identification and control of living cells. We discuss the dynamic model that can be used, as well as the practical aspects of selecting sensor and actuators. The control systems can either be realized on a computer (in silico feedback) or through genetic manipulations (in vivo feedback). As an example, we describe how de novo control systems can be interfaced with living cells and used to control

their behavior. Using computer controlled light pulses in combination with a genetically encoded light-responsive module and a flow cytometer, we demonstrate how *in silico* feedback control can be configured to achieve precise and robust set point regulation of gene expression.

Dynamical Models of Cell Populations

In this entry, we focus on a model of an essential biological process: gene expression. The goal is to come up with a mathematical model for gene expression that can be used for model-based control. Due to cell variability, we will work with a model that describes the average concentration of the product of gene expression (the regulated variable). This allows us to use population measurements and treat them as measurements of the regulated variable. We refer the reader to the entry ► [Stochastic Description of Biochemical Networks](#) in this encyclopedia for more information on stochastic models of biochemical reaction networks. In this framework, the model consist of an N -vector stochastic process $X(t)$ describing the number of molecules of each chemical species of interest in a cell. Given the chemical reactions in which these species are involve, the mean, $E[X(t)]$, of $X(t)$ evolves according to deterministic equations described by

$$\dot{E}[X(t)] = SE[w(X(t))],$$

where S is an $N \times M$ matrix that describes the stoichiometry of the M reactions described in the model, while $w(\cdot)$ is an M -vector of propensity functions. The propensity functions reflect the rate of the reactions being modeled. When one considers elementary reactions (see ► [Stochastic Description of Biochemical Networks](#)), the propensity function of the i th reaction, $w_i(\cdot)$, is a quadratic function of the form $w_i(x) = a_i + b_i^T x + c_i x^T Q_i x$. Typically, w_i is either a constant: $w_i(x) = a$, a linear function of the form $w_i(x) = bx_j$ or a simple quadratic of the form $w_i(x) = cx_j^2$. Following the same procedure, similar dynamical models can be derived that describe the evolution of higher-order moments

(variances, covariances, third-order moments, etc.) of the stochastic process $X(t)$.

Identification of Cell Population Models

The model structure outlined above captures the fundamental information about the chemical reactions of interest. The model parameters that enter the functions $w_i(x)$ reflect the reaction rates, which are typically unknown. Moreover, these reaction rates often vary between different cells, because, for example, they depend on the local cell environment, or on unmodeled chemical species whose numbers differ from cell to cell (Swain et al. 2002). The combination of this extrinsic parameter variability with the intrinsic uncertainty of the stochastic process $X(t)$ makes the identification of the values of these parameters especially challenging.

To address this combination of intrinsic and extrinsic variabilities, one can compute the moments of the stochastic process $X(t)$ together with the cross moments of $X(t)$ and the extrinsic variability. In the process, the moments of the parametric uncertainty themselves become parameters of the extended system of ordinary differential equations and can, in principle, be identified from data. Even though doing so requires solving a challenging optimization problem, effective results can often be obtained by randomized optimization methods. For example, Zechner et al. (2012) presents the successful application of this approach to a complex model of the system regulating osmotic stress response in yeast.

When external signals are available, or when one would like to determine what species to measure when, such moment-based methods can also be used in experiment design. The aim here is to determine *a priori* which perturbation signals and which measurements will maximize the information on the underlying chemical process that can be extracted from experimental data, reducing the risk of conducting expensive but uninformative experiments. One can show that, given a tentative model for the biochemical process, the moments of the stochastic process $X(t)$ (and

cross $X(t)$ -parameter moments in the presence of extrinsic variability) can be used to approximate the Fischer information matrix and hence characterize the information that particular experiments contain about the model parameters; an approximation of the Fischer information based on the first two moments was derived in Komorowski et al. (2011) and an improved estimate using correction terms based on moments up to order 4 was derived in Ruess et al. (2013). Once an estimate of the Fischer information matrix is available, one can design experiments to maximize the information gained about the parameters of the model. The resulting optimization problem (over an appropriate parametrization of the space of possible experiments) is again challenging but can be approached by randomized optimization methods.

Control of Cell Populations

There are two control strategies that one can implement. The control systems can be realized on a computer, using real-time measurements from the cell population to be controlled. These cells must be equipped with actuators that respond to the computer signals that close the feedback loop. We will refer to this as *in silico feedback*. Alternatively, one can implement the sensors, actuators, and control system in the entirety within the machinery of the living cells. At least in principle, this can be achieved through genetic manipulation techniques that are common in synthetic biology. We shall refer to this type of control as *in vivo feedback*. Of course some combination of the two strategies can be envisioned. In vivo feedback is generally more difficult to implement, as it involves working within the noisy uncertain environment of the cell and requires implementations that are biochemical in nature. Such controllers will work autonomously and are heritable, which could prove advantageous in some applications. Moreover, coupled with intercellular signaling mechanisms such as quorum sensing, in vivo feedback may lead to tighter regulation (e.g., reduced variance) of the cell population. On the other hand, in silico controllers are much easier to program, debug,

and implement and can have much more complex dynamics that would be possible with in vivo controllers. However, in silico controllers require a setup that maintains contact with all the cells to be controlled and cannot independently control large numbers of such cells. In this entry we focus exclusively on in silico controllers.

The Actuator

There could be several ways to send actuating signals into living cells. One consists of chemical inducers that the cells respond to either through receptors outside the cell or through translocation of the inducer molecules across the cellular membrane. The chemical signal captured by these inducers is then transduced to affect gene expression. Another approach we will describe here is induction through light. There are several light systems that can be used. One of these includes a light-sensitive protein called phytochrome B (PhyB). When red light of wavelength 650 nm is shined on PhyB in the presence of phyco-cyanobilin (PCB) chromophore, it is activated. In this active state it binds to another protein Pif3 with high affinity forming PhyB-Pif3 complex. If then a far-red light (730 nm) is shined, PhyB is deactivated and it dissociates from Pif3. This can be exploited for controlling gene expression as follows: PhyB is fused to a GAL4 binding domain (GAL4BD), which then binds to DNA in a specific site just upstream of the gene of interest. Pif3 in turn is fused to a GAL4 activating domain (GAL4AD). Upon red light induction, Pif3-Gal4AD complex is recruited to PhyB, where Gal4AD acts as a transcription factor to initiate gene expression. After far-red light is shined, the dissociation of GAL4BD-PhyB complex with Pif3-Gal4AD means that Gal4AD no longer activates gene expression, and the gene is off. This way, one can control gene expression – at least in open loop.

The Sensor

To measure the output protein concentration in cell populations, a fluorescent protein tag is needed. This tag can be fused to the protein of interest, and the fluorescence intensity emanating from each cell is a direct measure of the protein concentration in that cell. There are several

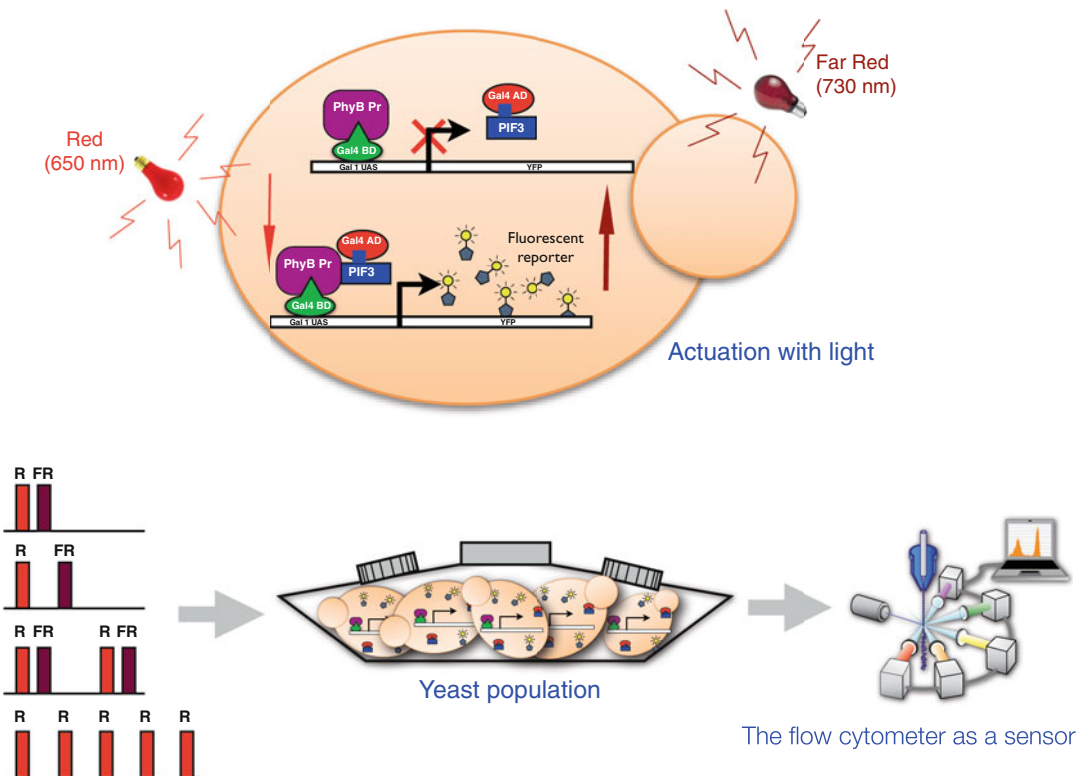
technologies for measuring fluorescence of cell populations. While fluorimeters measure the overall intensity of a population, flow cytometry and microscopy can measure the fluorescence of each individual cell in a population sample at a given time. This provides a snapshot measurement of the probability density function of the protein across the population. Repeated measurements over time can be used as a basis for model identification (Fig. 1).

The Control System

Equipped with sensors, actuators, and a model identified with the methods outlined above one can proceed to design control algorithms to regulate the behavior of living cells. Even though moment equations lead to models that look like conventional ordinary differential

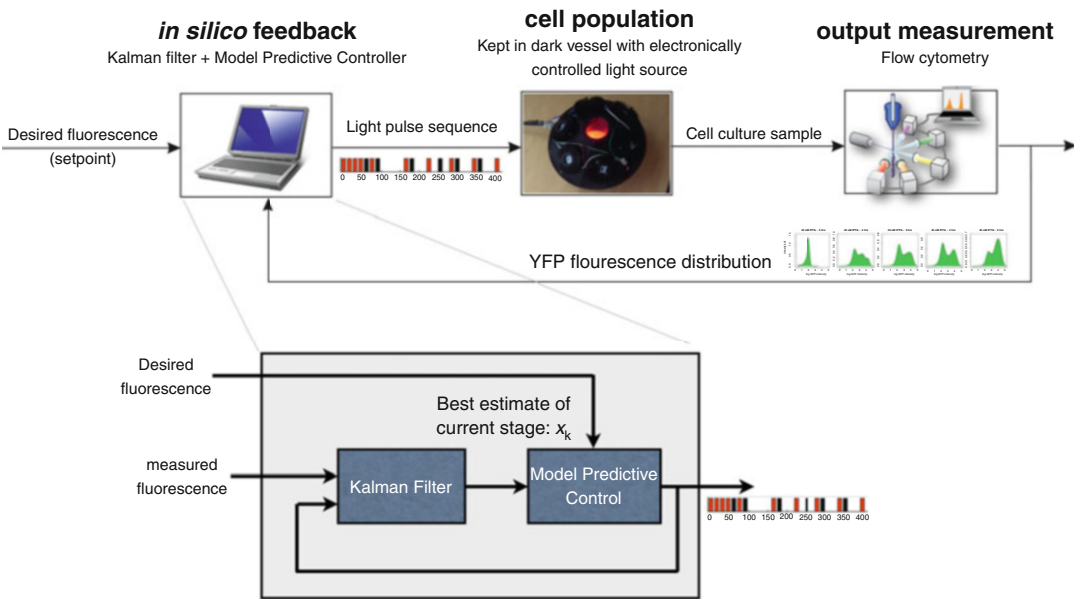
equations, from a control theory point of view, cell population systems offer a number of challenges. Biochemical processes, especially genetic regulation, are often very slow with time constants of the order of tens of minutes. This suggests that pure feedback control without some form of preview may be insufficient. Moreover, due to our incomplete understanding of the underlying biology, the available models are typically inaccurate, or even structurally wrong. Finally, the control signals are often unconventional; for example, for the light control system outlined above, experimental limitations imply that the system must be controlled using discrete light pulses, rather than continuous signals.

Fortunately advances in control theory allow one to effectively tackle most of these challenges. The availability of a model, for example, enables the use of model predictive control methods that



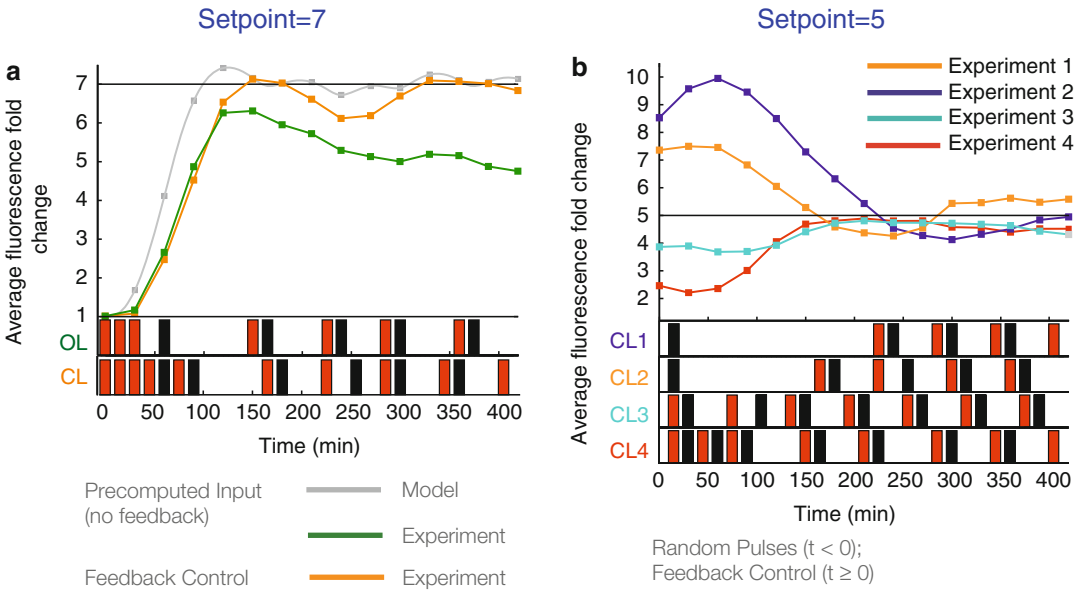
Identification and Control of Cell Populations, Fig. 1 Top figure: shows a yeast cell whose gene expression can be induced by light: red light turns on gene expression while far-red turns it off. Bottom figure: Each input light sequences can be applied to a culture of light responsive

yeast cells resulting in a corresponding gene expression pattern that is measured by flow cytometry. By applying multiple carefully chosen light input test sequences and looking at their corresponding gene expression patterns a dynamic model of gene expression can be identified



Identification and Control of Cell Populations, Fig. 2 Architecture of the closed-loop light control system. Cells are kept darkness until they are exposed to light pulse sequences from the in silico feedback controller. Cell culture samples are passed to the flow cytometer whose

output is fed back to the computer which implements a Kalman filter plus a Model Predictive Controller. The objective of the control is to have the mean gene expression level follow a desired set value



Identification and Control of Cell Populations, Fig. 3 Left panel: The closed-loop control strategy (orange) enables set point tracking, whereas an open-loop strategy (green) does not. Right panel: Four different experiments,

each with a different initial condition. Closed-loop control is turned on at time $t=0$ shows that tracking can be achieved regardless of initial condition. (See Miliadis et al. (2011))

introduce the necessary preview into the feedback process. The presence of unconventional inputs may make the resulting optimization problems difficult, but the slow dynamics work in our favor, providing time to search the space of possible input trajectories. Finally, the fundamental principle of feedback is often enough to deal with inaccurate models. Unlike systems biology applications where the goal is to develop a model that faithfully captures the biology, in population control applications even an inaccurate model is often enough to provide adequate closed-loop performance. Exploring these issues, Miliars-Argeitis et al. (2011) developed a feedback mechanism for genetic regulation using the light control system, based on an extended Kalman filter and a model predictive controller (Figs. 2 and 3). A related approach was taken in Uhlenendorf et al. (2012) to regulate the osmotic stress response in yeast, while Toettcher et al. (2011) develop what is effectively a PI controller for a faster cell signaling system.

Summary and Future Directions

The control of cell populations offers novel challenges and novel vistas for control engineering as well as for systems and synthetic biology. Using external input signals and experiment design methods, one can more effectively probe biological systems to force them to reveal their secrets. Regulating cell populations in a feedback manner opens new possibilities for biotechnology applications, among them the reliable and efficient production of antibiotics and biofuels using bacteria. Beyond biology, the control of populations is bound to find further applications in the control of large-scale, multi-agent systems, including those in transportation, demand response schemes in energy systems, crowd control in emergencies, and education.

Cross-References

- [Deterministic Description of Biochemical Networks](#)

- [Modeling of Dynamic Systems from First Principles](#)
- [Stochastic Description of Biochemical Networks](#)
- [System Identification: An Overview](#)

Bibliography

- Komorowski M, Costa M, Rand D, Stumpf M (2011) Sensitivity, robustness, and identifiability in stochastic chemical kinetics models. *Proc Natl Acad Sci* 108(21):8645–8650
- Miliars-Argeitis A, Summers S, Stewart-Ornstein J, Zuleta I, Pincus D, El-Samad H, Khammash M, Lygeros J (2011) In silico feedback for in vivo regulation of a gene expression circuit. *Nat Biotech* 29(12):1114–1116
- Ruess J, Miliars-Argeitis A, Lygeros J (2013) Designing experiments to understand the variability in biochemical reaction networks. *J R Soc Interface* 10: 20130588
- Swain P, Elowitz M, Siggia E (2002) Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proc Natl Acad Sci* 99(20):12795–12800
- Toettcher J, Gong D, Lim W, Weiner O (2011) Light-based feedback for controlling intracellular signaling dynamics. *Nat Methods* 8:837–839
- Uhlenendorf J, Miermont A, Delaveau T, Charvin G, Fages F, Bottani S, Batt G, Hersen P (2012) Long-term model predictive control of gene expression at the population and single-cell levels. *Proc Natl Acad Sci* 109(35):14271–14276
- Zechner C, Ruess J, Krenn P, Pelet S, Peter M, Lygeros J, Koepl H (2012) Moment-based inference predicts bimodality in transient gene expression. *Proc Natl Acad Sci* 109(21):8340–8345

Identification of Network Connected Systems

Michel Verhaegen

Delft Center for Systems and Control, Delft University, Delft, The Netherlands

Abstract

An overview of the identification of a large number of linear time invariant (LTI) dynamical system that are operating within a network is given. Compared to the well-established field of identifying LTI lumped parameter

systems, system identification of network systems has to deal with the usual large-scale dimensions and the topology of the network as well. A unified state space approach is presented where the parametrization of the network topology or the spatial nature of the network and the parametrization of the temporal properties is viewed as a parametrization of the structure of the system matrices in a large-scale state space model. Different parametrizations are reviewed as well as the consequences they may have on the computational challenges for estimating these parameters. Though the field of identification of network systems is in full expansion, this overview may highlight some directions for new developments within this field.

Keywords

Spatial-temporal parametrization ·
Input-output and state space model · Blind
identification

Introduction

The developments in hardware technology such as micro-optical-electro-mechanical (MOEMS) systems, wireless networks, and distributed computational technology contribute to the advancement of large network-connected systems. Examples are platoons of car, 2D arrays of wave-front sensors of measuring the optical disturbance induced by turbulence in ground-based telescopes, flow control, shape control, etc.

Two major classes of system identification approaches have been developed much along the lines of lumped parameter systems but now often inspired by particular applications. The first approach is based on the input-output transfer functions describing the dynamic connections in the network. This approach employs dynamic signal flow diagrams, and typical applications are from biological or computer networks (Gonçalves and Warnick 2008; Gevers et al. 2019). This approach will be indicated as the *signal flow approach*. The

second is based on structured (large) state space models and are, e.g., developed in the scope of designing sparse and/or distributed controller (efficiently) for network-connected systems. The latter is, e.g., done in D'Andrea and Dullerud (2003), Massioni and Verhaegen (2009), and Rice and Verhaegen (2009), and typical application areas are automated highway systems, formation flying, cross-directional control in paper processing, micro-cantilever array control for massive parallel data storage, or lumped parameter approximations of partial differential equations (PDEs); see applications discussed in D'Andrea and Dullerud (2003). Due to its connection with control design, this approach will be indicated by the *control engineering approach*.

For system identification a number of relevant problems are discussed in this overview. First in section “[Two Modeling Paradigms for Network-Connected Systems](#),” the (graphical) modeling that characterizes both approaches is presented. Here we address for the signal flow approach the recent question about network identifiability. Further merits and problems with both approaches are discussed. Then section “[Parametrizing Spatial-Temporal Dynamics](#)” elaborates on the second approach and discusses a number of parametrizations imposing structure on the system matrices of an LTI state space model. Such parametrization often requires knowledge of the network topology, that is, how the local dynamical systems communicate with one another. The section starts with presenting the parametrization with full prior knowledge of the location of non-zero entries in the system matrices. This would allow to treat identification problem of the signal flow approach as well. Then more black box parametrizations for system matrices that are banded or system matrices in so-called decomposable (Massioni and Verhaegen 2009) or multilinear state space models (Rogers and Russell 2013) are discussed. A brief algorithmic overview for estimating the parameters in these large-scale state space models is given in section “[Numerical Methodologies](#)”. Here we mainly present subspace(-like) algorithms that allow to derive structural information like the

size of the blocks in banded matrices from data. As this field is fully emerging, we present in section “[Summary and Future Directions](#)” a few of the many possibilities for further research.

Two Modeling Paradigms for Network-Connected Systems

Graphical Representation

Networks are generally represented using the language of graph theory, with nodes and edges. Two examples of such graphs are illustrated in Fig. 1. The left-hand side (a) represents the format used in control engineering, such as in D’Andrea and Dullerud (2003), while the signal flow diagram as used, e.g., in Gevers et al. (2019), is represented on the right-hand side (b).

For the control engineering approach, the nodes represent the systems in the network, such as S_i for $i = 1, \dots, 3$ in Fig. 1a, and the edges are the signals. With signal flow diagrams, the reverse convention holds, with nodes representing the signals, such as y_i for $i = 1, \dots, 3$ in Fig. 1b, and the edges are the systems. The topology or the network connection is represented by the graph adjacency or graph Laplacian matrix. A slight generalization of this concept is the pattern matrix (Massioni 2015). Such a pattern matrix for the simple example in Fig. 1 determines the topology or the spatial “dynamics” of the networks via the indices

1, 2, 3 in the gray circles. Using this (spatial) numbering, the pattern matrix is defined as a zero matrix except for the ones in the location ij when there is arrow going from node j to node i (not considering the summing elements). For this simple example, the pattern matrix $\mathcal{P}$ equals:

$$\mathcal{P} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

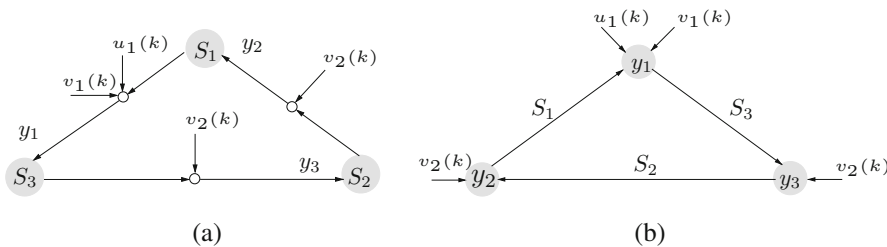
The pattern matrix of an undirected graph with two-sided arrows on the edges is symmetric and allows two-way communication between the nodes.

The Signal Flow Versus the Control Engineering Approach

Though both graphical representations in Fig. 1 equivalently represent the same network of systems, these representations have led to distinct approaches for analyzing large-scale distributed systems.

Let in the signal flow diagram of Fig. 1b the systems be represented by their causal transfer functions $S_i(q)$ (with q the shift operator such that $qx(k) = x(k+1)$); then the *dynamic structure function* (Gonçalves and Warnick 2008) is denoted as:

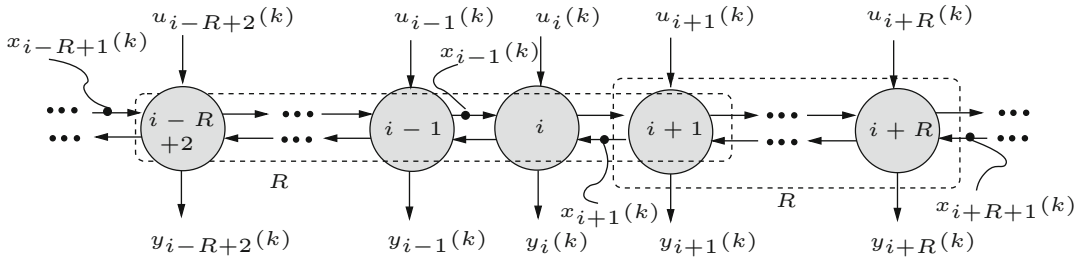
$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{bmatrix} = \begin{bmatrix} 0 & S_1(q) & 0 \\ 0 & 0 & S_2(q) \\ S_3(q) & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{bmatrix}$$



Identification of Network Connected Systems, Fig. 1

Two graphical representations of the same network connection of dynamic systems S_i for $i = 1, \dots, 3$. (a) The control engineering approach with the nodes representing the systems S_i with interconnecting signals depicted as edges. The small circles $\circ$ represent summing points, with

its output the sum of its input signals. (b) Signal flow diagram representation with the nodes representing the signals y_i for $i = 1, \dots, 3$ with interconnecting systems depicted as edges. The incoming signals to the nodes are summed. The arrows in both cases depicting the direction in which data is transferred



Identification of Network Connected Systems, Fig. 2 1D network with resp. R neighboring systems on left and right of the system at location $i+1$ in the network.

The intersystem communication as indicated by the state quantities $x_{\bullet}(k)$ is not observed

$$+ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u_1(k) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} \quad (1)$$

This can compactly be denoted (with the proper definition of corresponding vectors) as:

$$y(k) = G(q)y(k) + K(q)u(k) + w(k) \quad (2)$$

This dynamic structure function is used as the model representation of the signal flow diagrams in Gonçalves and Warnick (2008) and Gevers et al. (2019) and the many references therein. This modeling paradigm turned out to be very useful in addressing *identifiability* questions. Such as under what conditions of the network topology and availability of the node measurements can the causal transfer functions $G(q)$, $K(q)$ and the power spectrum of $w(k)$ be uniquely identified? Such “global” identifiability question is about the *network identifiability* or identifiability of transfer functions and not about structural identifiability of parameters in these transfer functions. A recent overview of network identifiability for the case that *all node variables in the network are measured* with possibly singularity of the power spectrum of $w(k)$ is given in Gevers et al. (2019). When it is possible to (globally) estimate the transfer functions $G(q)$, $K(q)$, it was remarked in Gevers et al. (2019) that then one can find as well the network topology as revealed by the zero pattern in these estimated global transfer functions. These nice results do not extend straightforwardly for the

case some of the node signals are missing. It should be remarked that such problems occur “naturally” in a control engineering analysis of network-connected systems as highlighted, e.g., in D’Andrea and Dullerud (2003). This advocates a state space approach.

In this overview such an approach is the indicated control engineering approach. We restrict its presentation to spatial-temporal dynamical systems with spatial dynamics specified on a regular spatial grid, such as on a line or rectangular. For generalizations on irregular grids, we refer, e.g., to Massioni (2015). On such regular grids, the index of a system in a node is specified by its coordinates in that p -dimensional grid. This results in the definition of p -D systems. An example of such a 1-D system without boundary conditions and only specifying the system indices for the sake of brevity in the nodes is illustrated in Fig. 2. The temporal dynamics may be specified by the order of the local (state space) model that represents the input-output behavior of a local (series of) node(s). The example illustrates that the nodes communicate their states, and hence all these quantities are missing.

On such regular spatial grids, we may have spatially invariant systems where all systems in the nodes are equal or spatially varying systems where the systems dynamics can vary over the nodes.

In this overview we focus mainly on the identification of spatial-temporal dynamical systems within the control engineering approach.

Parametrizing Spatial-Temporal Dynamics

A Priori Parametrization

Similar to many lumped parameter identification methods, the identification of a complex network system starts with the a priori parametrization of both the spatial and temporal dynamics. This is illustrated for the example in Fig. 1a using a state space framework. Let the system in node 1 be described by the local state space model:

$$\begin{aligned} x_1(k+1) &= A_1 x_1(k) + B_1 y_2(k) \\ y_1(k) &= C_1 x_1(k) + u_1(k) + v_1(k) \end{aligned} \quad (3)$$

In a similar way, we can model the other two local nodes. *Lifting* the state quantities to the global state vector $x(k) = [x_1(k)^T \ x_2(k)^T \ x_3(k)^T]^T$ and similarly for the global input $u(k) = u_1(k)$, for the global noise $v(k)$, and for the global output $y(k)$, we obtain the following (global) state space model:

$$\begin{aligned} x(k+1) &= \begin{bmatrix} A_1 & B_1 C_2 & 0 \\ 0 & A_2 & B_2 C_3 \\ B_3 C_1 & 0 & A_3 \end{bmatrix} x(k) \\ &+ \begin{bmatrix} 0 \\ 0 \\ B_3 \end{bmatrix} u(k) + \begin{bmatrix} 0 & B_1 \\ 0 & B_2 \\ B_3 & 0 \end{bmatrix} v(k) \\ y(k) &= \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix} x(k) \\ &+ \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & I \\ 0 & I \end{bmatrix} v(k) \end{aligned} \quad (4)$$

This global state space model can be compactly denoted as:

$$\begin{aligned} x(k+1) &= \mathcal{A}x(k) + \mathcal{B}u(k) + \mathcal{H}v(k) \\ y(k) &= \mathcal{C}x(k) + \mathcal{D}u(k) + \mathcal{G}v(k) \end{aligned} \quad (5)$$

This approach outlined for this simple example can be used for complex network systems, in general resulting in a state vector of extreme

large dimensions and sparse system matrices. The determination of the sparsity pattern of this matrix, i.e., where there are non-zero matrix blocks, falls under the banner of the spatial parametrization. The temporal parametrization comprises both the fixing of the size of these non-zero matrix blocks and the determination of the non-zero entries in these matrices. For the simple three node examples in Fig. 1, the observer canonical form parametrization of each system (assumed to be SISO) will yield an affine parametrization of the global state space model in (5).

Sparse System Matrices of (5)

The example discussed in the previous subsection illustrates that the global system matrices of a network may be sparse. In this subsection we present two sparse parametrizations and their rationale as to what classes of complex networks they may represent.

Decomposable systems When identical dynamical systems are linked in a network and when a communication link exists, it is identical; then such networks can be described as decomposable systems (Massioni and Verhaegen 2009). Examples are formation flying with identical satellites. The global system matrices in (5) of a decomposable system belong to the class of decomposable matrices. This class is defined next.

Definition 1 (Massioni and Verhaegen 2009)

For a given pattern matrix $\mathcal{P} \in \mathbb{R}^{N_n \times N_n}$ and matrices $M_a, M_b \in \mathbb{R}^{n_\ell \times n_r}$, the class of decomposable matrices contain all matrices $\mathcal{M} \in \mathbb{R}^{n_\ell N_n \times n_r N_n}$ of the following form:

$$\mathcal{M} = I_n \otimes M_a + \mathcal{P} \otimes M_b \quad (6)$$

with $\otimes$ representing the Kronecker product.

For the parametrization of decomposable systems, it is only necessary to provide the pattern matrix and to parametrize the matrix pair (M_a, M_b) for each system matrix in (5). For extreme efficient controller synthesis methods for such systems, we refer to Massioni and Verhaegen (2009). A comment on the identifica-

tion methodology is made in section “[Numerical Methodologies](#).”

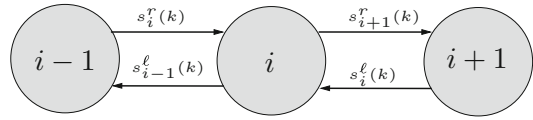
Extension of this parametrization are the so-called α -decomposable systems (Massioni 2014). With this generalization we now have α different systems, each having a decomposable form.

Banded or tri-diagonal system matrices The 1-D network depicted in Fig. 2 may result from the spatial discretization of a system governed via PDEs in one spatial dimension. An example is the series connection of mass spring systems as an approximation of a flexible beam (Liu 2015). Each of these local systems in a regular 1-D grid exchanges information that is in general not measured. When we consider exchange of the state vector of the left and right neighboring systems in Fig. 2 only, the system matrices in the global state space model become block tri-diagonal. When the additional assumption is made that the spatial dynamics are invariant, the system matrices further become block Toeplitz matrices.

This class of systems can also be extended to p -D systems where then the system matrices get multibanded sparse matrices. Such sparse matrices can often be made block tri-diagonal matrices by just permutation, e.g., using the Cuthill-McKee method (Cuthill and McKee 1969). When such permutation exists, such p -D systems may very well be approximated by systems with block tri-diagonal system matrices. A final class of system matrices that may very well be accurately approximated within this class is the class when the system matrices are exponentially spatially decaying (Haber and Verhaegen 2014).

Dense Data Sparse System Matrices of (5)

SSS Systems In D’Andrea and Dullerud (2003) a generic modeling framework was proposed for spatially interconnected dynamical systems. This class will again be explained for 1-D systems as depicted in Fig. 2 for the generalization that the unobservable quantities that represent the communication between the local systems also have spatial dynamics. It has been shown in Rice and Verhaegen (2009) that lifting such local



Identification of Network Connected Systems, Fig. 3

An autonomous 1-D spatial-temporal dynamic system consisting of three local nodes with spatial dynamics. The interacting signals $s_i^*(k)$ are unmeasurable and have spatial dynamics as represented by (7)

models into a global state space model like (5) as was done for the simple example in section “[A Priori Parametrization](#)”, the system matrices of that space model become so-called sequentially semi-separable (SSS) matrices. This lifting procedure and the resulting SSS matrix are shown for the simple autonomous case considered in Fig. 3 considering spatial varying dynamics. For the autonomous 1-D system in Fig. 3, the spatial-temporal dynamics of the i -th system is described as:

$$\begin{bmatrix} x_i(k+1) \\ s_{i+1}^r(k) \\ s_{i-1}^l(k) \end{bmatrix} = \begin{bmatrix} A_i & P_i & U_i \\ Q_i & R_i & 0 \\ V_i & 0 & W_i \end{bmatrix} \begin{bmatrix} x_i(k) \\ s_i^r(k) \\ s_i^l(k) \end{bmatrix} \quad (7)$$

The spatial-temporal dynamics for systems at nodes $i-1$ and $i+1$ are similar (with indices adapted) with the following constraints (for this simple case):

$$\begin{aligned} P_{i-1} &\equiv U_{i+1} \equiv Q_{i+1} \equiv R_{i+1} \equiv W_{i+1} \\ &\equiv R_{i-1} \equiv V_{i-1} \equiv W_{i-1} \equiv 0 \end{aligned}$$

Then the lifted state space model becomes:

$$\begin{aligned} \begin{bmatrix} x_{i-1}(k+1) \\ x_i(k+1) \\ x_{i+1}(k+1) \end{bmatrix} &= \begin{bmatrix} A_{i-1} & U_{i-1}V_i & U_{i-1}W_iV_{i+1} \\ P_iQ_{i-1} & A_i & U_iV_{i+1} \\ P_{i+1}R_iQ_{i-1} & P_{i+1}Q_i & A_{i+1} \end{bmatrix} \\ &\begin{bmatrix} x_{i-1}(k) \\ x_i(k) \\ x_{i+1}(k) \end{bmatrix} \end{aligned} \quad (8)$$

The system matrix of this autonomous system has the SSS structure. SSS systems are LTI systems of the form (5) with all system matrices having the SSS structure. This structure is defined for the spatially invariant case in Definition 2. For its generalization to the spatially varying case, we refer to Rice and Verhaegen (2009).

Definition 2 (Rice and Verhaegen 2009)

The matrix $\mathcal{A}$ is an SSS matrix when it is defined by its generator of seven matrices (P, R, Q, A, U, W, V) of resp. sizes $n \times r_2$, $r_2 \times r_2$, $r_2 \times n$, $n \times n$, $n \times r_1$, $r_1 \times r_1$, and $r_1 \times n$, in the following way:

$$\mathcal{A} = \begin{bmatrix} A & UV & UWV & UW^2V & UW^3V & UW^4V & UW^5V \\ PQ & A & UV & UWV & UW^2V & UW^3V & UW^4V \\ PRQ & PQ & A & UV & UWV & UW^2V & UW^3V \\ PR^2Q & PRQ & PQ & A & UV & UWV & UW^2V \\ PR^3Q & PR^2Q & PRQ & PQ & A & UV & UWV \\ PR^4Q & PR^3Q & PR^2Q & PRQ & PQ & A & UV \\ PR^5Q & PR^4Q & PR^3Q & PR^2Q & PRQ & PQ & A_s \end{bmatrix} \quad (9)$$

The generator of the SSS matrix is the (generally) small-dimensional matrices P, R, Q, A, U, W, V . Therefore parametrizing these matrices only makes the parametrization *sparse*, while the full matrices like $\mathcal{A}$ are dense. This explains the indication dense data sparse. Usually these dense matrices are never formed explicitly conducting all operations on the generators only (Rice and Verhaegen 2009).

What is particular about this matrix class is that all the corner matrix partitions above and below the block diagonal, two of which are depicted in (9), have low rank.

Matrix or multilinear state space models In multidimensional signal processing, the observations made are not vectors but multidimensional objects. For example, in imaging the recorded images are 2D objects that may be stored in matrices.

This motivated (Squin and Verhaegen 2019) to propose matrix state space models (MSSM) of the following form:

$$\begin{aligned} X(k+1) &= A_1 X(k) A_2^T + B_1 U(k) B_2^T \quad (10) \\ Y(k) &= C_1 X(k) C_2^T + V(k) \end{aligned}$$

Here the state, input, and output are given in matrix form by the matrices $X(k)$, $U(k)$, and $Y(k)$, resp. Each of these matrices is defined

on a grid, which for simplicity is taken in this overview as an $N \times N$ grid. If, for example, the entry of the output at one grid point is p , the size of the matrix $Y(k)$ is $Np \times N$. When vectorizing these matrix quantities, we get a global state space model like in (5) for $\mathcal{H}$ and $\mathcal{D}$ equal to zero, with the system matrices given by Kronecker products, such as:

$$\mathcal{A} = A_1 \otimes A_2$$

This approach can be generalized by considering higher Kronecker rank expressions of the system matrices, given, for example, for $\mathcal{A}$ as:

$$\mathcal{A} = \sum_{i=1}^{n_a} A_{1,i} \otimes A_{2,i} \quad (11)$$

Such expression is known to be able to approximate a given linear operator as highlighted, for example, in Bamieh and Filo (2018). Dynamical systems like (5) with its system matrices having the form as in (11) are called multilinear dynamical systems (MLDS) (Rogers and Russell 2013).

Numerical Methodologies

In this section we illustrate some of the numerical challenges related to the three classes of model parametrizations discussed in the previous sec-

tion. This is done by highlighting one representative solution for each class.

Prediction Error Methods for a Prior Parametrized Models

Having parametrized the global model (5) via the parameter vector θ that include the characterization of the noise term $v(k)$, one can propose a predictor $\hat{y}(k, \theta)$ of the output. To find a “best” estimate of θ , a prediction error cost function can be defined as:

$$J(\theta) = \frac{1}{N_t} \sum_{k=1}^{N_t} \|y(k) - \hat{y}(k, \theta)\|^2 \quad (12)$$

where N_t data points are considered. The solution of this optimization problem in the signal flow framework with parametrization of the dynamic structure function in (1) is followed in Materassi and Salapaka (2012) and Dankers et al. (2016). The latter paper discusses the topic of predictor input selection to exploit the given structure in the dynamic structure function for reducing the computational complexity. However for large-scale networks, this approach remains extremely complex.

When abiding to the state space approach, one is challenged with the identifiability of the parameter vector θ used to parametrize the system matrices. This can be dealt with a priori by analyzing the structural identifiability of the parameters as, e.g., done in van den Hof (2002) or taken care about during the calculations by including a regularization term in the cost function (12). The latter is demonstrated in sparse Bayesian estimation using the expectation maximization algorithm in Yue et al. (2018) for a very specific measurement scenario considering the following matrices in (5):

$$C = [I \ 0] \quad D = 0 \quad H v(k) = \sigma e(k)$$

$$\mathcal{G}v(k) = e(k)$$

for $\sigma \in \mathbb{R}_+$ and $e(k)$ Gaussian zero-mean white noise with identity covariance matrix. An interesting though preliminary development to bridge the two approaches outlined in section “[The Signal Flow Versus the Control](#)

[Engineering Approach](#)” is also outlined in Yue et al. (2018). For the specific case, the transfer function $K(q)$ in (2) is square and diagonal; a parametrization of the system matrices $\mathcal{A}, \mathcal{B}$ in the state space model (5) has been proposed that makes the corresponding underlying dynamic structure function model network identifiable. However such exchange of features between signal flow approach and control engineering approach does not hold in general. For example, it was remarked in Yue et al. (2018) that the sparsity of the state space model in (5) does not mean that the transfer functions in the dynamic structure function (1) are sparse.

One attempt to reduce the computational burden in prediction error methods (PEM) is via the sparse parametrization of SSS system matrices, as done in Rice and Verhaegen (2011). However apart from the possible reduction in computational cost, PEM for large-scale systems suffer in general from the drawbacks of getting initial estimates, local minima, etc. This is because for the large-scale case, no generic subspace methodology is yet available that can be used to provide initial estimates as done for the identification of small-dimensional LTI systems.

To overcome the computational burden of estimating all parameters at once, an important research activity is to try to split up the global cost function in independent parts, each trying to identify a local system in the network. This research activity requires knowledge of the network topology. Some preliminary results for the signal flow approach can be found in Weerts et al. (2019). Here the methodology called “abstraction of linear dynamic networks” is presented to remove unknown signals that influence local models by replacing these unknown signals by signals calculated from measured signals so to enable the identification of the local models.

Subspace Identification for Sparse Network Models

Different attempts have been made to extend important advantages of subspace identification to network-connected systems. One such extension is in Yu and Verhaegen (2018). This method allows to identify the classes of sparse networks

as defined in the section “[Sparse System Matrices of \(5\)](#)” by solving a series of local identification problems. Though overcoming the computational challenge in this way, the system identification challenge becomes that we have to deal with the missing information about the interaction between the different local systems as well as block sizes of the matrices in, e.g., the block tri-diagonal parametrization. This information is extracted from the data as is characteristic for subspace methods.

The main steps of the approach in Yu and Verhaegen (2018) are briefly highlighted for the network in Fig. 2 with the global model (5) for the simple case that $\mathcal{H}$ and $\mathcal{D}$ are zero, $\mathcal{G} = I$ and the other system matrices having the following form:

$$\mathcal{A} = \begin{bmatrix} A_1 & A_{1,r} & & & \\ A_{2,\ell} & A_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & & A_{N-1,r} & \\ & & & A_{N,\ell} & A_N \end{bmatrix} \quad (13)$$

$$\mathcal{B} = \text{Diag}(B_1, B_2, \dots, B_N),$$

$$\mathcal{C} = \text{Diag}(C_1, C_2, \dots, C_N)$$

The subspace identification methods focus on the identification of the i -th local system with model:

$$x_i(k+1) = A_i x_i(k) + [A_{i,\ell} \ A_{i,r}] \begin{bmatrix} x_{i-1}(k) \\ x_{i+1}(k) \end{bmatrix} \\ + B_i u_i(k) \quad y_i(k) = C_i x_i(k) + v_i(k) \quad (14)$$

These matrices characterize a spatially varying network. The missing information in addition to the local state vector $x_i(k)$ are the state vectors $x_{i-1}(k)$ and $x_{i+1}(k)$ of the neighboring nodes. This turns the identification problem into the class of *blind identification problems*. These are challenging problems for which no generic solution exists, and in general special structure of the problem needs to be exploited. To create such special structure, the following approach is taken in (Yu and Verhaegen 2018):

1. The system is partially lifted by collecting the local systems in the dashed boxes in Fig. 2

only for small values of R . For example, for the systems in the nodes $i+1 : i+R$, this results into the following state space model:

$$x_{i+1:i+R}(k+1) = A_{i+1:i+R} x_{i+1:i+R}(k) \\ + B_{i+1:i+R}^{\text{ext}} s_{i+1:i+R}(k) \\ + B_{i+1:i+R} u_{i+1:i+R}(k) \\ y_{i+1:i+R} = C_{i+1:i+R} x_{i+1:i+R}(k) \\ + v_{i+1:i+R}(k) \quad (15)$$

where the notation $\bullet_{i+1:i+R}(k)$ represents the stacking of all vectors or matrices $\bullet$ from nodes $i+1 : i+R$. Apart from the straightforwardly derived structure of the matrices $A_{i+1:i+R}$, $B_{i+1:i+R}$, and $C_{i+1:i+R}$, the matrix $B_{i+1:i+R}^{\text{ext}}$ has the following particular structure:

$$B_{i+1:i+R}^{\text{ext}} = \begin{bmatrix} A_{i+1,\ell} & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & A_{i+R,r} \end{bmatrix} \quad (16)$$

For these partially lifted state space model, the data equation that relates the block Hankel matrices of input and output data in subspace identification (Verhaegen 2015) is then subsequently constructed.

2. The structure of zeros in the matrix $B_{i+1:i+R}^{\text{ext}}$ in (16) and that in the similar matrix $B_{i-R+2:i+1}^{\text{ext}}$ is then exploited to find the state sequence $x_{i+1}(k)$ as the intersection between block Hankel matrices constructed from measured input and output data that appear in the data equations constructed in the previous step (Yu and Verhaegen 2018). In the same way, the state sequence $x_{i-1}(k)$ can be reconstructed. These intersection problems reveal the size of the state vectors $x_{i+1}(k)$ and $x_{i-1}(k)$ (which may be different).
3. The knowledge of the state sequences $x_{i+1}(k)$ and $x_{i-1}(k)$ allow to use classical subspace identification methods (Verhaegen 2015) to estimate (without parametrization) the

unknown system matrices and the size of the state vector $x_i(k)$ in (14).

Similar approaches by partial lifting have been derived for the identification of α -decomposable systems in Yu and Verhaegen (2018).

Subspace-Like Identification for Dense, Data Sparse Network Models

The identification of dense, data sparse network models turns out to be much harder than the sparse models considered in section “[Subspace Identification for Sparse Network Models](#).”

For the SSS systems, the methodology in Yu and Verhaegen (2018) does not apply as the partially lifted set of systems in the first step of the algorithm does no longer give rise to a sparse input weighting matrix of the unknown inputs of that lifted model like in (16). The derivation of a subspace algorithm for these class of systems is still ongoing research.

For the simple multilinear state space model in (10), a subspace-like identification methodology has recently been proposed. This method enables to derive information about key structural parameters like the size of the matrices A_1 and A_2 from data like in subspace methods; the numerical steps involve non-convex optimization problem, though of small size, and hence the name “subspace-like.”

For system identification, but also for control, these multidimensional systems postulate basic questions. An example is the “ambiguity” in terms of scaling and multiplication with invertible matrices of the system matrices that preserve the input-output behavior. For MSSM system such questions are dealt with in Siquin and Verhaegen (2019).

The solution in Siquin and Verhaegen (2019) considers the identification of the system matrices A_i, B_i, C_i in (10) up to “ambiguity” transformations. The method starts with the estimation of the (Markov) parameters $M_{i,r}$ and $M_{i,\ell}$ by approximating the input-output relationship under the assumption that (10) is stable:

$$Y(k) = \sum_{i=1}^{\infty} M_{i,r} U(k-i) M_{i,\ell}^T + V(k)$$

with an FIR model. Using alternating convex optimization, conditions for global convergence have been derived to find scaled variants of the parameters $M_{i,r}, M_{i,\ell}$ in the following way (Siquin and Verhaegen 2019):

$$\hat{M}_{i,r} \approx t_i M_{i,r} \quad \hat{M}_{i,\ell} \approx v_i M_{i,\ell} \quad \text{where } t_i v_i \approx 1$$

The approximation becomes an equality in the limit of increasing the order of the FIR model.

A bilinear optimization problem is then proposed in Siquin and Verhaegen (2019) to rescale these FIR parameters $\hat{M}_{i,r}, \hat{M}_{i,\ell}$ such that when stored into Hankel matrix, the latter gets (small) finite rank that should correspond to the orders of the system matrices A_1 and A_2 , respectively. These rescaled FIR parameters are then used in a 3D tensor framework (Siquin and Verhaegen 2019) to calculate the state sequence and the system matrices via SVDs. The latter provides structural information on the sizes of the system matrices in the model (10).

Summary and Future Directions

Two approaches are currently under full development for the identification of large-scale network-connected systems. Both have their merits and pitfalls. The selection of the most appropriate approach depends on what the model is used for. In this way the structured state space approach may be useful in the efficient design of controllers for network-connected systems as demonstrated in, e.g., in Rice and Verhaegen (2009) and Massioni and Verhaegen (2009). This is also the case for the identification methods presented in Siquin and Verhaegen (2019). The subspace-like methods presented in this overview may therefore open the avenue for *identification methods for distributed control of large-scale network-connected systems*.

This research field is rapidly expanding, and there are many possibilities for further research. Just a few examples are the development of new (subspace-like) algorithms for the open problems defined in this overview like for multilinear dynamical systems (MLDS). One interest-

ing development looming on the horizon is the parametrization and identification of network-connected systems in the context of tensor calculus (Sinquin and Verhaegen 2019; Bousse et al. 2017). Finally, the approaches outlined in this overview are able to identify large-scale state space models. A challenge remains to derive the network topology from these system matrices.

Cross-References

- [Networked Systems](#)
- [Subspace Techniques in System Identification](#)

Recommended Reading

To start working in this field, a compact introduction to the signal flow approach is Gevers et al. (2019). This paper also contains many interesting contributions in that approach.

A recent overview and introduction to the identification of MLDS are the PhD thesis of B. Sinquin (2019). Here also the integration of the identification methods with control of large-scale systems such as extreme adaptive optics (XAO) is discussed as well as a validation study on an AO breadboard with a large number of inputs and outputs ($O(10^3)$).

Bibliography

- Bamieh B, Filo M (2018) An input-output approach to structured stochastic uncertainty. Technical Report. arXiv:1806.07473v1
- Bousse M, Debals O, Lathauwer LD (2017) Tensor-based large-scale blind system identification using segmentation. *IEEE Trans Signal Process* 65(21):5770–5784
- Cuthill E, McKee J (1969) Reducing the bandwidth of sparse symmetric matrices. In: *Proceedings of the 1969 24th national conference of ACM*, New York, pp 157–172
- D’Andrea R, Dullerud G (2003) Distributed control for spatially interconnected systems. *IEEE Trans Autom Control* 48(9):1478–1495
- Dankers A, van den Hof P, Heuberger P, Bombois X (2016) Identification of dynamic models in complex networks with prediction error methods: predictor input selection. *IEEE Trans Autom Control* 61(4):937–952
- Gevers M, Bazanella A, Pimentel G (2019) Identifiability of dynamical networks with singular noise spectra. *IEEE Trans Autom Control* 64(6):2473–2479
- Gonçalves J, Warnick S (2008) Necessary and sufficient conditions for dynamical structure reconstruction of LTI networks. *IEEE Trans Autom Control* 53(7):1670–1674
- Haber A, Verhaegen M (2014) Subspace identification of large-scale interconnected systems. *IEEE Trans Autom Control* 59(10):2754–2759
- Liu Q (2015) Modeling, distributed identification and control of spatially-distributed systems with application to an actuated beam. Ph.D. dissertation, Institute of Control Systems, Hamburg University of Technology
- Massioni P (2014) Distributed control α -heterogeneous dynamically coupled systems. *Syst Control Lett* 72:30–35
- Massioni P (2015) Decomposition methods for distributed control and identification. Ph.D. dissertation, Delft Center for Systems and Control, Delft University of Technology
- Massioni P, Verhaegen M (2009) Distributed control for identical dynamically coupled systems: a decomposition approach. *IEEE Trans Autom Control* 54(1):124–135
- Materassi D, Salapaka M (2012) On the problem of reconstructing an unknown topology via locality properties of the wiener filter. *IEEE Trans Autom Control* 57(7):1765–1777
- Rice J, Verhaegen M (2009) Distributed control: a sequential semi-separable approach for spatially heterogeneous linear systems. *IEEE Trans Autom Control* 54(6):1270–1283
- Rice J, Verhaegen M (2011) Efficient system identification of heterogeneous distributed systems via a structure exploiting extended Kalman filter. *IEEE Trans Autom Control* 56(7):1713–1718
- Rogers LLM, Russell S (2013) Multilinear dynamical systems for tensor time series. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds) *Proceedings of advances in neural information system*, vol 26. Curran Associates, Inc., pp 2634–2642
- Sinquin B (2019) Structured matrices for predictive control of large and multi-dimensional systems. Ph.D. dissertation, Delft University of Technology
- Sinquin B, Verhaegen M (2019) K4sid: large-scale subspace identification with Kronecker modeling. *IEEE Trans Autom Control* 64(3):960–975
- van den Hof J (2002) Structural identifiability of linear compartmental systems. *IEEE Trans Autom Control* 43(6):800–818
- Verhaegen M (2015) Subspace techniques in system identification. In: Baillieul J, Samad T (eds) *Encyclopedia of systems and control*. Springer, London
- Weerts H, Linder J, Enqvist M, van den Hof P (2019) Abstractions of linear dynamic networks for input selection in local module identification. Technical Report. arXiv:1901.00348v1

- Yu C, Verhaegen M (2018) Subspace identification of individual systems operating in a network (si²on). *IEEE Trans Autom Control* 63(4):1120–1125
- Yue Z, Thunberg J, Ljung L, Gonçalves J (2018) A state-space approach to sparse dynamic network reconstruction. Technical Report. arXiv:1811.08677v1

Image-Based Estimation for Robotics and Autonomous Systems

A. P. Dani

Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA

Abstract

This entry discusses methods to estimate the structure of the objects/scene observed using a series of moving camera images. To recover the structure, it is sufficient to estimate the depth of the feature points on the object. To this end, two observer-based depth estimation methods are presented. The first method is a concurrent learning-based observer for estimating the depth of the static feature points and the second method is an unknown input observer for estimating the depth of the moving feature points.

Keywords

Structure from motion · Range estimation · Nonlinear observers · Machine learning

Introduction

Recovering the structure of a static scene (i.e., the 3D Euclidean coordinates of feature points), using a moving camera is termed as ‘structure from motion (SfM)’. A number of solutions to the SfM problem are given in the form of batch or offline methods (Hartley and Zisserman 2003; Chaumette and Hutchinson 2006; Fathian et al. 2018) and causal or online methods (Jankovic

and Ghosh 1995; Dixon et al. 2003; Dani et al. 2012a). The solutions in computer vision applications mostly rely on offline processing of the data from a collection of images of the scene observed from different viewpoints. New advances in deep learning provide deep convolution neural network (CNN)-based solutions for estimating depth of every pixel in the image, but these methods require lot of training data to train the CNNs and do not utilize any geometric relationship information inherent to the problem. When the camera sensor is used in the context of navigation or visual feedback control, real-time estimation of the feature depth is required using online methods. The causal or online methods rely on image-based depth estimation using series of images obtained from a single moving camera and camera motion information. Feature depth estimation is one of the important topics in vision-based navigation and visual feedback control design for robotics and autonomous systems. The fundamental problem behind feature depth estimation is a triangulation problem. Since the object is assumed to be stationary, a moving camera can capture snapshots of the object from two different locations, and triangulation is feasible. When the camera and object both are moving, then the triangulation is not feasible from two views, which brings additional challenges to the design of online methods to recover the depth of feature points.

Many advances have been made in the field of online feature depth estimation of feature on static objects using extended Kalman filtering (Matthies et al. 1989) and nonlinear observer schemes (Jankovic and Ghosh 1995; Dixon et al. 2003; Hu et al. 2008; Dani et al. 2012b). The main idea of using observers is to utilize the most recent camera motion (linear and angular velocities and/or accelerations) and feature points data from the current image along with past depth estimates to update the current depth estimate. For the feature depth observers to converge, observability condition based on persistency of excitation (PE) must be satisfied. The PE condition provides conditions on camera motions that are least and most informative for the inverse depth estimation. In many practical application

scenarios, the image motion is very close to the least informative motion from the camera images. Examples of such scenarios include vertical landing of quadcopter/flying cars using downward-looking camera feedback, an approach motion along optical axis of camera for picking an object or a fruit by a robot manipulator using camera in-hand feedback. Monocular image feedback does not provide reliable feature depth information in these cases. Various other methods can be used to avoid this problem, such as a stereo camera pair or design of active perception, planning, and active control methods (Spica et al. 2017) that generate more informative camera motion for estimation of the feature depth. The use of stereo camera may not always be a feasible option due to its form factor and requires precise calibration and recalibration. Active structure from motion methods may not always find a solution to the path planning problem and may run into the issue of losing the features from the field of view (FOV). Also, in practical scenarios, the motion generated by most informative paths for depth estimation may not be desirable for successful completion of the control task. In addition, if the object is also moving, then the standard observers for depth estimation do not work since additional terms related to unknown motion of the moving object appear in the dynamics along with its depth parameter. Some offline solutions exist to solve this problem (Avidan and Shashua 2000), but they are not very useful for real-time feedback control.

In contrast, this chapter examines recent and emerging efforts that provide solutions to two important problems – (i) relaxation of restricted motion conditions present in depth estimation problem for static scene and (ii) the estimation of depth in dynamic scene. For the *first* problem, a concurrent learning (CL) method from adaptive control is used to design depth observers that reliably estimate the depth of the feature points when the camera motion paths are not very informative. The CL-based observers use data stored from the previous camera motion and depth estimates in the online depth estimation scheme. For the *second* problem, advances in the field of unknown input observer (UIO)

provide online solutions to the depth estimation in dynamics scenes where the unknown motion of the object is modeled as an unknown input. In the domain of online methods for depth estimation, the solutions to the aforementioned two problems provide significant advances for the utility of these algorithms in practical scenarios commonly observed in robotics and autonomous systems applications.

Image-Based Inverse Depth Dynamics

The movement of a perspective camera observing a scene results in the change of location of a feature point belonging to a static object in the image plane. Let $\bar{m}(t) = [X, Y, Z]^T \in \mathbb{R}^3$ and $m_n(t) = [\frac{X}{Z}, \frac{Y}{Z}, 1]^T \in \mathbb{R}^3$ be the Euclidean and normalized Euclidean coordinates of a feature point belonging to a static or moving object captured by a moving camera in the camera reference frame $\mathcal{F}_C$ with known camera velocities in the world reference frame $\mathcal{F}^*$. The normalized Euclidean coordinates can be converted to pixel coordinates $p = Am_n$ where $A \in \mathbb{R}^{3 \times 3}$ is the known camera calibration matrix. To estimate the depth, define an auxiliary vector $y(t) \in \mathbb{R}^3$, given by $x(t) = [\frac{X}{Z}, \frac{Y}{Z}, \frac{1}{Z}]^T$. Let $s(t) \in \mathbb{R}^2$ be an image plane feature point with first two components of $x(t)$ and $\chi \in \mathbb{R}$ be the inverse depth associated with the feature point. For a single feature point, define the state as $s = [x_1(t), x_2(t)]^T$ and the inverse depth as $\chi = x_3(t)$. Taking the time derivative of s , the dynamics of s as a function of linear and angular velocities of the camera can be expressed as

$$\dot{s} = f_m(s, \omega_c) + \Omega(s, v_c)\chi + d_s(s, v_p, \chi) \quad (1)$$

where $f_m(s, \omega_c) \in \mathbb{R}^2$ and $\Omega(s, v_c) \in \mathbb{R}^2$ are functions of measurable quantities or known quantities, $d_s(s, v_p, \chi) \in \mathbb{R}^2$ is a function of unmeasurable quantities, $v_c = [v_{cX}, v_{cY}, v_{cZ}]^T \in \mathbb{R}^3$ is the measurable linear velocity, $\omega_c = [\omega_{cX}, \omega_{cY}, \omega_{cZ}]^T \in \mathbb{R}^3$ is the measurable angular velocity of the camera, and $v_p = [v_{pX}, v_{pY}, v_{pZ}]^T \in \mathbb{R}^3$ is the linear

velocity of the moving point object. The dynamics associated with the inverse depth χ can be modeled as

$$\dot{\chi} = f_u(s, \chi, u) + d_\chi(\chi, v_p) \quad (2)$$

where $d_\chi(v_p, \chi) \in \mathbb{R}$ is a function of unmeasurable quantities. The state derivative $\dot{s} \in \mathbb{R}^2$ is not measurable and can only be estimated. If the object is stationary, then the terms $d_s(s, v_p, \chi)$ and $d_\chi(\chi, v_p)$ become zero and, hence, disappear from (1) and (2).

Problem Definition Given the measurements of feature points, $s(t)$ in successive frames and camera velocities $v_c(t)$ and $\omega_c(t)$ estimate the inverse depth $\frac{1}{\chi(t)}$ and, hence, 3D coordinate of the feature point, $\hat{m}(t)$. *Problem 1* – Perform the estimation when the object in the scene is stationary. *Problem 2* – Perform the estimation when the object in the scene is moving with non-zero unmeasurable velocity.

Depth Observers

Concurrent Learning-Based Depth Observer for Static Objects

CL (Chowdhary et al. 2013; Kamalapurkar et al. 2017) is a recently developed technique in adaptive control for parameter estimation, where the knowledge of recorded state information data, also known as the history stack of the system, is leveraged to estimate the parameters and achieve state tracking. CL methods can be used to guarantee parameter convergence in adaptive control without relying on the PE condition. Learning from recorded data is effective since it is based on the model error, which is closely related to the parameter estimation error. The use of history of system state information relaxes the PE condition to a finite excitation condition, which depends upon the rank of the regressor matrix (Chowdhary and Johnson 2010). Such a condition can be easily monitored online in comparison to the PE condition.

In the depth estimation dynamics (1)–(2), the inverse depth appears as a time-varying parameter in (1) with associated known dynamics. The convergence of depth estimation observers designed for the system in (1)–(2) relies on the satisfaction of the PE condition. For example, results in Chen and Kano (2004) and Dixon et al. (2003) provide discontinuous and continuous observer structure for depth estimation, respectively, under the assumption of known camera motion. Observer structures Dani et al. (2012a, b) provide a reduced order globally exponentially stable observer under fully known camera motion and partially unknown camera motion, respectively. All these depth observers require PE condition to be satisfied for their convergence results. Although many camera motions in practice satisfy the PE condition, in certain practical scenarios, motions that occur naturally during vision-based control tasks do not satisfy the PE condition. Then, the fundamental question is whether the concept of CL can be used to build a memory in the depth observer structure to relax the PE condition for estimating the depth in (1)–(2). Results in Rotithor et al. (2019) illustrate that such modifications to the depth observers are possible and provide full order observer structure to the problem. Since the depth is a time-varying parameter in the PDS, the exact estimation of the depth (asymptotic convergence of the observer error) in the absence of PE condition using CL-based observers is not possible. But the stability analysis of the observer structure with history stack information reveals that the recent history of the measurable part of the system state, state estimates, and camera motion contains information about the current inverse depth $\chi(t)$ and can be used to estimate the depth within a uniform ultimate bound even when the camera motion does not satisfy the PE condition.

Unknown Input Observer for Moving Objects

The problem of estimating the feature point depth in the presence of object motion adds another level of complexity to the estimation problem. In the system dynamics, additional unmeasurable

moving object's velocities appear that are multiplied by the unknown depth parameter. From the geometric perspective, batch solutions are provided in Avidan and Shashua (2000) for special object motions, such as straight lines or conic trajectories given five or nine views, respectively, and for more general curves in Kaminiski and Teicher (2004). Batch algorithms use an algebraic relationship between 3D coordinates of points in the camera coordinate frame and corresponding 2D projections on the image frame collected over n images to estimate the depth and structure. For visual servo control or video-based surveillance tasks, online structure estimation algorithms are required. Results in Dani et al. (2011) show that causal algorithms can be developed for estimating the structure of a moving object using a moving camera in the unknown input observer (UIO) framework. Conditions on object motion are developed when the estimation is exact in the sense of asymptotic convergence of the observer error. The object is assumed to be moving on a ground plane with arbitrary velocities observed by a downward-looking camera with arbitrary linear motion in 3D space. The advantage of causal methods of depth estimation is that no assumptions are required on the minimum number of points or minimum number of views necessary to estimate the structure in the design of the observer structure. In addition, online depth estimation methods are readily amenable to their incorporation in the feedback control loop structure.

Summary and Future Directions

Many online methods are available to estimate the depth of the feature points from the monocular camera feedback. These methods utilize the recursive estimation methods that use the structure of the nonlinearity of system and are robust to the external disturbances and sensor noise. An alternative to the aforementioned nonlinear observer schemes is to utilize observer theory from positive systems since feature depth

is always positive. Another alternative is to scale up the observer structure to include multiple feature points and include geometric constraints of the scene learned using machine learning methods, i.e., learn the physical relationship between the feature points in the observer structure. This confluence of machine learning and observer theory has a good potential of providing solutions to practical problems. Also, these methods can be extended to cases where the intermittent image or camera motion feedback is available to the observer scheme (cf. Parikh et al. 2018).

The utility of the online observers for depth or relative depth estimation for long-term simultaneous localization and mapping (SLAM) and long-term navigation of the autonomous systems using image-based feedback needs to be tested. Specifically, studying the CL-based observers in the context of utilizing memory in the estimation scheme by storing and observing similar patterns occurred in the past can lead to more reliable estimation scheme for long-term autonomy and navigation. With the continued development of the CL-based and other learning schemes in observer design, more long-term estimation schemes are likely to emerge. With the advancements of deep learning, image-based estimation for imitation learning or learning from demonstration (Ravichandar and Dani 2019) in robotics and autonomous systems will also be a potential direction of growth.

Cross-References

- ▶ [Cooperative Manipulators](#)
- ▶ [Disaster Response Robot](#)
- ▶ [Flexible Robots](#)
- ▶ [Force Control in Robotics](#)
- ▶ [Model-Free Reinforcement Learning-Based Control for Continuous-Time Systems](#)
- ▶ [Multiagent Reinforcement Learning](#)
- ▶ [Multiple Mobile Robots](#)
- ▶ [Parallel Robots](#)
- ▶ [Physical Human-Robot Interaction](#)

- [Redundant Robots](#)
- [Rehabilitation Robots](#)
- [Reinforcement Learning and Adaptive control](#)
- [Reinforcement Learning for Approximate Optimal Control](#)
- [Reinforcement Learning for Control Using Value Function Approximation](#)
- [Robot Grasp Control](#)
- [Robot Learning](#)
- [Robot Motion Control](#)
- [Robot Teleoperation](#)
- [Robot Visual Control](#)
- [Underactuated Robots](#)
- [Underwater Robots](#)
- [Walking Robots](#)
- [Wheeled Robots](#)

Recommended Reading

There are many excellent books describing the structure from motion and depth estimation in deeper detail (see Hartley and Zisserman 2003; Horn et al. 1986). The other papers cited in this entry will also provide greater understanding (Matthies et al. 1989; Jankovic and Ghosh 1995; Avidan and Shashua 2000; Chaumette and Hutchinson 2006).

Bibliography

- Avidan S, Shashua A (2000) Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence. *IEEE Trans Pattern Anal Mach Intell* 22(4):348–357
- Chaumette F, Hutchinson S (2006) Visual servo control. I. Basic approaches. *IEEE Robot Autom Mag* 13(4): 82–90
- Chen X, Kano H (2004) State observer for a class of nonlinear systems and its application to machine vision. *IEEE Trans Autom Control* 49(11):2085–2091
- Chowdhary G, Johnson E (2010) Concurrent learning for convergence in adaptive control without persistency of excitation. In: 2010 49th IEEE conference on decision and control (CDC). IEEE, pp 3674–3679
- Chowdhary G, Yucelen T, Muhlegg M, Johnson EN (2013) Concurrent learning adaptive control of linear systems with exponentially convergent bounds. *Int J Adapt Control Signal Process* 27(4): 280–301
- Dani A, Kan Z, Fischer N, Dixon WE (2011) Structure estimation of a moving object using a moving camera: an unknown input observer approach. In: *IEEE conference on decision and control*, Orlando, pp 5005–5010
- Dani A, Fischer N, Kan Z, Dixon W (2012a) Globally exponentially stable observer for vision-based range estimation. *Mechatronics* 22(4):381–389
- Dani A, Fisher N, Dixon WE (2012b) Single camera structure and motion. *IEEE Trans Autom Control* 57(1): 241–246
- Dixon WE, Fang Y, Dawson DM, Flynn TJ (2003) Range identification for perspective vision systems. *IEEE Trans Autom Control* 48:2232–2238
- Fathian K, Ramirez-Paredes JP, Doucette EA, Curtis JW, Gans NR (2018) QuEst: a quaternion-based approach for camera motion estimation from minimal feature points. *IEEE Robot Autom Lett* 3(2): 857–864
- Hartley R, Zisserman A (2003) Multiple view geometry in computer vision. Cambridge University Press, Cambridge
- Horn B, Klaus B, Horn P (1986) Robot vision. MIT press
- Hu G, Aiken D, Gupta S, Dixon W (2008) Lyapunov-based range identification for a paracatadioptric system. *IEEE Trans Autom Control* 53(7):1775–1781
- Jankovic M, Ghosh BK (1995) Visually guided ranging from observations of points, lines and curves via an identifier based nonlinear observer. *Syst Control Lett* 25(1):63–73
- Kamalapurkar R, Reish B, Chowdhary G, Dixon WE (2017) Concurrent learning for parameter estimation using dynamic state-derivative estimators. *IEEE Trans Autom Control* 62(7):3594–3601
- Kaminski JY, Teicher M (2004) A general framework for trajectory triangulation. *J Math Imaging Vis* 21(1–2): 27–41
- Matthies L, Kanade T, Szeliski R (1989) Kalman filter-based algorithms for estimating depth from image sequences. *Int J Comput Vis* 3(3):209–238
- Parikh A, Kamalapurkar R, Dixon WE (2018) Target tracking in the presence of intermittent measurements via motion model learning. *IEEE Trans Robot* 34(3):805–819
- Ravichandar H, Dani A (2019) Learning pose dynamics from demonstrations via contraction analysis. *Auton Robots* 43(4):897–912
- Rotithor G, Saltus R, Kamalapurkar R, Dani AP (2019) Observer design for structure from motion using concurrent learning. In: *Proceedings of American control conference*
- Spica R, Robuffo Giordano P, Chaumette F (2017) Coupling active depth estimation and visual servoing via a large projection operator. *Int J Robot Res* 36(11): 1177–1194

Image-Based Formation Control of Mobile Robots

Guoqiang Hu, Zhi Feng, and Renhe Guan
School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore,
Singapore

Abstract

This chapter addresses an image-based cooperative formation control problem of multiple mobile robots. To develop the image-based feedback controller, the homography matrix or essential matrix techniques are employed to achieve geometric relationships that are beneficial to image-based formation control. Then, image-based cooperative state estimation and cooperative formation control designs are considered, respectively. Finally, a short discussion section is provided to summarize existing research and to point out several future research directions.

Keywords

Image-based formation · Cooperative state estimation · Cooperative control

Introduction

Robot formation control (e.g., see Egerstedt and Hu (2001), Balch and Arkin (1998), Das et al. (2002), Paley and Leonard (2007), Sun et al. (2009), and Ren (2010), just name a few) is an interesting topic and has a wide range of applications in robotics and transportation. This chapter will focus on cooperative state estimation and cooperative formation control, two important problems under this topic.

The cooperative state estimation problem is motivated by a vision-based formation control problem based on relative measurements (Cook and Hu 2012). One of the convenient ways to make relative measurements within a formation of robots is to use cameras mounted to the robots.

Because of the field of view constraints, occlusions, and resolution that limit the cameras' measurement capabilities, it is unlikely that each robot will be able to make all the measurements necessary to completely describe the formation. That is, each robot can only measure part of the relative state information, while the rest of the information is measured by other robots. In order for the robots to have a precise estimate of the state in a distributed manner, there needs to be a method for combining and distributing the information measured by different robots.

Cooperative formation control of multi-robot systems has caught a growing interest in both the robotics and control communities. The objective of this problem is to enable a team of robots to maintain a certain formation configuration while performing a group motion. Several methods are proposed to address this problem (e.g., virtual structure method (Egerstedt and Hu 2001), behavior-based method (Balch and Arkin 1998), leader-following method (Das et al. 2002), potential function method (Paley and Leonard 2007), synchronous formation method (Sun et al. 2009), and consensus method (Ren 2010)). The position and orientation of each robot in a global reference frame are often required for cooperative formation control of the robotic network. However, it is desirable to remove the dependence on a global reference frame due to sensing and communication limits in practical systems. The removal of global reference frame leads toward describing the state of the formation in terms of the relative positions and orientations of the robots or in terms of the geometry of the formation. Cameras and computer vision algorithms are capable of making relative measurements.

Geometric Relationships

This section focuses on the geometric relationships that are beneficial to image-based formation control. To develop control laws, it is assumed that images can be obtained, analyzed, and transmitted to the controller without restricting the control rates. One type of data source used in

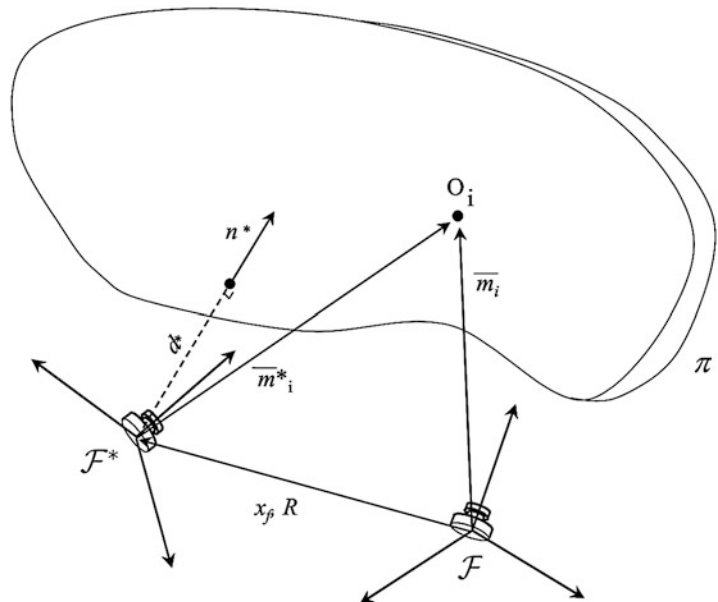
image-based feedback control is feature point which can be identified and tracked among images. With these feature points, the rotation and scaled translation of camera can be estimated, based on homography matrix or essential matrix techniques. In the following, the homography matrix for four coplanar points and essential matrix for eight non-coplanar feature points are introduced briefly.

Homography Matrix

Suppose that there are four coplanar but non-collinear feature points in a plane π as illustrated in Fig. 1. It is reasonable to suppose that feature points lie on a plane, which does not reduce the generality of the results. If there does not exist four available target feature points, then methods like virtual parallax algorithm can be exploited to create virtual planes from non-planar feature points.

In Fig. 1, $\mathcal{F}$ represents the coordinate frame which is affixed to the current camera pose, while the stationary coordinate frame $\mathcal{F}^*$ is affixed to the constant expected camera position and orientation. The Euclidean coordinates of feature points O_i are given by $\bar{m}_i(t) \in \mathbb{R}^3$ and $\bar{m}_i^*(t) \in \mathbb{R}^3$ in frames $\mathcal{F}$ and $\mathcal{F}^*$, respectively

Image-Based Formation Control of Mobile Robots, Fig. 1 Geometric relationships between coordinate frames



$$\begin{aligned}\bar{m}_i &= [x_i(t) \quad y_i(t) \quad z_i(t)]^T \\ \bar{m}_i^* &= [x_i^* \quad y_i^* \quad z_i^*]^T\end{aligned}\quad (1)$$

where $x_i(t)$, $y_i(t)$, and $z_i(t) \in \mathbb{R}$ stand for the Euclidean coordinates of the i th feature point and x_i^* , y_i^* , and $z_i^* \in \mathbb{R}$ are Euclidean coordinates of the corresponding feature points $\mathcal{F}^*$. From the direct geometry relationship, it can be found that

$$\bar{m}_i = x_f + R\bar{m}_i^* \quad (2)$$

where $R(t) \in SO(3)$ is a rotation matrix to describe the orientation of frame $\mathcal{F}^*$ with respect to $\mathcal{F}$. $x_f \in \mathbb{R}$ represents the translation from these two frames, and it is expressed in the frame $\mathcal{F}$. Then, $\bar{m}_i$ and $\bar{m}_i^*$ can be normalized as

$$m_i = [m_{xi} \quad m_{yi} \quad 1]^T = \begin{bmatrix} \frac{x_i}{z_i} & \frac{y_i}{z_i} & 1 \end{bmatrix}^T \quad (3)$$

$$m_i^* = [m_{xi}^* \quad m_{yi}^* \quad 1]^T = \begin{bmatrix} \frac{x_i^*}{z_i^*} & \frac{y_i^*}{z_i^*} & 1 \end{bmatrix}^T \quad (4)$$

Then, plugging Equations (3) and (4) into Equation (2), the following relationship between $m_i(t)$ and m_i^* can be attained as

$$m_i = \frac{z_i^*}{z_i} \left(R + \frac{x_f}{d^*} n^{*T} \right) m_i^* \quad (5)$$

where $\alpha_i(t) \in \mathbb{R}$ can be denoted as $\alpha_i(t) = \frac{z_i^*}{z_i}$ and Euclidean homography matrix $H(t) \in \mathbb{R}^{3 \times 3}$ is $H(t) = R + \frac{x_f}{d^*} n^{*T}$. The matrix $H(t)$ includes the translation vector $x_f(t)$, the distance $d^* \in \mathbb{R}$ from the origin of $\mathcal{F}^*$ to the plane π , the rotation matrix R , and $n^* \in \mathbb{R}^3$ representing the unit normal vector to the plane π . In the image coordinate frame, each feature point O_i has the pixel coordinate $p_i(t) \in \mathbb{R}^3$, $p_i^* \in \mathbb{R}^3$ expressed in the current and expected image as

$$p_i = [u_i \quad v_i \quad 1]^T \quad p_i^* = [u_i^* \quad v_i^* \quad 1]^T \quad (6)$$

and $u_i(t)$, $v_i(t)$, u_i^* , and v_i^* belong to $\mathbb{R}$. According to the pinhole camera model,

$$p_i = A m_i \quad p_i^* = A m_i^* \quad (7)$$

$$A = \begin{bmatrix} \alpha & -\alpha \cot \varphi & u_0 \\ 0 & \frac{\beta}{\sin \varphi} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

where $A \in \mathbb{R}^{3 \times 3}$ is the constant camera calibration matrix, and $u_0, v_0 \in \mathbb{R}$ are the pixel coordinates of image center. $\alpha, \beta \in \mathbb{R}$ stand for the product of camera scaling factors and the focal length, and $\varphi \in \mathbb{R}$ is the skew angle between the camera axes. Usually, this angle is small and can be neglected in some occasions. Further, (5) can be expressed with p_i^* as

$$p_i = \alpha_i (A H A^{-1}) p_i^* \quad (9)$$

The projective homography of the camera can be described by (9). With the four feature points, the Euclidean homography matrix H can be solved, and the relative pose between two frames is also able to be obtained with H .

Essential Matrix

Assume that there are at least eight points and they have been already matched by some image processing methods. It is not necessary for these

eight matched feature points to be coplanar. Instead, no four feature points are coplanar in this case. With such eight feature points, another camera pose estimation method based on the essential matrix can be derived as follows.

Similarly, a pair of matched feature points m_j^* and m_j from two frames $\mathcal{F}$ and $\mathcal{F}^*$ can be represented as

$$m_j(t) = \begin{bmatrix} x_j(t) & y_j(t) & 1 \\ z_j(t) & z_j(t) & 1 \end{bmatrix}^T$$

$$m_j^* = \begin{bmatrix} x_j^* & y_j^* & 1 \\ z_j^* & z_j^* & 1 \end{bmatrix}^T \quad (10)$$

Also, the coordinates of each point in the images are related by

$$z_j m_j = z_j^* R m_j^* + x_f \quad (11)$$

Let the following $[x]_{\times}$ define a skew-symmetric matrix so that $[x]_{\times} m = x \times m$, where $\times$ here denotes the cross product between two vectors.

$$[x]_{\times} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \quad (12)$$

where x_1, x_2 , and $x_3 \in \mathbb{R}$ are the elements of the vector $x = [x_1, x_2, x_3]^T$. Pre-multiply Equation (11) by $[x_f]_{\times}$, it can be found that

$$z_j [x_f]_{\times} m_j = z_j^* [x_f]_{\times} R m_j^* \quad (13)$$

Note that $z_j m_j^T [x_f]_{\times} m_j = 0$ and $z_j m_j (x_f \times m_j) = 0$. Thus,

$$z_j^* m_j^T [x_f]_{\times} R m_j^* = m_j^T E m_j^* = 0 \quad (14)$$

The matrix E here is the essential matrix and can be denoted as $E = [x_f]_{\times} R$. If these eight feature points or more points are available, the essential matrix E can be estimated using linear algebra methods. Then the transformation matrix between two camera frames is solved with the matrix E .

Graph Theory

In this section, we present some basic graph theory concepts following Diestel (2000) that are essential in the study of multi-robot systems.

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ represent a graph and $\mathcal{V} \in \{1, \dots, N\}$ be the set of vertices. The set of edges is denoted as $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Assume that there is no self-loops in the graph, i.e., $(i, i) \notin \mathcal{E}$. $\mathcal{N}_i = \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}\}$ denotes the neighborhood set of vertex i . In a directed graph $\mathcal{G}$, $(i, j) \in \mathcal{E}$ denotes that the node j can obtain information from the node i , but not necessarily vice versa. $\mathcal{G}$ contains a directed spanning tree if there is a node that can reach other nodes via a directed path. $\mathcal{G}$ is undirected if for $(i, j) \in \mathcal{E}$, edge $(j, i) \in \mathcal{E}$. Let $\bar{\mathcal{G}} = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$ represent a graph of a network with one leader and N followers, where $\bar{\mathcal{E}} \subseteq \bar{\mathcal{V}} \times \bar{\mathcal{V}}$, $\bar{\mathcal{V}} = \{0, 1, 2, \dots, N\}$, and 0 are associated with the leader. Then, the neighbor set of node i can be denoted as $\bar{\mathcal{N}}_i = \{j \neq i, (j, i) \in \bar{\mathcal{E}}\}$. $A = [a_{ij}] \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix of $\mathcal{G}$, where $a_{ij} > 0$ if and only if $(j, i) \in \mathcal{E}$, else $a_{ij} = 0$. The in-degree and out-degree of node i are $d_i^I = \sum_{j=1}^N a_{ij}$ and $d_i^O = \sum_{j=1}^N a_{ji}$, respectively. Then, a graph is balanced if $d_i^I = d_i^O$ for all i . $L \triangleq D - A \in \mathbb{R}^{N \times N}$ is called the Laplacian matrix of $\mathcal{G}$, where $D = [d_{ij}] \in \mathbb{R}^{N \times N}$ is the in-degree matrix with $d_{ij} = 0$, $i \neq j$, and $d_{ii} = \sum_{j=1}^N a_{ij}$. The matrix L is symmetric for an undirected graph, while for a directed one, it is not necessarily symmetric. In a leader-following network, $H \triangleq L + B \in \mathbb{R}^{N \times N}$ represents the information-exchange matrix, where B is a diagonal matrix whose i th diagonal element is a_{i0} with $a_{i0} > 0$ if $(0, i) \in \bar{\mathcal{E}}$ and $a_{i0} = 0$, otherwise.

Consider several examples for undirected and directed graphs in Figs. 2, 3, and 4. The graph in Fig. 2 is connected and undirected, where the adjacency matrix and Laplacian matrix are symmetric. In contrast, the graph in Fig. 3 is directed with a spanning tree from each node, where the adjacency matrix and Laplacian matrix are not symmetric. Figure 4 depicts the directed graph with a spanning tree from the leader 0, where H is not symmetric.

Cooperative State Estimation

The cooperative state estimation is crucial in achieving robot formation control (Cook et al. 2012). Usually in multi-robot control process, all robots are only equipped with limited sensors like cameras, which can only detect and measure part of relative state information, while the rest of information is measured by other robots. It is mainly because these sensors have limited measurement ranges both in depth and angle, which makes it nearly impossible for robots to observe states of all other ones. In order for a robot to have a precise estimation of the states in the robot group, distributed cooperative state estimation methods are needed.

In recent years, there have been several methods proposed to solve this problem. In general, they can be categorized into Bayesian-based approaches, non-Bayesian-based approaches, and distributed Kalman filter-based approaches. In most Bayesian-based methods, e.g., in Kwon and Hwang (2019), the prior knowledge like robot model, initial states, and multi-robot topology structure is always integrated with the posterior information including measurements from sensors, transmitted data from other robots. With this information, Bayesian-based schemes can be employed in the multi-robot system to obtain more precise state estimation.

However, under certain conditions where prior knowledge is not available, non-Bayesian-based approaches will become effective and suitable. For example, sometimes, we may know little about the system dynamics of the robot, environment information, and network topology among multiple robots. In Nedic et al. (2017), a distributed non-Bayesian learning algorithm is proposed for multiple robots with complicated network structure, and it can ensure that all agents in the network agree on the hypothesis that best explains the measurement data. Also, a nonasymptotic, explicit, and geometric convergence rate is provided. In Mateos and Giannakis (2012), a distributed recursive least squares method is presented to solve the cooperative estimation problem for multiple robots with ad

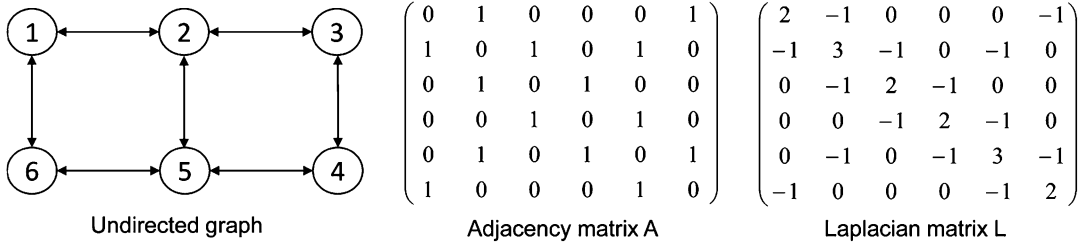


Image-Based Formation Control of Mobile Robots, Fig. 2 Example of an undirected graph

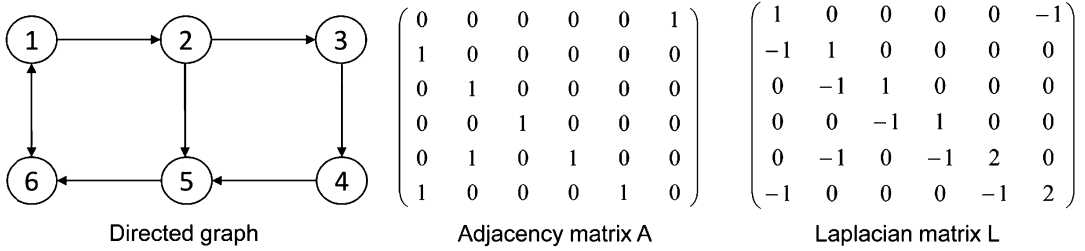


Image-Based Formation Control of Mobile Robots, Fig. 3 Example of a directed graph

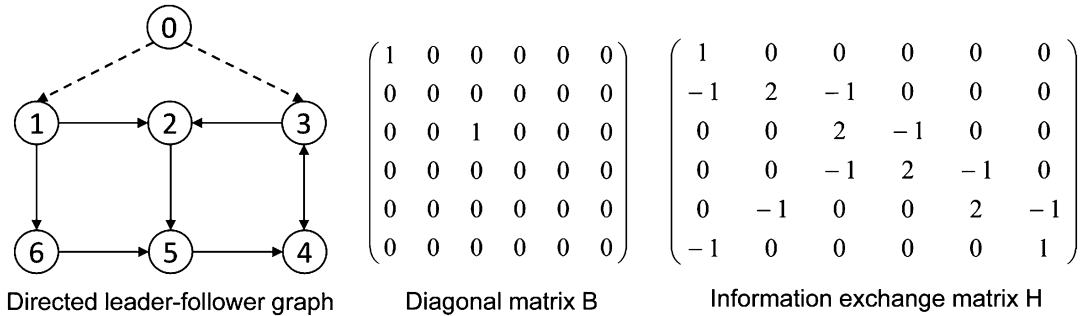


Image-Based Formation Control of Mobile Robots, Fig. 4 Example of a directed leader-following graph

hoc wireless sensors. The distributed iteration laws are attained by minimizing the exponentially weighted least squares cost with the alternating minimization algorithm.

In addition, there are works reported on cooperative estimation using distributed Kalman filter. In Carli et al. (2008), distributed Kalman filters are designed for consensus control problems. These presented filters can estimate the states of target robots and drive the states of all robots to achieve final consensus goal together. Also, in Tu and Sayed (2012), diffusion strategies are utilized for cooperative state estimation. It is shown that diffusion network has a faster convergence rate and can reach lower mean square deviation than consensus networks. The mean square stability of

diffusion network is not sensitive to the provided weights, either. As for Wang et al. (2017), the cases where the process model of interest is uncertain to all robots are discussed. With the proposed methods in these papers, all local robots can identify the underlying model and estimate the expected states.

Cooperative Formation Control

Over the past decades, vision-based cooperative formation control has attracted considerable attention due to the low cost and rich information that cameras can provide in comparison with other traditional sensors (e.g., laser range finders,

radar, sonars, etc.). In a multi-robot system, position and orientation measurements are typically required in cooperative formation control. One observation is that the global position system may not be available in certain environments, e.g., indoor and underwater situations. The error drift is usually inevitable when using inertial measurement units, which leads to inaccuracy or even failure in achieving robot formation. LIDAR sensors have also been widely utilized to provide navigation and localization, while they are expensive.

In contrast, rather than using the aforementioned sensors, cameras enabling a reach data set in a relatively cheap and versatile manner attract increasingly attention to recast the navigation and control problem in terms of the imaged-based vision feedback. Solving the cooperative formation control problem exclusively with onboard vision sensors is challenging, because cameras provide the view angles to the other robots but not the distances or positions/velocities as required in Egerstedt and Hu (2001), Balch and Arkin (1998), Das et al. (2002), Paley and Leonard (2007), Sun et al. (2009), and Ren (2010). The image-based visual feedback relies on the features that are extracted from images, and controllers are derived via image features.

There have been a few works on vision-based formation control in Das et al. (2002), Vidal et al. (2004), Moshtagh et al. (2009), and Mariottini et al. (2009). The omnidirectional cameras are utilized in Das et al. (2002) and Vidal et al. (2004) for multi-robot formation control via an input-output feedback linearization. In particular, a centralized control framework is proposed in Das et al. (2002) for vision-based leader-follower formation control. Both camera-intrinsic calibration parameters and vertical pose of the camera are supposed a priori known to compute both the bearing and distance. Vidal et al. (2004) further introduced a motion segmentation technique to achieve formation of nonholonomic mobile robots with calibrated vision sensors. To remove the distance measurement, the vision-based localization problem is introduced in Mariottini et al. (2009) where the leader adopts an extended Kalman filtering to localize its followers, and computes the control inputs and

guides the formation in a centralized fashion. A visual-based distributed control architecture is introduced in Moshtagh et al. (2009) where the proposed controllers are coordinate-free and rely on only measurements of bearing and optical flow rather than distance measurements or communication of heading information among neighbors.

Notice that the aforementioned results assume that the robots have omnidirectional (or pancontrolled) cameras and that the linear and angular velocities of leader are either communicated to or estimated by followers. However, when communication between robots is absent and/or their onboard camera sensors have limited capabilities (e.g., limited range and angle of view), then the robots can typically stay connected if and only if the leader is always visible to followers. Hence, the latter specification imposes a set of visibility constraint. A related work is Morbidi et al. (2011) that aims to enable groups of mobile robots to visually maintain formations in the absence of communication. Recently, formation control under visibility constraints is studied in Panagou and Kumar (2009) for multi-robots with unicycle kinematics in known obstacle environments via vision-based formation control.

Summary and Future Directions

In contrast to existing coordination works in multi-robot systems based on either position or distance measurements, the image-based cooperative estimation and formation control problems are considered. Since the onboard cameras usually have limited capabilities in practice, each robot can only measure part of the relative measurements within a formation of robots. A consensus-based method can be leveraged to synthesize the cooperative state estimation algorithm such that the measurements made by each individual robot are integrated to obtain a complete description of the formation. This method can be applied to vision-based cooperative formation control of multiple mobile robots in the absence of global reference frames.

Achieving distributed coordinated control of real-world, vision-based multi-robot systems is a

difficult problem because of the associated sensing and communication constraints, complexity, and uncertainties (e.g., field of view (FOV) constraint, camera frame rate limit, image noise, disturbances, unmodeled dynamics, propagation of errors, and dynamically changing topologies, etc.). It is an important issue due to its wide range of potential applications in surveillance and search, highway traffic monitoring, intelligent transportation, environment monitoring, and unmanned exploration of dangerous areas. Vision-based robot systems have enormous potential to become the best among man-made coordinated autonomous systems. At present, however, sensing, communication, and coordination complexities limit the applications.

The emerging research area of image-based cooperative estimation and control is far from being fully explored, and many problems still remain unsolved. One common feature in the existing works is that the underlying graph is time-invariant and undirected. This may not be valid in practical tasks as the onboard camera of each robot has limited sensing abilities and the vision may be blocked or lost in practical dynamical environments. As a result, the interaction network is modeled as dynamic and directed information-exchange topologies. In such a situation, the cooperative state estimation and cooperative control problems would become much more complicated.

Another future direction is time-varying formation tracking control of multi-robot systems. The existing cooperative formation control works assume that the scale and pattern of the formation is always fixed, while in real robot applications, e.g., source seeking or target enclosing, the formation of a team of robots may change as time evolves and is required to track a trajectory to perform tasks. In such scenarios, time-varying formation tracking problems arise where robots can not only keep the time-varying formation but also track the trajectory. New distributed tools are demanded to handle this case.

In addition to cooperative estimation and formation control, other practical tasks required to accomplish with visual-based measurements include imaged-based target tracking,

rendezvous, swarming, mapping, etc. To achieve the missions, additional requirements on the sensing, communication, and computation complexities have to be taken into account, especially in an unknown dynamic environment to avoid collisions among robots and obstacles.

Cross-References

- [Flocking in Networked Systems](#)
- [Image-Based Robot Control](#)
- [Information-Based Multi-Agent Systems](#)
- [Multi-Vehicle Routing](#)
- [Robot Visual Control](#)

Bibliography

- Carli R, Chiuso A, Schenato L, Zampieri S (2008) Distributed Kalman filtering based on consensus strategies. *IEEE J Sel Areas Commun* 26(4):622–633
- Cook J, Hu G (2012) Vision-based triangular formation control of mobile robots. In: Chinese control conference, Hefei, pp 5146–5151
- Cook J, Hu G, Feng Z (2012) Cooperative state estimation in vision-based robot formation control via a consensus method. In: Chinese control conference, Hefei, pp 6461–6466
- Balch T, Arkin R (1998) Behavior-based formation control for multi-robot systems. *IEEE Trans Robot Autom* 14(6):926–939
- Das AK, Fierro R, Kumar V, Ostrowski JP, Spletzer J, Taylor CJ (2002) A vision-based formation control framework. *IEEE Trans Robot Autom* 18(5):813–825
- Diestel R (2000) Graph theory. Springer, New York
- Egerstedt M, Hu X (2001) Formation constrained multi-agent control. *IEEE Trans Robot Autom* 17(6):947–951
- Kwon C, Hwang I (2019) Sensing-based distributed state estimation for cooperative multiagent systems. *IEEE Trans Autom Control* 64(6):2368–2382
- Mariottini GL, Morbidi F, Prattichizzo D, Valk NV, Michael N, Pappas G, Daniilidis K (2009) Vision-based localization for leader–follower formation control. *IEEE Trans Robot* 25(6):1431–1438
- Mateos G, Giannakis GB (2012) Distributed recursive least squares: stability and performance analysis. *IEEE Trans Sig Process* 60(7):3740–3754
- Morbidi F, Bullo F, Prattichizzo D (2011) Visibility maintenance via controlled invariance for leader–follower vehicle formations. *Automatica* 47(5):1060–1067
- Moshtagh N, Michael N, Jadbabaie A, Daniilidis K (2009) Vision-based, distributed control laws for motion coordination of nonholonomic robots. *IEEE Trans Robot* 25(4):851–860

- Nedic A, Olshevsky A, Uribe CA (2017) Fast convergence rates for distributed non-Bayesian learning. *IEEE Trans Autom Control* 62(11):5538–5553
- Paley RSD, Leonard NE (2007) Stabilization of planar collective motion: all-to-all communication. *IEEE Trans Autom Control* 52(5):811–824
- Panagou D, Kumar V (2009) Cooperative visibility maintenance for leader-follower formations in obstacle environments. *IEEE Trans Robot* 30(4):831–834
- Ren W (2010) Consensus tracking under directed interaction topologies: algorithms and experiments. *IEEE Trans Control Syst Technol* 18(1):230–237
- Sun D, Wang C, Shang W, Feng G (2009) A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations. *IEEE Trans Robot* 25(5):1074–1086
- Tu S, Sayed A (2012) Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. *IEEE Trans Sig Process* 60(12):6217–6234
- Vidal R, Shakernia O, Sastry S (2004) Following the flock: distributed formation control with omnidirectional vision-based motion segmentation and visual servoing. *IEEE Robot Autom Mag* 11(4):14–20
- Wang S, Ren W, Chen J (2017) Fully distributed dynamic state estimation with uncertain process models. *IEEE Trans Control Netw Syst* 5(4):1841–1851

Image-Based Robot Control

Chien Chern Cheah and Shangke Lyu
School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore,
Singapore

Abstract

Humans are able to sense and respond to changes in the real world without an exact knowledge of sensory-to-motor transformation. By using visual feedback, we can pick up various new objects or tools and manipulate them easily to perform various visually guided tasks without an accurate knowledge of the kinematics and dynamics. This basic ability gives us a high degree of flexibility in dealing with unforeseen changes in the real world. Inspired by this natural phenomenon, much progress has been obtained in vision-based control of robots. This entry introduces the basic concepts in image-based robot control from a system control theory point of view.

Keywords

Image based control · Robot control · Adaptive control

Introduction

Vision-based control or visual servoing has been extensively used in many modern control applications such as visual-based grasping and assembly in factory automation, autonomous navigation and micro-manipulation. In vision-based control problems, the control inputs are generated based on visual feedback information obtained from a vision system which typically consists of cameras and image processing algorithms. In general, visual servoing methods can be classified into two main categories including position/pose-based visual servoing (PBVS) and image-based visual servoing (IBVS) (Hutchinson et al. 1996). For PBVS, the object of interest is the 3D position or pose parameters of the target, and the task requirements are defined in Cartesian space. In this approach, the exact camera models are required to determine the position or orientation information in Cartesian space based on the visual information obtained from the vision system in image space. In IBVS, the task requirements are directly defined in image space. Since the image error information is directly used to regulate the control input, IBVS methods are more robust to camera calibration errors. In robot visual servo control schemes, the cameras are either placed at fixed positions, which is named as eye-to-hand configuration or mounted on the robots, which is named as eye-in-hand configuration (Hutchinson et al. 1996).

In image-based control, the control objective can be expressed as a regulation or tracking problem of the visual feature error defined as $\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_d$, where $\Delta \mathbf{x}$ denotes the vector of visual feature error, $\mathbf{x}$ denotes the visual feature, and $\mathbf{x}_d$ denotes the desired visual feature or trajectory in image space. Since the errors are specified in image space and control inputs of the robots are issued at joint level or actuator level, a transformation matrix is needed to convert the error in

image space into input commands in joint space. In robotics, Jacobian matrix plays the role of transforming the image error to joint commands. Let $\dot{\mathbf{q}}$ denote the joint velocity in joint space and $\dot{\mathbf{x}}$ denote the velocity of the visual feature point in image coordinate; the relationship between $\dot{\mathbf{q}}$ and $\dot{\mathbf{x}}$ is defined as (Hutchinson et al. 1996):

$$\dot{\mathbf{x}} = \mathbf{J}_s \mathbf{J}_r \dot{\mathbf{q}} = \mathbf{J} \dot{\mathbf{q}} \quad (1)$$

where $\mathbf{J}$ denotes the overall Jacobian matrix from joint space to image space, $\mathbf{J}_r$ denotes the Jacobian matrix from joint space to Cartesian space, and $\mathbf{J}_s$ denotes the image Jacobian matrix from Cartesian space to image space.

The joint commands for image-based robot control can be designed in two ways depending on the architectures of the robot control systems. Most industrial robot systems have a closed architecture control system where the user inputs can only be specified as position or velocity commands. A joint-space controller or inner controller is used to control the robot, but the controller is fixed, and the torque or voltage control inputs to the motors are not accessible by the users. In this case, the image-based controller is added as an external feedback loop as illustrated in Fig. 1. For robot systems with an open-architecture control system, the inputs can be specified as input torques or voltage to the motors or actuators, and the image-based feedback controller can be designed directly.

Traditional visual servoing techniques are mostly defined as the reference joint velocity as illustrated in Fig. 1. A typical visual servo controller is given as (Hutchinson et al. 1996):

$$\dot{\mathbf{q}}_r = -k_p \mathbf{J}^+ \Delta \mathbf{x} \quad (2)$$

where $\dot{\mathbf{q}}_r$ denotes the reference velocity command in joint space and $\mathbf{J}^+$ denotes the pseudo-inverse of $\mathbf{J}$. The pseudo-inverse of the Jacobian matrix, $\mathbf{J}^+$, is used to transform the servo error from image space to joint space. The reference joint velocity command is then used as the reference inputs of the inner loop to control the robot, and thus the accuracy of the visual servoing also relies on the performance of the inner controller. Such controller is also commonly named as kinematic visual servo controller as the dynamics of the robots or the inner control loop are not taken into consideration in the design and analysis of the closed-loop control systems. The advantage of the kinematic visual servoing approach is that it can be easily implemented on most industrial robotic systems with closed control architecture. It can also be shown that the external loop is locally stable in presence of camera calibration errors.

Dynamic Image-Based Controller

One issue in the design of the kinematic visual servo controller is that the effects of robot dynamics or the interactions with the inner control loop are neglected, and therefore the stability of the overall system is not analyzed. For robots with open-architecture control system, the image-based controller can be designed directly as the torque or voltage inputs as shown in Fig. 2, taking into consideration the effects of the robot dynamics.

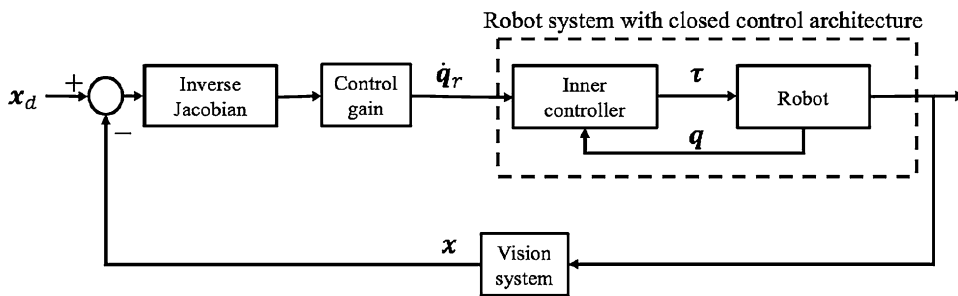
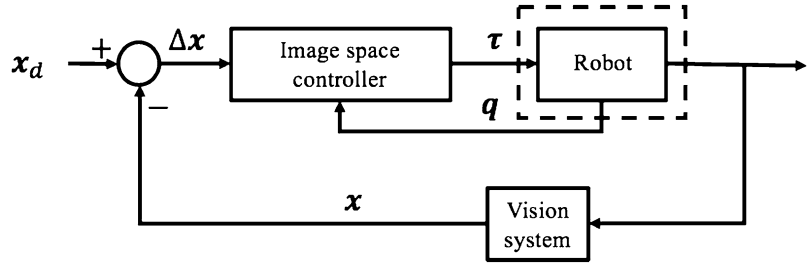


Image-Based Robot Control, Fig. 1 Vision based control with closed control architecture

Image-Based Robot Control, Fig. 2 Image based control with open control architecture



The dynamic model of a robot manipulator is given as (Arimoto 1996):

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \left[\frac{1}{2}\dot{\mathbf{M}}(\mathbf{q}) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \right] \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (3)$$

where $\mathbf{M}(\mathbf{q})$ is the manipulator inertia matrix which is symmetric and positive definite; $[\frac{1}{2}\dot{\mathbf{M}}(\mathbf{q}) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})]\dot{\mathbf{q}}$ denotes the centripetal and Coriolis torques, where $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})$ is skew-symmetric; and $\mathbf{g}(\mathbf{q})$ represents the vector of gravitational torques, $\boldsymbol{\tau}$ stands for the vector of control input.

Approximate Jacobian Setpoint Control

A simplest form of dynamic image-based robot control is the proportional-derivative (PD) controller designed in torque control mode as follows: (Cheah et al. 2003):

$$\boldsymbol{\tau} = -\hat{\mathbf{J}}^T \mathbf{K}_p \text{sat}(\Delta \mathbf{x}) - \mathbf{K}_v \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (4)$$

where $\hat{\mathbf{J}}$ denotes the approximate Jacobian matrix, $\text{sat}(\Delta \mathbf{x})$ denotes a saturated function, and $\mathbf{K}_v$ and $\mathbf{K}_p$ denote the constant gain matrices which are positive definite. The basic idea is to form a saturated virtual spring force $\mathbf{K}_p \text{sat}(\Delta \mathbf{x})$ in image space and convert it to torque in joint space by using the transpose Jacobian matrix. The role of the saturated function is to bound the visual feedback error $\Delta \mathbf{x}$ when it is large. The basic idea stems from visually guided reaching tasks of human where only the directional information is used when the position error is large. That is, the magnitude is not important at the beginning stage and thus can be saturated, and hence only the directional information is used. The magnitude of the position error is only

monitored when the target is nearer. The second term in (4) is a joint-spring damping term, and the last term is used to compensate gravitational torques. The stability of the closed-loop robot control system shown by (3) and (4) can be guaranteed by using Lyapunov method, and the details can be found in Cheah et al. (2003).

Adaptive Jacobian Control

The PD controller with gravity compensator in (4) is used for setpoint task, and the estimated parameters of the approximate Jacobian matrix are not updated upon to improve the performance. For tracking control problem in presence of uncertain parameters, adaptive control approach (Slotine and Li 1987) is widely adopted, in which the estimated parameters are adjusted online by using the feedback error. The key challenges in developing image-based adaptive robot control (Cheah et al. 2006) are that the kinematic parameters are not linearly parameterized in task-space dynamics and hence cannot be updated together with the dynamic parameters. The depth, kinematic, and dynamic parameters have to be updated separately to preserve stability.

Let $\mathbf{J}_s = \mathbf{Z}^{-1}\mathbf{L}$ in (1) where $\mathbf{Z}$ denoted as a diagonal matrix that contains the depth information of the feature points with respect to the camera image frame and $\mathbf{L}$ a Jacobian matrix. Note that both $\mathbf{Z}\dot{\mathbf{x}}$ and $\mathbf{J}_m\dot{\mathbf{q}}$ where $\mathbf{J}_m = \mathbf{L}\mathbf{J}_r$ are linear in sets of kinematic and camera parameters such that $\mathbf{J}_m\dot{\mathbf{q}} = \mathbf{Y}_k(\mathbf{q}, \dot{\mathbf{q}})\boldsymbol{\theta}_k$ and $\mathbf{Z}\dot{\mathbf{x}} = \mathbf{Y}_z(\mathbf{q}, \dot{\mathbf{x}})\boldsymbol{\theta}_z$ where $\mathbf{Y}_k(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{Y}_z(\mathbf{q}, \dot{\mathbf{x}})$ denote the kinematic regressor matrix and depth regressor matrix, respectively, and $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_z$ denote the unknown constant kinematic parameter vector and the unknown constant depth parameter vector, respectively.

An adaptive Jacobian image-based controller is given as:

$$\begin{aligned} \tau = & -\hat{\mathbf{J}}_m^T(\hat{\boldsymbol{\theta}}_k)\hat{\mathbf{Z}}^{-1}(\hat{\boldsymbol{\theta}}_z)(\mathbf{K}_v\Delta\dot{\mathbf{x}} + \mathbf{K}_p\Delta\mathbf{x}) \\ & + \mathbf{Y}_d(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\hat{\boldsymbol{\theta}}_d \end{aligned} \quad (5)$$

where

$$\dot{\mathbf{q}}_r = \hat{\mathbf{J}}_m^+(\hat{\boldsymbol{\theta}}_k)\hat{\mathbf{Z}}(\hat{\boldsymbol{\theta}}_z)[\dot{\mathbf{x}}_d - \alpha(\mathbf{x} - \mathbf{x}_d)] \quad (6)$$

with $\Delta\dot{\mathbf{x}} = \dot{\mathbf{x}} - \dot{\mathbf{x}}_d$ and $\dot{\mathbf{x}} = \hat{\mathbf{Z}}^{-1}\hat{\mathbf{J}}_m\dot{\mathbf{q}}$ with $\hat{\mathbf{Z}}$ and $\hat{\mathbf{J}}_m$ denoted as the approximate matrices of $\mathbf{Z}$ and $\mathbf{J}_m$, respectively, and $\mathbf{Y}_d(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)$ denotes the dynamic regressor matrix. The update laws to estimate the unknown dynamic, kinematic, and depth parameters, respectively, are:

$$\dot{\hat{\boldsymbol{\theta}}}_d = -\mathbf{L}_d\mathbf{Y}_d(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)(\dot{\mathbf{q}} - \dot{\mathbf{q}}_r) \quad (7)$$

$$\dot{\hat{\boldsymbol{\theta}}}_k = \mathbf{L}_k\mathbf{Y}_k^T(\mathbf{q}, \dot{\mathbf{q}})\hat{\mathbf{Z}}^{-1}(\mathbf{K}_v\Delta\dot{\mathbf{x}} + \mathbf{K}_p\Delta\mathbf{x}) \quad (8)$$

$$\dot{\hat{\boldsymbol{\theta}}}_z = -\mathbf{L}_z\mathbf{Y}_z^T(\mathbf{q}, \dot{\mathbf{x}})\hat{\mathbf{Z}}^{-1}(\mathbf{K}_v\Delta\dot{\mathbf{x}} + \mathbf{K}_p\Delta\mathbf{x}), \quad (9)$$

where $\mathbf{L}_d$, $\mathbf{L}_k$, and $\mathbf{L}_z$ are gain matrices that are symmetric and positive definite. The detailed stability analysis can be found in Cheah and Li (2015).

Separation of Kinematic and Dynamic Control Loops

In kinematic visual servoing, the controller is implemented as an outer feedback control loop as illustrated in Fig. 1, and the stability of the overall system is not analyzed since the external control loop and inner control loop are treated as independent systems. In dynamic visual servoing, the stability of the system can be analyzed directly, but it is not applicable for robots with closed control architecture. One way to overcome these problems is the use of a design approach (Wang 2016), which realizes the separate designs of the kinematic and dynamic control loops while preserving the stability of the overall system when integrating them together.

For the kinematic control loop, the controller is specified as the joint velocity command that can be directly fed into the inner joint servo loop as:

$$\dot{\mathbf{q}}_r = \hat{\mathbf{J}}^+(\hat{\boldsymbol{\theta}}_k)\dot{\mathbf{x}}_d - \hat{\mathbf{J}}^T(\hat{\boldsymbol{\theta}}_k)\alpha\Delta\mathbf{x} \quad (10)$$

where the estimated kinematic parameters are updated as:

$$\dot{\hat{\boldsymbol{\theta}}}_k = \mathbf{L}_k\mathbf{Y}_k^T(\mathbf{q}, \dot{\mathbf{q}})\Delta\mathbf{x}. \quad (11)$$

Let the velocity tracking error for the inner joint servo control loop be defined as $\mathbf{s} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r$, if the inner control loop is designed so that the joint velocity error $\mathbf{s} \in L_2 \cap L_\infty$, then there exist a positive constant l_m such that $\int_0^t \mathbf{s}^T \mathbf{s} d\sigma \leq l_m, \forall t \geq 0$. The detailed stability analysis can be found in Wang (2016).

This design approach can also be used for robots with opened control architecture. In this case, a torque controller can be designed using (10) as:

$$\tau = -\mathbf{K}\mathbf{s} + \mathbf{Y}_d(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\hat{\boldsymbol{\theta}}_d \quad (12)$$

with an update law for dynamic parameters given as:

$$\dot{\hat{\boldsymbol{\theta}}}_d = -\mathbf{L}_d\mathbf{Y}_d(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\mathbf{s}. \quad (13)$$

The details can be found in Wang (2016).

Summary and Future Directions

Image-based robot control is robust to modelling uncertainty and thus releases users from tedious camera calibration and model identification tasks. To integrate with different robot control architectures, the visual servoing commands can be specified either as joint position or velocity reference inputs or as torque control inputs. It is important to consider the interaction between kinematic control loop and dynamic or inner control loop in the design of the image-based control for the robotic systems with closed control

architecture. Although the separation approach provides a way to design and integrate the kinematic control loop and dynamic control loop, the joint velocity tracking error is required to be square-integrable and bounded. Alleviating this constraint can further widen the potential applications of the image-based control theory to robotic systems.

Most industrial robots do not come with external sensors, and vision systems have to be added and deployed according to various applications. As different cameras and configurations lead to different models, it is still difficult for general users to derive the structure of the Jacobian matrices. Advances in machine intelligence have led to the interests in developing learning techniques for robot systems with unknown structures. Neural networks can be used to estimate the Jacobian matrix, but one issue is that it does not generalize well beyond the training data. Some key challenges are the development of neural network Jacobian control methods that achieve better generalization property and the development of theoretical frameworks in understanding the image-based learning control methods.

Recommended Reading

This entry presents a brief overview of image-based robot control. Several good tutorial papers and the references for pioneer work on kinematic visual servoing can be found in Hutchinson et al. (1996) and Chaumette and Hutchinson (2006). The pioneer result on robot controller design based on Lyapunov method (Slotine and Li 1991) can be found in Takegaki and Arimoto (1981) and more details and references for dynamic image-based robot control can be found in Cheah and Li (2015).

Cross-References

- [Lyapunov's Stability Theory](#)
- [Nonlinear Adaptive Control](#)
- [Robot Motion Control](#)
- [Robot Visual Control](#)

Bibliography

- Arimoto S (1996) Control theory of nonlinear mechanical systems. Oxford University Press
- Chaumette F, Hutchinson S (2006) Visual servo control. I. Basic approaches. *IEEE Robot Autom Mag* 13(4): 82–90
- Cheah CC, Li X (2015) Task-space sensory feedback control of robot manipulators, vol 73. Springer, New York
- Cheah CC et al (2003) Approximate Jacobian control for robots with uncertain kinematics and dynamics. *IEEE Trans Robot Autom* 19(4):692–702
- Cheah CC, Liu C, Slotine JJE (2006) Adaptive tracking control for robots with unknown kinematic and dynamic properties. *Int J Robot Res* 25(3):283–296
- Hutchinson S, Hager G, Corke P (1996) A tutorial on visual servo control. *IEEE Trans Robot Autom* 12(5):651–670
- Slotine JJE, Li W (1987) On the adaptive control of robot manipulators. *Int J Robot Res* 6(3):49–59
- Slotine JE, Li W (1991) Applied nonlinear control. Prentice Hall, Englewood Cliffs
- Takegaki M, Arimoto S (1981) A new feedback method for dynamic control of manipulators. *J Dyn Syst Meas Control* 103(2):119–125
- Wang H (2016) Adaptive control of robot manipulators with uncertain kinematics and dynamics. *IEEE Trans Autom Control* 62(2):948–954

Industrial Cyber-Physical Systems

Alf J. Isaksson

ABB AB, Corporate Research, Västerås, Sweden

Abstract

The notion of cyber-physical systems (CPS) is applicable to many different application domains, for example, the transportation systems and the energy systems. This article deals with industrial CPS, which we define as CPS concepts applied to the process and manufacturing industries. Originating from traditional control systems, the industry is already since a number of years undergoing a major shift toward more connected systems, also reflected in various national initiatives such as Industrie 4.0 in Germany, Smart Manufacturing in the

USA, and Intelligent Manufacturing in China. We first give a brief historic perspective on the industrial automation systems and how that is currently evolving into more general CPS. Then some ongoing trends and future challenges are discussed. This includes the shift toward more autonomous systems and challenges such as analysis of CPS including modelling and cyber-physical security.

Keywords

Cyber-physical systems · Manufacturing · Process automation

Introduction

By definition, cyber-physical systems (CPS) are technical systems characterized by the combination of cyber elements such as communication, computing, and control with processes governed by the laws of the physics. In modern society, they can be found in many different domains, such as energy systems, transportation systems, healthcare applications, as well as process and manufacturing industries. See Allgöwer et al. (2019) for a recent paper that provides an extensive overview of several of these domains.

This article will focus its attention to one of the domains, viz., process and manufacturing industries. This domain was arguably the first application to undergo a development in the direction of CPS. Computer-based supervisory control was first deployed already in the late 1950s. Then, with the introduction of the micro-processor, the first distributed control systems (DCS) were developed in the 1970s.

The history of the Internet runs more or less parallel to that of the distributed control systems, with the first networks being developed in the late 1960s and early 1970s. Despite this, the control system network – often referred to as the operational technology (OT) system – was for many years at the manufacturing site kept separate from the office network, usually referred to as the information technology (IT) system.

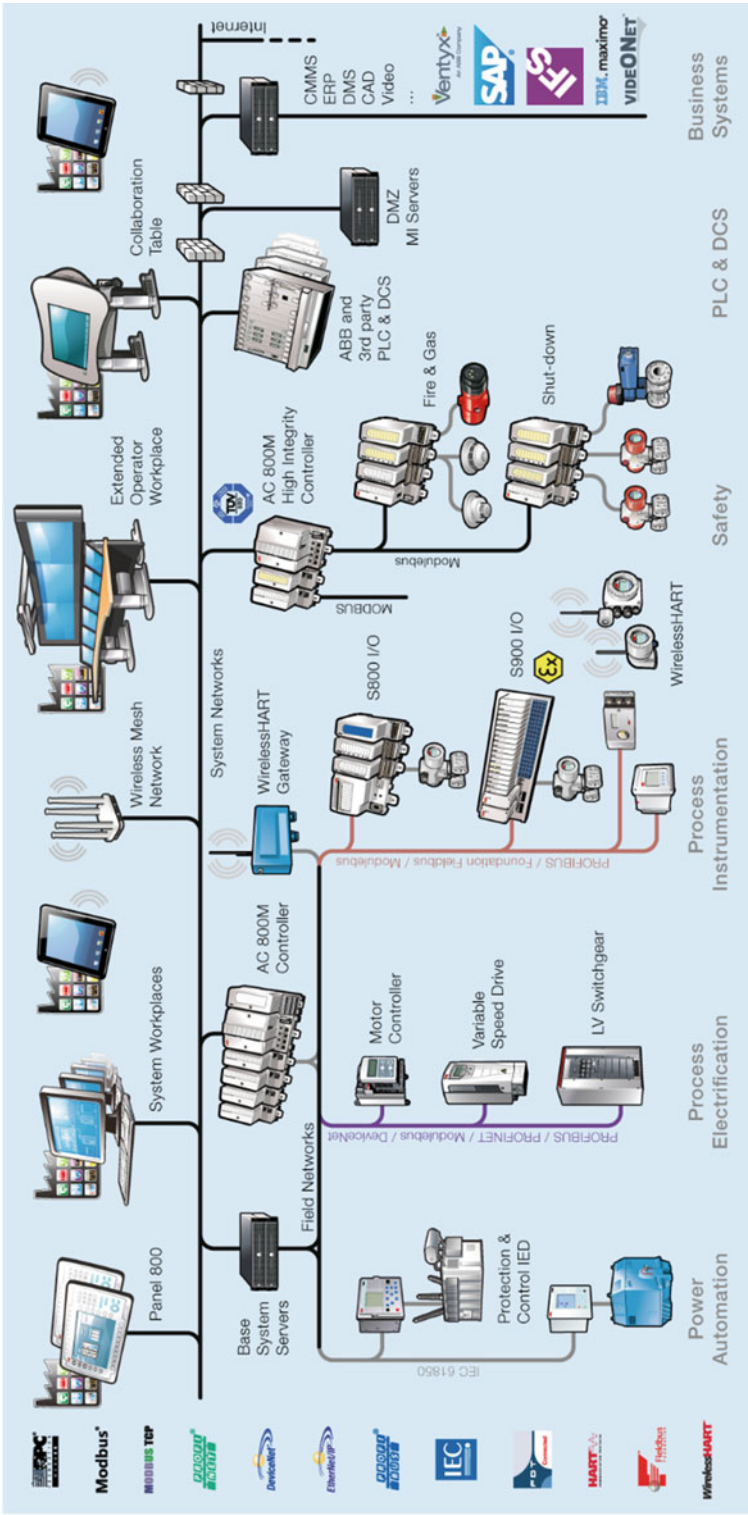
Similarly and simultaneously, there has been the development of mobile telecommunication which for the early generations (1G and 2G) had no real impact on industrial automation. It is basically only in the last decade with 3G and 4G and the introduction of smartphones and mobile apps that a convergence of industrial automation, the Internet, and mobile communication has been noticed.

The trend toward improved connectivity and dramatically increased computational power is now rapidly removing the boundaries between OT and IT. The purpose of this article is to give a brief historic account of the development of industrial CPS, as well as to point to some ongoing and future directions of this progress.

Brief History of Industrial CPS

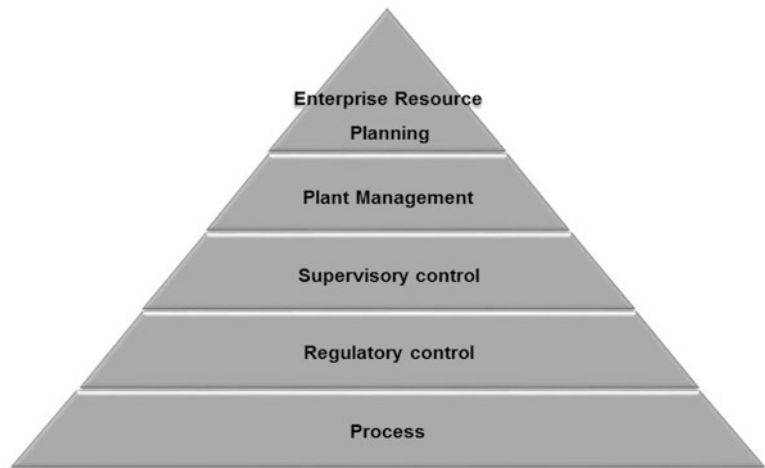
As already mentioned in the introduction, distributed control systems (DCS) were applied in the process industry in the 1970s. In many ways, their architecture has not changed much since. Despite the term “distributed,” they are surprisingly centralized. Typically, measurements of continuous variables, such as temperatures, pressures, flows, etc., are collected via 4–20 mA cables or increasingly with buses such as Foundation Fieldbus or PROFIBUS via I/O units into process computers – often referred to as controllers – that reside in a centralized so-called marshalling room. Then the controllers are connected via a server network to engineering and operator work stations. See Fig. 1 for a typical modern DCS structure.

As indicated to the right in Fig. 1, a modern DCS is integrated with higher levels of the automation hierarchy (see Fig. 2 for the traditional automation pyramid) such as computerized maintenance management systems (CMMS) and enterprise resource planning (ERP). This trend toward being able to access most automation systems from one entry point has led to the introduction of the term Collaborative Process Automation Systems (CPAS) (see the book edited by Hollender 2010). State of the art in many process industries, however, is that the different



Industrial Cyber-Physical Systems, Fig. 1 State-of-the-art distributed control system (ABB's System 800xA – with permission)

Industrial Cyber-Physical Systems, Fig. 2 The traditional automation pyramid



levels of the automation pyramid are still taken care of by different departments using different software, in different geographic locations.

A similar development started even earlier for discrete manufacturing where, starting in the automotive industry, hard-wired relays were replaced by programmable logic controllers (PLC). While a DCS controller usually has both analog and digital inputs and outputs, a PLC instead primarily has digital, i.e., on/off, signals as input and output. However, in recent years, there is a clear convergence where DCS controllers and PLCs have relatively overlapping functionality and performance. PLCs tend to be located nearer to (or integrated into) the machine and thus in fact more distributed. They are often connected to other PLCs in a network and also commonly to a supervisory control and data acquisition (SCADA) system. Hence historically, the hierarchy in Fig. 2 has been equally valid also for manufacturing systems.

Clearly, there is also an increased interplay and integration between CPS for manufacturing and CPS in other domains. For example, Sand and Terwiesch (2013) point out that there is an increasing integration between the process and power automation. In the future, it will be more and more important to take advantage of the fluctuating electricity prices (Mitra et al. 2014), often referred to as industrial demand-side management. Similarly, industrial CPS is of course tightly connected to the transportation systems

since, for example, integrating supply chain and production planning has clear economic benefits (Carlsson et al. 2014).

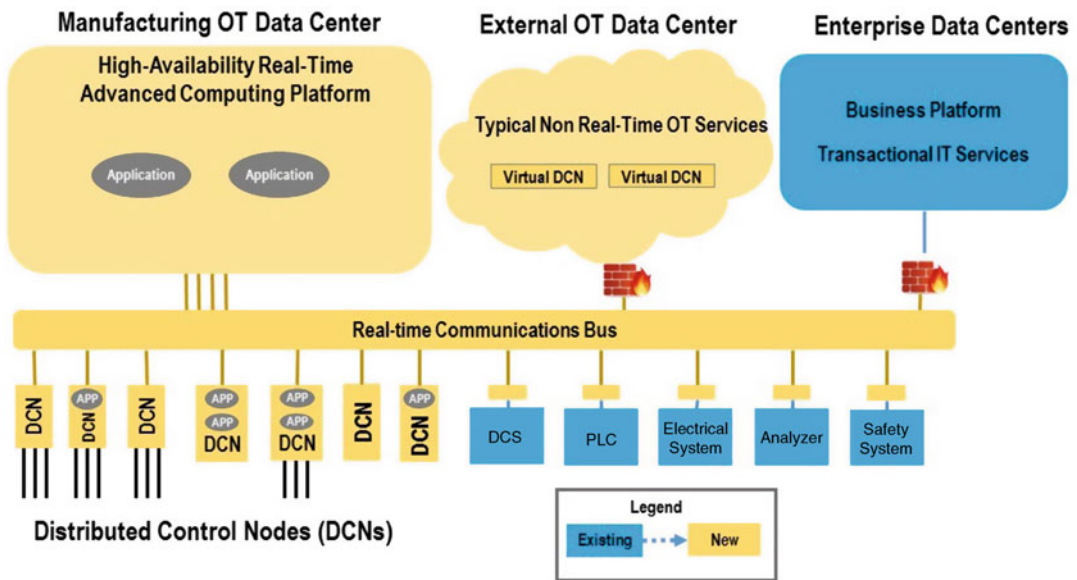
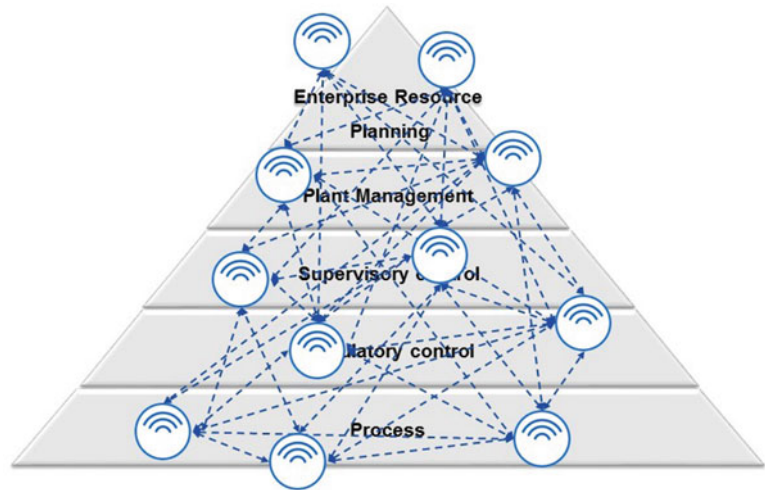
Summary and Future Directions

An immediate impact of the so-called industrial Internet of Things (IIoT) is that measurement devices, for example, vibration measurements for rotating equipment, may be directly connected to Internet via Bluetooth, WiFi, or soon 5G without necessarily going through the DCS. This opens up entirely new possibilities which have mainly so far been utilized for condition monitoring, and diagnostics. Figure 3 shows a vision, presented in, for example, Monostori et al. (2016) and Isaksson et al. (2018), that from a communication perspective the future factory has no hierarchy and that all products and devices are interconnected in a mesh network.

Even though this vision may not be reached in the short term, it is clear that the information flow will not continue to follow the traditional hierarchy of Fig. 2. ExxonMobil presented in 2016 its visions toward the future automation architecture (see Forbes 2016). They promote increased use of open standards for process automation, which has now been further developed into a first version of a standard O-PASTM by The Open Group (2019).

A somewhat simplified picture of the proposed O-PASTM standard architecture is depicted in

Industrial Cyber-Physical Systems, Fig. 3 Vision on communication in the future factory



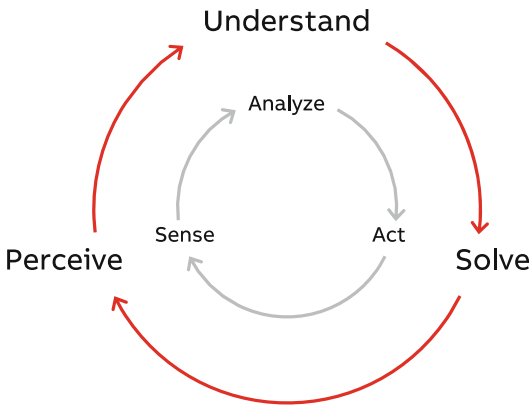
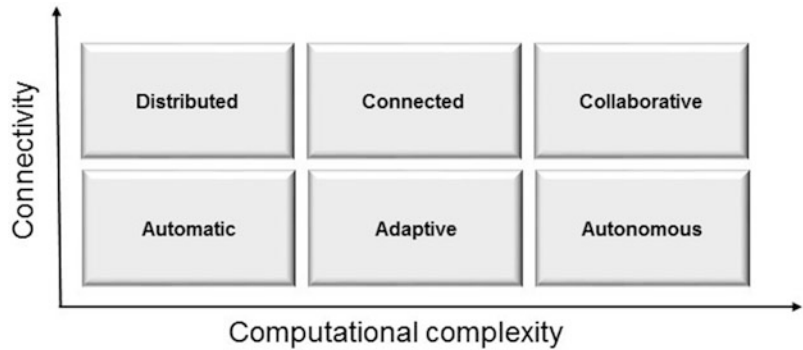
Industrial Cyber-Physical Systems, Fig. 4 Potential future control architecture proposed by The Open Group (with permission)

Fig. 4, with color coding indication that the functionality of today's DCS (and PLC) will be distributed between newly defined distributed control nodes (DCN) and a local operation platform (and of course potentially also to a global cloud). At the lowest level, the connection to the common real-time bus could be through a standardized DCN as suggested by The Open Group. However, an even more disruptive scenario – utilizing local intelligence in the devices (commonly referred to as “edge computing”) instead of using separate

DCNs together with 5G communication – would bring us closer to the Industrie 4.0 vision of Fig. 3. This will, however, require a considerable standardization effort to harmonize both communication and control configurations, since devices will be from many different suppliers.

An interesting classification of CPS is introduced in Allgöwer et al. (2019) which points out that a simultaneous development along with two axes is observed (see Fig. 5). Firstly, along the axis of increased computational complexity,

Industrial Cyber-Physical Systems, Fig. 5 Proposed classification of CPS adopted from Allgöwer et al. (2019)



Industrial Cyber-Physical Systems, Fig. 6 Loops in classical control systems (grey) and autonomous systems (red)

we see a trend toward more autonomous systems. This trend began with the development of self-driving cars, but the concept of autonomous systems is now being discussed also in relation to process industries (Gamer et al. 2020). Secondly, with increasing connectivity, we have cyber-physical systems of systems (CPSoS) which in their most advanced form become collaborative.

While a conventional automation system performs predefined instructions within a limited scope of operation, an autonomous system's feedback loop adds an outer loop. Where a classical control loop can be broken down into the phases of sense, analyze, and act, the autonomy applies similar principles but on a more complex level, dealing also with what is not known or foreseen. This can be described in the steps of perceive,

understand, and solve (see Fig. 6), which all may contain artificial intelligence as one important enabling technology (see Gamer et al. (2020) for a more elaborate discussion).

One major challenge of industrial CPS is how to analyze, in advance, the behavior of the system. As control engineers, we are used to being able to analyze stability and to assess, for instance, whether responses will be oscillatory based on dynamic models of the system. CPS are dynamic and hybrid in nature. Even though there are promising approaches, for example, based on temporal logic, the sheer complexity makes analysis complicated, in particular with, as discussed above, increasing use of machine learning components in the closed loop which do not lend themselves easily to theoretical analysis. The fallback solution is then to verify as much as possible the functionality by simulation. In factory automation, the so-called virtual commissioning is already becoming a standard procedure (Lee and Park 2014), and this will eventually be the case also for process automation with models generated automatically from the topology given in process diagrams (Arroyo et al. 2016).

The challenge is then shifted to reducing the modelling effort of each component. More recently, triggered by the notion of a Digital Twin (Negri et al. 2017; Uhlemann et al. 2017), there is a trend toward using models along the entire life cycle of automation. This will require (first principles) physical modelling for basically all components and systems. A vision would be that every product, machine, or device comes with a stored Digital Twin, which would of

course include simulation models also for all control elements. However, this will inevitably need a major standardization effort on modelling formats.

Cross-References

- [Control Hierarchy of Large Processing Plants: An Overview](#)
- [Programmable Logic Controllers](#)

Recommended Reading

General introductions to cyber-physical systems can be found in, for example, the books Alur (2015) and Lee and Seshia (2011). The already mentioned paper by Allgöwer et al. (2019) gives a very good overview of some of the future challenges to CPS theory. With increased connectivity comes also the risk of cybersecurity attacks. For a recent historic account, see Middleton (2017), and for a survey of academic activity from an automatic control perspective, see Zacchia Lun et al. (2019).

Bibliography

- Allgöwer F, Borges de Sousa J, Kapinski J, Mosterman P, Oehlerking J, Panciatici P, Prandini M, Rajhans A, Tabuada P, Wenzelburger P (2019) Position paper on the challenges posed by modern applications to cyber-physical systems theory. *Nonlinear analysis: hybrid systems*, vol 34, pp 147–165
- Alur R (2015) *Principles of cyber-physical systems*. The MIT Press, Cambridge, London
- Arroyo E, Hoernicke M, Rodriguez P, Fay A (2016) Automatic derivation of qualitative plant simulation models from legacy piping and instrumentation diagrams. *Comput Chem Eng* 92:112–132
- Carlsson D, Flisberg P, Rönqvist M (2014) Using robust optimization for distribution and inventory planning for a large pulp producer. *Comput Oper Res* 44:214–225
- Forbes H (2016) ExxonMobil's quest for the future of process automation. *ARC Insights*
- Gamer T, Hoernicke M, Kloepper B, Bauer R, Isaksson AJ (2020) The autonomous industrial plant – future of process engineering, operations and maintenance. *J Process Control* 88:101–110
- Hollender M (2010) Collaborative process automation systems. *International Society of Automation, Research Triangle Park*
- Isaksson AJ, Harjunoski I, Sand G (2018) The impact of digitalization on the future of control and operations. *Comput Chem Eng* 114:122–129
- Lee CG, Park SC (2014) Survey on the virtual commissioning of manufacturing systems. *J Comput Des Eng* 1(3):213–222
- Lee EA, Seshia SA (2011) *Introduction to embedded systems – a cyber-physical systems approach*, LeeSashia.org
- Middleton B (2017) *A history of cyber security attacks – 1980 to PRESENT*. CRC Press/Taylor & Francis Group, Boca Raton
- Mitra S, Pinto JM, Grossmann I (2014) Optimal multi-scale capacity planning for power-intensive continuous processes under time-sensitive electricity prices and demand uncertainty. Part I: modeling. *Comput Chem Eng* 65:89–101
- Monostori L, Kádár B, Bauernhansl T, Kondoh S, Kumara S, Reinhart G, Sauer O, Schuh G, Sihn W, Ueda K (2016) Cyber-physical systems in manufacturing. *CIRP Ann Manuf Technol* 65:621–641
- Negri E, Fumagalli L, Macchi M (2017) A review of the roles of digital twin in CPS-based production systems. *Proc Manuf* 11:939–948
- Open Group (2019) O-PAS standard, version 1.0: part 1 – technical architecture overview (informative). The Open Group, Reading
- Sand G, Terwiesch P (2013) Closing the loops: an industrial perspective on the present and future impact of control. *Eur J Control* 19:341–350
- Uhlemann TH-J, Lehmann C, Steinhilper R (2017) The digital twin: realizing the cyber-physical production systems for industry 4.0. *Proc CIRP* 61:335–340
- Zacchia Lun Y, D'Innocenzo A, Smarra F, Malavolta I, Di Benedetto MD (2019) State of the art of cyber-physical systems security: an automatic control perspective. *J Syst Softw* 149:174–216

Industrial MPC of Continuous Processes

Mark L. Darby

CMiD Solutions, Houston, TX, USA

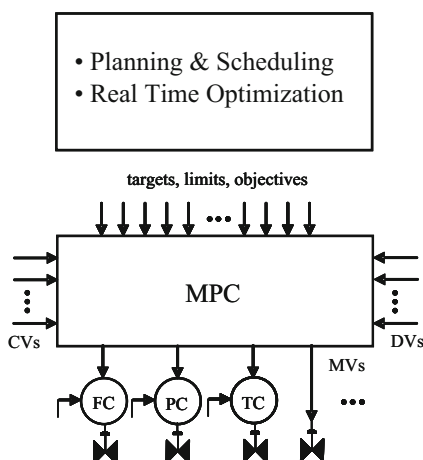
Abstract

Model predictive control (MPC) has become the standard approach for implementing constrained, multivariable control of industrial

continuous processes. These are processes designed to operate around nominal steady-state values, which include many of the important processes found in the refining and chemical industries. The following provides an overview of MPC, including its history, major technical developments, and how MPC is applied today in practice. Current and possible future developments are provided.

Keywords

Model predictive control · Multivariable systems · Constraints · Process testing · Modeling · Process identification



Industrial MPC of Continuous Processes, Fig. 1
Industrial control and decision hierarchy

Introduction

Model predictive control (MPC) refers to a class of control algorithms that explicitly incorporate a process model for predicting the future response of a plant and relies on optimization as the means of determining control action. At each sample interval, MPC computes a sequence of future plant input signals that optimize future plant behavior. Only the first of the future input sequence is applied to the plant, and the optimization is repeated at subsequent sample intervals.

MPC provides an integrated solution for controlling non-square systems with complex dynamics, interacting variables, and constraints. MPC has become a standard in the continuous process industries, particularly in refining and chemicals, where it has been widely applied for over 25 years. In most commercial MPC products, an embedded steady-state optimizer is cascaded to the MPC controller. The MPC steady-state optimizer determines feasible, optimal settling values of the manipulated and controlled variables. The MPC controller then optimizes the dynamic path to optimal steady-state values.

The scope of an MPC application may include a unit operation such as a distillation column or reactor, or a larger scope such as multiple distillation columns, or a scope that combines

reaction and separation sections of a plant in one controller. MPC is positioned in the control and control decision hierarchy of a processing facility as shown in Fig. 1. The variables associated with MPC consist of manipulated variables (MVs), controlled variables (CVs), and disturbance variables (DVs). CVs include variables normally controlled at a fixed value such as a product impurity and those considered constraints, for example, limits related to capacity or safety that may only be sometimes active. DVs are measurements that are treated as feedforward variables in MPC. The manipulated variables are typically setpoints of underlying PID controllers but may also include valve position signals. Most of the targets and limits are local to the MPC, but others come directly from real-time optimization (if present), or indirectly from planning/scheduling, which are normally translated to the MPC in an open-loop manner by operations personnel.

Linear and nonlinear model forms are found in industrial MPC applications; however, the majority of the applications continue to rely on a linear model, identified from data generated from a dedicated plant test. Nonlinearities that primarily affect system gains are often adequately controlled with linear MPC through gain scheduling or by applying linearizing static transformations. Nonlinear MPC applications tend to be reserved for those applications where nonlinearities

ties are present in both system gains and dynamic responses and the controller must operate at significantly different targets. Most nonlinear MPC applications today have fewer MVs and CVs than linear MPC applications.

Origins and History

MPC has its origins in the process industries in the 1970s. 1978 marked the first published description of predictive control under the name IDCOM, an acronym for Identification and Command (Richalet et al. 1978). A short time later, Cutler and Ramaker (1979) published a predictive control algorithm under the name dynamic matrix control (DMC). Both approaches had been applied industrially for several years before the first publications appeared. These predictive control approaches targeted the more difficult industrial control problems that could not be adequately handled with other methods, either with conventional PID control or with advanced regulatory control (ARC) techniques that rely on single-loop controllers augmented with overrides, feedforwards/decouplers, and custom logic.

The basic idea behind the predictive control approach is shown in Fig. 2 for the case of a single-input single-output, stable system. Future predictions of inputs and outputs are denoted with the hat symbol and shown as dashes; double indexes, $t|k$, indicate future values at time t based on information up to and including time k . The dynamic optimization problem is to bring future predicted outputs ($\hat{y}_{k|k+1}, \dots, \hat{y}_{k|k+P}$) close to a desired trajectory over a prediction horizon, P , by means of a future sequence of inputs ($\hat{u}_{k|k}, \dots, \hat{u}_{k|k+M-1}$) calculated over a control horizon M . The trajectory may be a constant setpoint. When the MPC embeds a steady-state optimizer, future predicted outputs and inputs are driven to their respective steady-state optimums, y^{ss} and u^{ss} . In the general case, the dynamic optimization is performed subject to constraints that may be imposed on future inputs and outputs. Only the first of the future moves is implemented and the optimization is repeated at

the next time instant. Feedback, which accounts for unmeasured disturbances *and* model error, is incorporated by adjusting all future output predictions, prior to the optimization, based on the difference between the output measurement y_k and the previous prediction $\hat{y}_{k|k-1}$, denoted by $d_{k|k}$ (i.e., the prediction error at time instant k). For the case of a stable model, most MPCs shift the entire prediction by $d_{k|k}$. Future predicted values of the outputs depend on both past *and* future values inputs. If no future input changes are made (at time k or after), the model can be used to calculate the future “free” output response, $y_{t,k}^0$, which will ultimately settle at a new steady-state value based on the settling time (or time to steady state of the model, T_{ss}). For the unconstrained case, it is straightforward to show that the optimal result is a linear control law that depends only on the error between the desired trajectory and the free output response.

The predictive approach seemed to contrast with the state-space optimal control method of the time, the linear quadratic regulator (LQR). Later research exposed the similarities to LQR and also internal model control (IMC) (Garcia and Morari 1982), although these techniques did not solve an online optimization problem. Optimization-based control approaches became feasible for industrial applications due to (1) the slower sampling requirements of most industrial control problems (on the order of 1 minute or slower) and the hierarchical implementations in which MPC provides setpoints to lower-level PID controllers which execute on a much faster sample time (on the order of seconds or faster).

Although the basic ideas behind MPC remain, industrial MPC technology has changed considerably since the first formulations in the late 1970s. Qin and Badgwell (2003) describe the enhancements to MPC technology that occurred over the next 20 plus years until the late 1990s. Enhancements since that time are highlighted in Darby and Nikolaou (2012). These improvements to MPC reflect increases in computer processing capability and additional requirements of industry, which have led to increased functionality and tools/techniques to simplify implementation. A summary of the significant enhancements that

have been made to industrial MPC is highlighted below.

Constraints: Posing input and output constraints as linear inequalities, expressed as a function of the future input sequence (Garcia and Morshedi 1986), and solved by a standard quadratic program or an iterative scheme which approximates one.

Two-stage formulations: Limitations of a single objective function led to two-stage formulations to handle MV degrees of freedom (constraint pushing) and steady-state optimization via a linear program (LP).

Integrators. In their native form, impulse and step response models can be applied only to stable systems (in which the impulse response model coefficients approach zero). Extensions to handle integrating variables included embedding a model of the difference of the integrating signal or integrating a fraction of the current prediction error into the future (implying an increasing $|d_{k+j|k}|$ for $j \geq 1$ in Fig. 2). The desired value of an integrator at steady state (e.g., zero slope) can be incorporated into two-stage formulations (see, e.g., Lee and Xiao 2000).

State-space models. The first state-space formulation of MPC, which was introduced in the late 1980s (Marquis and Broustail 1988), allowed MPC to be extended to integrating and unstable processes. It also made use of the Kalman filter which provided additional capability to estimate

plant states and unmeasured disturbances. Later, a state-space MPC offering was developed based on an infinite horizon (for both control and prediction) (Froisy 2006). These state-space approaches provided a connection back to unconstrained LQR theory.

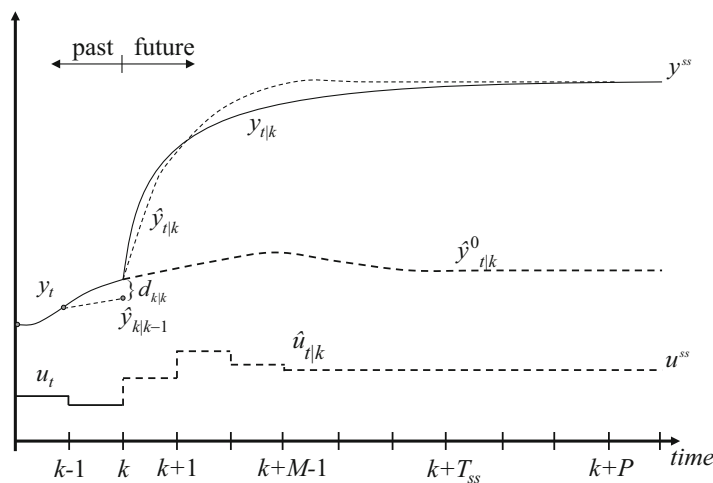
Nonlinear MPC. The first applications of nonlinear MPC, which appeared in the 1990s, were based on neural net models. In these approaches, a linear dynamic model was combined with a neural net model that accounted for static nonlinearity (Demoro et al. 1997; Zhao et al. 2001).

The late 1990s saw the introduction of an industrial nonlinear MPC based on first principles models derived from differential mass and energy balances and reaction kinetic expressions, expressed in differential algebraic equation (DAE) form (Young et al. 2002).

A process where nonlinear MPC is routinely applied is polymer manufacturing.

Identification techniques. Multivariable prediction error techniques are now routinely used. More recently, industrial application of subspace identification methods has appeared, following the development of these algorithms in the 1990s. Subspace methods incorporate the correlation of output measurements in the identification of a multivariable state-space model, which can be used directly in a state-space MPC or converted to an impulse or step response model.

Industrial MPC of Continuous Processes, Fig. 2 Predictive control approach



Testing methods. The 1990s saw increased use of automatic testing methods to generate data for (linear) dynamic model identification using uncorrelated binary signals. Since the 2000, closed-loop testing methods have received considerable attention. The motivation for closed-loop testing is to reduce implementation time and/or effort of the initial implementation as well as the ongoing need to reidentify the model of an industrial application in light of process changes. These closed-loop testing methods, which require a preliminary or existing model, utilize uncorrelated dither signals introduced as either biases to the controller MVs or injected through the steady-state LP or QP, where additional logic or optimization of the test protocol may be performed (Kalafatis et al. 2006; MacArthur and Zhan 2007; Zhu et al. 2012).

Mathematical Formulation

While there are differences in how the MPC problem is formulated and solved, the following general form captures most of the MPC products (Qin and Badgwell 2003), although not all terms may be present in a given product:

$$\min_{\Delta \mathbf{u}} \left[\sum_{j=1}^P \left\| \hat{\mathbf{y}}_{k+j|k} - \mathbf{y}_{k+j|k}^{ref} \right\|_{\mathbf{Q}_j}^2 + \sum_{j=1}^P \left\| \mathbf{s}_{k+j|k} \right\|_{\mathbf{T}_j}^2 + \sum_{j=1}^{M-1} \left\| \hat{\mathbf{u}}_{k+j|k} - \mathbf{u}^{ss} \right\|_{\mathbf{R}_j}^2 + \sum_{j=1}^{M-1} \left\| \Delta \mathbf{u}_{k+j|k} \right\|_{\mathbf{S}_j}^2 \right]$$

subject to

$$\left. \begin{aligned} \hat{\mathbf{x}}_{k+i|k} &= \mathbf{f}(\hat{\mathbf{x}}_{k+i-1|k}, \hat{\mathbf{u}}_{k+i-1|k}), \quad i = 1, \dots, P \\ \hat{\mathbf{y}}_{k+i|k} &= \mathbf{g}(\hat{\mathbf{x}}_{k+i|k}, \hat{\mathbf{u}}_{k+i|k}), \quad i = 1, \dots, P \end{aligned} \right\} \text{Model equations}$$

$$\left. \begin{aligned} \mathbf{y}^{\min} - \mathbf{s}_i &\leq \hat{\mathbf{y}}_{k+i|k} \leq \mathbf{y}^{\max} + \mathbf{s}_i, \quad i = 1, \dots, P \\ \mathbf{s}_j &\geq 0, \quad i = 1, \dots, P \end{aligned} \right\} \text{Output constraints/slacks}$$

$$\left. \begin{aligned} \mathbf{u}^{\min} &\leq \hat{\mathbf{u}}_{k+i|k} \leq \mathbf{u}^{\max}, \quad i = 0, \dots, M-1 \\ -\Delta \mathbf{u}^{\min} &\leq \Delta \mathbf{u}_{k+i|k} \leq \Delta \mathbf{u}^{\max}, \quad i = 0, \dots, M-1 \end{aligned} \right\} \text{Input constraints}$$

where the minimization is performed over the future sequence of inputs $\mathbf{U} \triangleq \hat{\mathbf{u}}_{k|k}, \hat{\mathbf{u}}_{k+1|k}, \dots, \hat{\mathbf{u}}_{k+M-1|k}$. The four terms in the objective function represent conflicting quadratic penalties ($\|\mathbf{x}\|_{\mathbf{A}}^2 \triangleq \mathbf{x}^T \mathbf{A} \mathbf{x}$); the penalty matrices are most always diagonal. The first term penalizes the error relative to a desired reference trajectory (cf. Fig. 2) originating at $\hat{\mathbf{y}}_{k|k}$ and terminating at a desired steady state, $\mathbf{y}^{ss}$; the second term penalizes output constraint violations over the prediction horizon (constraint softening); the third term penalizes inputs deviations from a desired steady state, either manually specified or calculated. The fourth term penalizes input changes as a means of trading off output tracking and input movement (move suppression).

The above formulation applies to both linear and nonlinear MPC. For linear MPCs, except for state-space formulations, there are no state equations, and the outputs in the dynamic model

are a function of only past inputs, such as with the finite step response model.

When a steady-state optimizer is present in the MPC, it provides the steady-state targets for $\mathbf{u}^{ss}$ (in the third quadratic term) and $\mathbf{y}^{ss}$ (in the output reference trajectory). Consider the case of linear MPC with an LP as the steady-state optimizer. The LP is typically formulated as

$$\min_{\Delta \mathbf{u}^{ss}} \mathbf{c}_u^T \Delta \mathbf{u}^{ss} + \mathbf{c}_y^T \Delta \mathbf{y}^{ss} + \mathbf{q}_+^T \mathbf{s}^+ + \mathbf{q}_-^T \mathbf{s}^-$$

subject to

$$\left. \begin{aligned} \Delta \mathbf{y}^{ss} &= \mathbf{G}^{ss} \Delta \mathbf{u}^{ss} \\ \mathbf{u}^{ss} &= \mathbf{u}_{k-1} + \Delta \mathbf{u}^{ss} \\ \mathbf{y}^{ss} &= \mathbf{y}_{k+T_{ss}|k}^0 + \Delta \mathbf{y}^{ss} \end{aligned} \right\} \text{Model equations}$$

$$\mathbf{y}^{\min} - \mathbf{s}^- \leq \mathbf{y}^{ss} \leq \mathbf{y}^{\max} + \mathbf{s}^+ \left\} \text{Output constraints}$$

$$\left. \begin{array}{l} \mathbf{u}^{\min} \leq \mathbf{u}^{ss} \leq \mathbf{u}^{\max} \\ -M\Delta\mathbf{u}^{\max} \leq \mathbf{u}^{ss} \leq M\Delta\mathbf{u}^{\max} \end{array} \right\} \text{Input constraints}$$

$\mathbf{G}^{ss}$ is formed from the gains of the linear dynamic model. Deviations outside minimum and maximum output limits ($\mathbf{s}^-$ and $\mathbf{s}^+$, respectively) are penalized, which provide constraint softening in the event all outputs cannot be simultaneously controlled within limits. The weighting in $\mathbf{q}_-$ and $\mathbf{q}_+$ determine the relative priorities of the output constraints. The input constraints, expressed in terms of $\Delta\mathbf{u}^{\max}$, prevent targets from being passed to the dynamic optimization that cannot be achieved. The resulting solution $-\mathbf{u}^{ss}$ and $\mathbf{y}^{ss}$ provides a consistent, achievable steady state for the dynamic MPC controller. Notice that for inputs, the steady-state delta is applied to the current value and, for outputs, the steady-state delta is applied to the steady-state prediction of the output without future moves, after correcting for the current model error (cf. Fig. 2). If a real-time optimizer is present, its outputs, which may be targets for CVs and/or MVs, are passed to the MPC steady-state optimizer and considered with other objectives but at lower weights or priorities.

Some additional differences or features found in industrial MPCs include:

1-Norm formulations where absolute deviations, instead of quadratic deviations, are penalized.

Use of zone trajectories or “funnels” with small or no penalty applied if predictions remain within the specified zone boundaries.

Use of a minimum movement criterion in either the dynamic or steady-state optimizations, which only lead to MV movement when CV predictions go outside specified limits. This can provide controller robustness to modeling errors.

Multiobjective formulations which solve a series of QP or LP problems instead of a single one and can be applied to the dynamic or steady-state optimizations. In these formulations higher-priority objectives are solved first, followed by lesser-priority objectives with the solution of the higher-priority objectives becoming equality constraints in subsequent optimizations (Maciejowski 2002).

MPC Design

Key design decisions for a given application are the number of MPC controllers and the selection of the MVs, DVs, and CVs for each controller; however, design decisions are not limited to just the MPC layer. The design problem is one of deciding on the best overall structure for the MPC(s) and the regulatory controls, given the control objectives, expected constraints, qualitative knowledge of the expected disturbances, and robustness considerations. It may be that existing measurements are insufficient and additional sensors may be required. In addition, a measurement may not be updated on a time interval consistent with acceptable dynamic control, for example, laboratory measurements and process composition analyzers. In this case, a soft sensor, or inferential estimator, may need to be developed from measurements such as temperature and pressure.

MPC is frequently applied to a major plant unit, with the MVs selected based on their sensitivity to key unit CVs and plant economics. Decisions regarding the number and size of the MPCs for a given application depend on plant objectives, (expected) constraints, and also designer preferences. When the objective is to minimize energy consumption based on fixed or specified feed rate, multiple smaller controllers can be used. In this situation, controllers are normally designed based on the grouping of MVs with the largest effect on the identified CVs, often leading to MPCs designed for individual sections of equipment, such as reactors and distillation columns. When the objective is to maximize feed (or certain products), larger controllers are normally designed, especially if there are multiple constraints that can limit plant throughput. The MPC steady-state LP or QP is ideally suited to solving the throughput maximization problem by utilizing all available MVs. The location of the most limiting constraints can impact the number of MPCs. If the major constraints are near the front end of the plant, one MPC can be designed which connects these constraints with key MVs such as feed rates, and other MPCs designed for the rest of the plant. If the major

constraints are located near the back of the plant, then a single MPC is normally considered. A single MPC has the advantage that throughput changes are better coordinated as a result of the prediction and optimization of future CVs and MVs.

The feed maximization objective is a major reason why MPCs have become larger with the advances in computer processing capability. However, there is generally a higher requirement on model consistency for larger controllers due to the increased number of possible constraint sets against which the MPC can operate. A larger controller can also be harder to implement and understand. This is a reason why some practitioners prefer implementing smaller MPCs at the potential loss of benefits.

MPC Practice

An MPC project is typically implemented in the following sequence:

Pretest and Preliminary MPC Design.
Plant Testing.
Model and Controller Development.
Commissioning.

These tasks apply whether the MPC is linear or nonlinear, but with some differences, primarily model development and in plant testing. In nonlinear MPC, key decisions are related to the model form and level of rigor. Note that with a fundamental model, lower-level PID loops must be included in the model, if the dynamics are significant; this is in contrast to empirical modeling, where the dynamics of the PID loops are embedded in the plant responses. A fundamental model will typically require less plant testing and make use of historical operating data to fit certain model parameters such as heat transfer coefficients and reaction constants. Historical data and/or data from a validated nonlinear static model can also be used to develop nonlinear static models (e.g., neural net) to combine with empirical dynamic models. As mentioned earlier, most industrial applications continue to rely on empirical linear dynamic

models, fit to data from a dedicated plant test. This will be the basis in the following discussion.

In the pretest phase of work, the key activity is one of determining the base-level regulatory controls for MPC, tuning of these controls, and determining if current plant instrumentation is adequate. The existing regulatory control scheme may need to be changed (i.e., modifying the loop pairing) to provide improved disturbance rejection or to better control constraints. It is common to retune a significant number of PID loops. Significant benefits often result from the pretest alone.

A range of testing approaches are used in plant testing for linear MPC, including both manual and automatic (computer-generated) test signal designs, most often in open loop but, increasingly, in closed loop. Most input testing continues to be based on uncorrelated signals, implemented either manually or from computer-generated random sequences. Model accuracy requirements dictate accuracy across a range of frequencies which is achieved by varying the duration of the steps. Model identification runs are made throughout the course of a test to determine when model accuracy is sufficient and a test can be stopped.

In the next phase of work, modeling of the plant is performed. This includes constructing the overall MPC model from individual identification runs, for example, deciding which models are significant and judging the models characteristics (dead times, inverse response settling time, gains) based on engineering/process and a priori knowledge. An important step is analyzing, and adjusting if necessary, the gains of the constructed model to insure the models gains satisfy mass balances and gain ratios do not result in fictitious degrees of freedom (due to model errors) that the steady-state optimizer could exploit. Also included is the development of any required inferentials or soft sensors, typically based on multivariate regression techniques such as principal component regression (PCR), principal component analysis (PCA), and partial least squares (PLS) or sometimes based on a fundamental model.

During controller development, initial controller tuning is performed. This relates to establishing criteria for utilizing available degrees of freedom and setting control variable priorities. In addition, initial tuning values are established for the dynamic control. Steady-state responses corresponding to expected constraint scenarios are analyzed to determine if the controller behaves as expected, especially with respect to the steady-state changes in the manipulated variables.

Commissioning involves testing and tuning the controller against different constraint sets. It is common to modify or revisit model decisions made earlier. In the worst case, control performance may be deemed unacceptable, and the control engineer is forced to revisit earlier decisions such as the base-level regulatory strategy or plant model quality, which would require retesting and reidentification of portions of the plant model. The main commissioning effort typically takes place over a 2- to 3-week period but can vary based on the size and model density of the controller. In reality, commissioning, or, more accurately, controller maintenance, is an ongoing activity. It is important that the operating company has in-house expertise that can be used to answer questions (“why is the controller doing that?”), troubleshoot, and modify the controller to reflect new operating modes and constraint sets.

Current and Future Directions

Future developments are expected to follow extensions of current approaches. We are seeing continued enhancements in the area of closed-loop testing, for example, automatic adjustment of test signal amplitudes, options for closed-loop update of controller models, and the combining of control and identification objectives (“dual” control). A possible enhancement would be to formulate the plant test as a DOE (design of experiments) optimization problem that could, for example, target specific models or model parameters.

In the model identification area, extensions have started to appear which allow constraints to be imposed, for example, on dead times or gains, thus allowing a priori knowledge to be

used. Another important area that has seen recent emphasis, and where more development can be expected, is in monitoring and diagnosis, for example, detecting which submodels of MPC have become inaccurate and significantly impacting control and thus require reidentification.

As mentioned earlier, one of the advantages of state-space modeling is the inherent flexibility to model unmeasured disturbances (i.e., $d_{k+j|j}$, cf. Fig. 2); however, these have not found widespread use in industry. A useful enhancement would be a framework for developing and implementing improved estimators in a convenient and transparent manner that would be applicable to traditional FIR- and FSR-based MPCs.

An area receiving increased attention concerns the coordination of multiple MPCs in a cascade arrangement. Most of the applications and research has focused on a single MPC or multiply distributed MPCs. These newer coordinators which, in the past used the same models as the underlying MPCs, may use different models (possibly nonlinear). In some cases the coordinator is solving higher-level plant objectives that historically were addressed by real-time optimization (RTO).

In the area of nonlinear control, the use of hybrid modeling approaches has increased, for example, integrating known fundamental model relationships with neural net or linear time-varying dynamic models. The motivation is in reducing complexity and controller execution times. The use of hybrid techniques can be expected to further increase, especially if nonlinear control is to be applied more broadly to larger control problems. Even in situations where control with linear MPC is adequate, there may be benefits from the use of hybrid or fundamental models, even if the models are not directly used online. The model could always be linearized for online use. Potential benefits would be quicker model development and model consistency. In the longer term, one can foresee a more general modeling and control environment where the user would not have to be concerned with the distinction between linear and nonlinear models and would be able to easily incorporate known relationships into the controller model.

Cross-References

- [Control Hierarchy of Large Processing Plants: An Overview](#)
- [Model Predictive Control in Practice](#)
- [Model-Based Performance Optimizing Control](#)
- [Real-Time Optimization of Industrial Processes](#)

Bibliography

- Cutler CR, Ramaker BL (1979) Dynamic matrix control – a computer control algorithm. AIChE National Meeting, Houston
- Darby ML, Nikolaou M (2012) MPC: current practice and challenges. *Control Eng Pract* 20:328–352
- Demoro E, Axelrud C, Johnson D, Martin G (1997) Neural network modeling and control of polypropylene process. In: Society of plastic engineers int'l conference, Houston, pp 487–499
- Froisy JB (2006) Model predictive control – building a bridge between theory and practice. *Comput Chem Eng* 30:1426–1435
- Garcia CE, Morari M (1982) Internal model control. 1. A unifying review and some new results. *Ind Eng Chem Process Des Dev* 21:308–323
- Garcia CE, Morshedi AM (1986) Quadratic programming solution of dynamic matrix control (QDMC). *Chem Eng Commun* 46:73–87
- Kalafatis K, Patel K, Harmse M, Zheng Q, Craig M (2006) Multivariable step testing for MPC projects reduces crude unit testing time. *Hydrocarb Process* 86:93–99
- Lee JH, Xiao J (2000) Use of two-stage optimization in model predictive control of stable and integrating. *Comput Chem Eng* 24:1591–1596
- MacArthur JW, Zhan C (2007) A practical multi-stage method for fully automated closed-loop identification of industrial processes. *J Process Control* 17:770–786
- Maciejowski J (2002) *Predictive control with constraints*. Prentice Hall, Englewood Cliffs
- Marquis P, Broustail JP (1988) SMOC, a bridge between state space and model predictive controllers: application to the automation of a hydrotreating unit. In: *Proceedings of the 1988 IFAC workshop on model based process control*, Atlanta, pp 37–43
- Qin JQ, Badgwell TA (2003) A survey of industrial model predictive control. *Control Eng Pract* 11:733–764
- Richalet J, Rault A, Testud J, Papon J (1978) Model predictive control: applications to industrial processes. *Automatica* 14:413–428
- Young RE, Bartusiak D, Fontaine RW (2002) Evolution of an industrial nonlinear model predictive controller. In: *Sixth international conference on chemical process control*, Tucson, CACHE American Institute of Chemical Engineers, pp 342–351
- Zhao H, Guiver JP, Neelakantan R, Biegler L (2001) A nonlinear industrial model predictive controller using integrated PLS and neural net state space model. *Control Eng Pract* 9:125–133
- Zhu Y, Patwardhan R, Wagner S, Zhao J (2012) Towards a low cost and high performance MPC: the role of system identification. *Comput Chem Eng* 51:124–135

Inertial Navigation

Renato Zanetti¹ and Christopher D'Souza²

¹University of Texas at Austin, Austin, TX, USA

²NASA Johnson Space Center, Houston, TX, USA

Abstract

Inertial navigation refers to the calculation of position, velocity, and orientation using measurements from inertial measurement units (IMUs) comprising accelerometers and gyroscopes. In particular, inertial navigation systems integrate forward in time the initial state of the systems replacing dynamic models with IMU measurements. This entry reviews the inertial navigation equations in a compact concise manner. Both the attitude and translational integration equations are presented. The work focuses on strapdown inertial measurement units, with particular emphasis on the algorithms used to compensate for the vehicle's rotation.

Keywords

Inertial measurement unit · Coning correction · Sculling correction · Navigation

Introduction

Inertial navigation has been a critical component of aerospace control systems since their inception. In 1953, Charles S. Draper's space inertial reference equipment famously guided a B-29 bomber from Massachusetts to Los Angeles without the aid of a pilot. This spurred

the development of onboard inertial navigation systems. Following the development of inertial instruments for ballistic missile navigation systems, inertial measurement instruments were used on the Apollo missions for the Saturn V rockets, the Command Module and the Lunar Module.

State feedback controllers often require an observer to reconstruct the state from the observations. In aerospace applications, the controlled state variables usually consist of (at least) position, velocity, and attitude; the process of determining these quantities is referred to as navigation. Typical navigation systems contain models of both the dynamics and the measurements in order to estimate the state. In virtually all inertial navigation systems, the dynamic models of acceleration and angular velocity are replaced by measured quantities (often referred to as “model replacement mode”).

The first inertial navigation systems used in aerospace applications (like Apollo) usually consisted of a gyro-stabilized platform mounted on gimbals designed to remain fixed in inertial space. The orientation of the system with respect to an inertial frame is therefore directly measurable as roll, pitch, and yaw angles at the base of the gimbals. Linear accelerometers are mounted on the stable platform to measure inertial nongravitational accelerations. Simple electronics are used to integrate the acceleration to obtain position and velocity. Alternatively, reference integrating gyros and integrating accelerometers can be both mounted on the stable platform to sense attitude changes and velocity changes, respectively.

Gimbaled inertial measurement units (IMUs) are still used today for high-precision systems, but their cost, weight, and reliance on moving mechanical parts reduce their popularity in favor of strapdown IMUs. The advent of inexpensive and lightweight digital computers has allowed a shift in complexity from hardware to software. In strapdown IMUs, the mechanical gimbals are replaced by strapdown systems, in which sensors are fixed with respect to the vehicle. Measured quantities are now with respect to the rotating vehicle frame, and software/firmware is used to

compensate for the vehicle’s motion. Figure 1 depicts the gimbaled IMU used by Apollo.

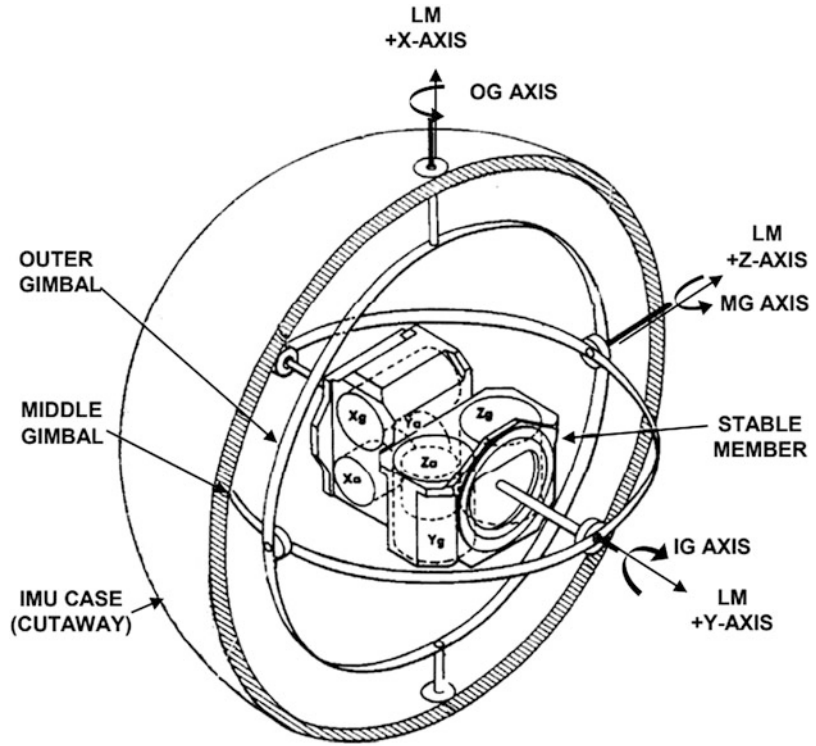
Inertial navigation is the process of navigating with respect to an inertial reference frame. This process is most often done by propagating the state of the system forward in time using IMU measurements (i.e., internal measurements). Inertial navigation systems can be aided with external measurements (such as global positioning system or optical measurements) in order to improve the estimation accuracy. With external aiding, the inertial navigation equations are used in the state propagation phase of an estimator such as an extended Kalman filter (EKF).

The goal of this entry is to discuss the inertial navigation equations used in a strapdown system, which is by far the most commonly used type of inertial navigation system today. To maximize the accuracy of the system, measurements are acquired at very high rates, in the order of several hundred or thousands of Hertz, and they are often preprocessed by the sensor’s firmware to accumulate them to a lower rate and to compensate for the rotation of the vehicle; these compensations are called coning (for rotational motion) and sculling (for translational motion). The navigation flight software uses these compensated and accumulated measurements to produce a state estimate at a much lower frequency (typically tens of Hertz).

Inertial Navigation

Strapdown IMUs are advantageous from a cost, weight, and mechanical complexity standpoint, but they are usually less accurate than gimbaled IMUs and introduce different challenges, such as coning and sculling. Coning corrections are sometimes called “non-commutivity rate” corrections. In gimbaled IMUs, the navigation base or platform is stable (sometimes referred to as the “stable member”) with respect to inertial space. In a gimbaled IMU, the inertial attitude is measured directly, and the nongravitational acceleration is measured in inertial coordinates. This is not the case for strapdown IMUs, which measure angular rates and specific accelerations (ones that

Inertial Navigation,
Fig. 1 Gimbaled IMU
 used in Apollo. (Image
 Credit: NASA)



do not include gravity) in a vehicle-fixed coordinate system. These measurements are used by a digital computer that integrates position, velocity, and attitude forward in time. In order to avoid loss of information, rather than sampling and holding the acceleration and angular velocity measurements, they are usually accumulated by the sensor. The accumulated measurements are often internally compensated for repeatable measurement errors with factory calibrated parameters and for the rotation of the vehicle with coning and sculling compensation. These compensations are presented in section “[Coning and Sculling](#)”. The propagation equations using the accumulated and compensated IMU measurements are presented in section “[Inertial Navigation Equations](#)”.

Preliminaries

Vector quantities such as position, velocity, and angular velocity are represented by a lowercase bold symbol. All vectors are column vectors, and the frame in which the vector’s components are expressed is denoted with a superscript: i for

inertial, b for vehicle-fixed (body-fixed), and e for Earth-centered Earth-fixed (ECEF); for example, $\mathbf{r}^i$ is the inertial position. A second superscript to the left of a “dot” indicates the frame in which time derivatives are taken. For example, the inertial time evolution of position $\mathbf{r}$ and velocity $\mathbf{v}$ are

$${}^i\dot{\mathbf{r}}^i(t) = \mathbf{v}^i(t), \quad (1)$$

$${}^i\dot{\mathbf{v}}^i(t) = \mathbf{g}^i(\mathbf{r}^i) + \mathbf{a}^i(t), \quad (2)$$

where $\mathbf{g}(\mathbf{r})$ represents the gravitational acceleration and $\mathbf{a}(t)$ the sum of all nongravitational accelerations. The rotation vector $\boldsymbol{\phi}$ is defined as the product of the Euler angle ϕ and axis $\mathbf{n}$

$$\boldsymbol{\phi}^b = \phi \mathbf{n}^b, \quad (3)$$

and evolves in time according to the Bortz equation Markley and Crassidis (2014)

$$\dot{\phi}^b(t) = \omega^b(t) + \frac{1}{2}\phi^b(t) \times \omega^b(t) + \left[1 - \frac{\phi(t) \sin \phi(t)}{2(1 - \cos \phi(t))}\right] \frac{\phi^b(t) \times (\phi^b(t) \times \omega^b(t))}{\phi(t)^2}, \quad (4)$$

where $\omega^b(t)$ is the body angular velocity with respect to the inertial frame.

The quaternion is defined as

$$\bar{\mathbf{q}} = \begin{bmatrix} \sin(\phi/2) \mathbf{n} \\ \cos(\phi/2) \end{bmatrix} = \begin{bmatrix} \mathbf{q}_v \\ q_s \end{bmatrix}, \quad (5)$$

and describes the rotation by the angle ϕ around the axis $\mathbf{n}$ is a globally non-singular parameterization. We are particularly interested in the inertial-to-body quaternion $\bar{\mathbf{q}}_i^b$ that evolves as

$$\dot{\bar{\mathbf{q}}}_i^b(t) = \frac{1}{2} \begin{bmatrix} \omega^b(t) \\ 0 \end{bmatrix} \otimes \bar{\mathbf{q}}_i^b(t), \quad (6)$$

where $\otimes$ is the modified quaternion multiplication defined as such that quaternions multiply in the same order as attitude matrices (direction cosine matrices).

The direction cosine matrix $\mathbf{T}_i^b$ (i.e., transformation matrix or attitude matrix) transforms vector coordinates from the inertial frame to the body frame and is obtained from the quaternion $\bar{\mathbf{q}}_i^b$ as

$$\mathbf{T}_i^b = \mathbf{T}(\bar{\mathbf{q}}_i^b) = \mathbf{I}_3 - 2q_s[\mathbf{q}_v \times] + 2[\mathbf{q}_v \times]^2, \quad (7)$$

where the skew-symmetric cross product matrix $[\mathbf{w} \times]$ is defined such that for all three-dimensional vectors $\mathbf{w}$ and $\mathbf{v}$, the following equivalence holds: $\mathbf{w} \times \mathbf{v} = [\mathbf{w} \times]\mathbf{v}$.

The continuous-time inertial navigation equations are then given by

$${}^i\dot{\mathbf{r}}^i(t) = \mathbf{v}^i(t), \quad (8)$$

$${}^i\dot{\mathbf{v}}^i(t) = \mathbf{g}^i(\mathbf{r}^i(t)) + \mathbf{T}(\bar{\mathbf{q}}_i^b)^T \mathbf{a}^b(t), \quad (9)$$

$$\dot{\bar{\mathbf{q}}}_i^b(t) = \frac{1}{2} \begin{bmatrix} \omega^b(t) \\ 0 \end{bmatrix} \otimes \bar{\mathbf{q}}_i^b(t). \quad (10)$$

To increase accuracy without integrating the above equations at thousands of Hertz, acceleration and angular velocity are usually not the output of the sensor, rather their accumulated values are

$$\Delta\phi_k^b(i) = \int_{t_k+(i-1)\Delta t_s}^{t_k+i\Delta t_s} \omega^b(\tau) d\tau, \quad (11)$$

$$\Delta\mathbf{v}_k^b(i) = \int_{t_k+(i-1)\Delta t_s}^{t_k+i\Delta t_s} \mathbf{a}^b(\tau) d\tau, \quad (12)$$

where Δt_s is the sensor's sampling time, typically thousands of Hertz. Our first task is to accumulate the measurements into more manageable frequencies to output them to the flight computer while compensating them for the body frame's rotation.

Coning and Sculling

Since the vehicle is, generally, rotating, we cannot simply add many $\Delta\phi_k^b(i)$ and $\Delta\mathbf{v}_k^b(i)$ together because they are expressed in different frames, i.e., at each sample time, the body-fixed frame has a different orientation. Compensating for this effect in the angular velocity is called coning, while performing the same for accelerations is called sculling.

Coning Compensation

The goal is to downsample the gyro measurement while compensating for the frame rotation. Let Δt_k be the IMU output interval; Δt_k is an integer multiple of the IMU's internal sample time Δt_s . The output of the IMU is the total rotation of the body frame over Δt_k , and this rotation is usually expressed via a rotation vector. Therefore, we need to solve the Bortz equation (4) for $\Delta\theta_k^b = \phi^b(t_k + \Delta t_k)$ with initial condition $\phi^b(t_k) = \mathbf{0}$.

The IMU output frequency $1/\Delta t_k$ is usually of the order of hundreds of Hertz; therefore, $\phi(t_k + \Delta t_k)$ is a small quantity, and we can proceed by linearizing the Bortz equation centered at zero

$${}^i\dot{\phi}^b \simeq \omega^b + \frac{1}{2}\phi^b \times \omega^b, \quad \phi(t_k) = \mathbf{0}, \quad (13)$$

and approximating its solution as

$$\begin{aligned}\Delta\theta_k^b &= \phi^b(t_k + \Delta t_k) = \int_0^{\Delta T} \omega^b(t_k + \tau) d\tau \\ &+ \frac{1}{2} \int_0^{\Delta T} \left[\int_0^\tau \omega^b(t_k + \xi) d\xi \right] \\ &\times \omega^b(t_k + \tau) d\tau.\end{aligned}\quad (14)$$

The first of the two terms in this equation is simply given by sum of all $\Delta\theta^b(t_s)$ terms, while the second is called coning correction. Equation (14) is solved by approximating all integrals as sums. Assume $\Delta t_k = N \Delta t_s$ and let $\Delta\phi_k^b(i)$ be the measured attitude increment over the sampling time Δt_s :

$$\begin{aligned}\Delta\phi_k^b(i) &= \phi^b(t_k + i\Delta t_s) \\ &- \phi^b(t_k + (i-1)\Delta t_s), \quad i=1, \dots, N.\end{aligned}$$

Then

$$\begin{aligned}\Delta\theta_k^b &= \sum_{i=1}^N \Delta\phi_k^b(i) \\ &+ \sum_{i=1}^{N-1} \sum_{j=i+1}^N \kappa_{ij} \Delta\phi_k^b(i) \times \Delta\phi_k^b(j).\end{aligned}\quad (15)$$

In a pure coning environment (angular velocity vector rotating at a constant rate), the contribution to the coning integral is independent of absolute time. Thus, taking advantage of this property allows the generalized form of the coning integral to be simplified to

$$\begin{aligned}&\sum_{i=1}^{N-1} \sum_{j=i+1}^N \kappa_{ij} \Delta\phi_k^b(i) \times \Delta\phi_k^b(j) \\ &= \left[\sum_{i=1}^{N-1} \kappa_i \Delta\phi_k^b(i) \right] \times \Delta\phi_k^b(N),\end{aligned}\quad (16)$$

where κ_i are the constant coning coefficients that can be calculated to optimize the results.

In a pure coning environment, the coning coefficients can be evaluated to any order in the coning algorithm. These are the so-called

Ignagni correction coefficients (Ignagni 1996, 1990, 1998) and are shown in Table 1.

Sculling Compensation

Let $\Delta\mathbf{v}_k^b(i)$ be the integrated nongravitational acceleration over one sampling time Δt_s

$$\Delta\mathbf{v}_k^b(i) = \int_{t_k + (i-1)\Delta t_s}^{t_k + i\Delta t_s} \mathbf{a}^b(\tau) d\tau.$$

The goal of sculling is to accumulate the sampled $\Delta\mathbf{v}_k^b(i)$ into lower frequency outputs while compensating for the rotation of the body-fixed frame. As before, the IMU output interval is Δt_k , and $\Delta t_k = N \Delta t_s$. The lower frequency and compensated accelerometer output is denoted by $\Delta\mathbf{v}_k^b$ and can be coordinatized in the body-fixed frame at the beginning, middle, or end of the interval Δt_k . For our purposes, we will develop the equations for the beginning of the interval:

$$\Delta\mathbf{v}_k^b = \int_{t_k}^{t_{k+1}} \mathbf{T}_{b(\tau)}^{b_k} \mathbf{a}^b(\tau) d\tau, \quad (17)$$

where the coning correction is anchored at t_k . The DCM from time t to t_k , where the time interval is small, is given by:

$$\begin{aligned}\mathbf{T}_{b(t)}^{b_k} &= \left(\mathbf{T}_{b_k}^{b(t)} \right)^T = \mathbf{I}_3 + \frac{\sin \phi(t)}{\phi(t)} \left[\phi^b(t) \times \right] \\ &+ \frac{1 - \cos \phi(t)}{\phi(t)^2} \left[\phi^b(t) \times \right]^2 \\ &\approx \mathbf{I}_3 + \left[\phi^b(t) \times \right],\end{aligned}$$

where $\phi^b(t_k) = \mathbf{0}$ and the approximations are valid for small ϕ . Therefore, $\Delta\mathbf{v}_k^b$ could be approximated as

$$\begin{aligned}\Delta\mathbf{v}_k^b &= \int_{t_k}^{t_{k+1}} \mathbf{a}^b(\tau) d\tau \\ &+ \int_{t_k}^{t_{k+1}} \phi^b(\tau) \times \mathbf{a}^b(\tau) d\tau.\end{aligned}\quad (18)$$

A fundamental limitation in the above equation is that the second term does not accurately represent the high-frequency content of the signal. A solu-

**Inertial
Navigation, Table 1**
Coning correction
coefficients for pure coning
motion

Coefficient	4 Step	5 Step	6 Step	7 Step	8 Step	9 Step
κ_1	0.514286	0.496032	0.501082	0.499709	0.500078	0.499980
κ_2	0.876191	1.041667	0.986580	1.004177	0.998735	1.000375
κ_3	2.038095	1.289683	1.579221	1.471523	1.509824	1.496725
κ_4	—	2.728175	1.695671	2.124476	1.951293	2.018250
κ_5	—	—	3.41925	2.096873	2.675591	2.426488
κ_6	—	—	—	4.110936	2.494839	3.231230
κ_7	—	—	—	—	4.802975	2.890517
κ_8	—	—	—	—	—	5.495258

tion is to split Eq. (18) as the sum of two equal components and integrate one of them by parts to more accurately represent the higher frequencies:

$$\begin{aligned} \Delta \mathbf{v}_k^b &= \int_{t_k}^{t_{k+1}} \mathbf{a}^b(\tau) d\tau + \frac{1}{2} \int_{t_k}^{t_{k+1}} \boldsymbol{\phi}^b(\tau) \times \mathbf{a}^b d\tau \\ &+ \frac{1}{2} \Delta \boldsymbol{\theta}_k^b \times \int_{t_k}^{t_{k+1}} \mathbf{a}^b(\tau) d\tau \\ &+ \frac{1}{2} \int_{t_k}^{t_{k+1}} \left[\int_{t_{k-1}}^{\tau} \mathbf{a}^b(\sigma) d\sigma \right] \times \boldsymbol{\omega}^b(\tau) d\tau \end{aligned} \quad (19)$$

$$\begin{aligned} &= \int_{t_k}^{t_{k+1}} \mathbf{a}^b(\tau) d\tau + \frac{1}{2} \Delta \boldsymbol{\theta}_k^b \times \int_{t_k}^{t_{k+1}} \mathbf{a}^b(\tau) d\tau \\ &+ \frac{1}{2} \int_{t_k}^{t_{k+1}} \left[\boldsymbol{\phi}^b(\tau) \times \mathbf{a}^b(\tau) \right. \end{aligned} \quad (20)$$

$$\left. + \left(\int_{t_{k-1}}^{\tau} \mathbf{a}^b(\sigma) d\sigma \right) \times \boldsymbol{\omega}^b(\tau) \right] d\tau. \quad (21)$$

We now discuss each of the terms in Eq. (21) (Savage 1998). The first term represents a pure integration/summation of the accelerometer outputs. The second term represents a rotation correction, accounting for the fact that the frames are rotating. The third term represents the integrated contribution of the high-frequency content of the specific force.

Pure sculling motion is defined as the motion where the angular velocity and the acceleration both rotate at a constant and equal rate. Under pure sculling motion, we can rewrite the integral in Eq. (21) with the same coning coefficients

as before to obtain the sculling computational algorithm:

$$\mathbf{v}_k^b = \sum_{i=1}^K \Delta \mathbf{v}_k^b(i) \quad (22)$$

$$\begin{aligned} \Delta \mathbf{v}_k^b &= \mathbf{v}_k^b + \frac{1}{2} \mathbf{v}_k^b \times \left[\sum_{i=1}^K \Delta \boldsymbol{\phi}_k^b(i) \right] \\ &+ \left[\sum_{i=1}^K \kappa_i \Delta \boldsymbol{\phi}_k^b + (i) \right] \times \mathbf{v}_k^b \\ &+ \left[\sum_{i=1}^K \kappa_i \Delta \mathbf{v}_k^b(i) \right] \times \left[\sum_{i=1}^K \Delta \boldsymbol{\phi}_k^b(i) \right] \end{aligned} \quad (23)$$

where the κ_i are the *coning* correction coefficients shown in Table 1.

Inertial Navigation Equations

The coning and sculling corrections and accumulation of high-rate IMU measurements into more manageable rates are often done by the IMU firmware. The outputs $\Delta \boldsymbol{\theta}_k^b$ and $\Delta \mathbf{v}_k^b$ of the sensor are then used by the flight software to integrate the position and attitude of the vehicle, i.e., to obtain an inertial solution. Different integration schemes can be used. For high-precision positioning, an additional rotation correction can be added to the position propagation equations (Savage 1998); for fast systems (tens of Hertz), it is common to use the following simple equations:

$$\boldsymbol{\alpha}_k^i = \mathbf{g}^i(\mathbf{r}^i(t_k)) \Delta t_k + \mathbf{T}(\bar{\mathbf{q}}_i^b(t_k))^T \Delta \mathbf{v}_k^b, \quad (24)$$

$$\mathbf{r}^i(t_{k+1}) = \mathbf{r}^i(t_k) + \mathbf{v}^i(t_k)\Delta t_k + \frac{1}{2}\boldsymbol{\alpha}_k^i\Delta t_k, \quad (25)$$

$$\mathbf{v}^i(t_{k+1}) = \mathbf{v}^i(t_k) + \boldsymbol{\alpha}_k^i, \quad (26)$$

$$\begin{aligned} \bar{\mathbf{q}}_i^b(t_{k+1}) &= \begin{bmatrix} \sin(\|\Delta\boldsymbol{\theta}_k^b\|/2) & \Delta\boldsymbol{\theta}_k^b/\|\Delta\boldsymbol{\theta}_k^b\| \\ \cos(\|\Delta\boldsymbol{\theta}_k^b\|/2) & \end{bmatrix} \otimes \bar{\mathbf{q}}_i^b(t_k). \end{aligned} \quad (27)$$

Often, it is desired to express state quantities (position, velocity, and attitude) in the ECEF rather than inertial frame. Let $\boldsymbol{\Omega}_E^e$ be Earth's rotation rate in ECEF coordinates. Then the equations in the ECEF frame are given by Titterton and Weston (2004)

$$\begin{aligned} \boldsymbol{\alpha}_k^e &= \mathbf{g}^e(\mathbf{r}^e(t_k))\Delta t_k + \mathbf{T}(\bar{\mathbf{q}}_e^b(t_k))^T \Delta\mathbf{v}_k^b \\ &\quad - 2\boldsymbol{\Omega}_E^e \times \mathbf{v}^e(t_k) - \boldsymbol{\Omega}_E^e \\ &\quad \times (\boldsymbol{\Omega}_E^e \times \mathbf{r}^e(t_k)), \end{aligned} \quad (28)$$

$$\boldsymbol{\vartheta}_k^b = \Delta\boldsymbol{\theta}_k^b - \mathbf{T}(\bar{\mathbf{q}}_e^b(t_k))\boldsymbol{\Omega}_E^e\Delta t_k, \quad (29)$$

$$\mathbf{r}^e(t_{k+1}) = \mathbf{r}^e(t_k) + \mathbf{v}^e(t_k)\Delta t_k + \frac{1}{2}\boldsymbol{\alpha}_k^e\Delta t_k, \quad (30)$$

$$\mathbf{v}^e(t_{k+1}) = \mathbf{v}^e(t_k) + \boldsymbol{\alpha}_k^e, \quad (31)$$

$$\bar{\mathbf{q}}_e^b(t_{k+1}) = \begin{bmatrix} \sin(\|\boldsymbol{\vartheta}_k^b\|/2) & \boldsymbol{\vartheta}_k^b/\|\boldsymbol{\vartheta}_k^b\| \\ \cos(\|\boldsymbol{\vartheta}_k^b\|/2) & \end{bmatrix} \otimes \bar{\mathbf{q}}_e^b(t_k). \quad (32)$$

More accurate higher-order integration methods can also be employed. It is possible to use local Earth frames, e.g., north-east-down, and express the position with latitude, longitude, and altitude. These frames possess singularities at the poles that can be overcome by the wander angle mechanization (Ignagni 2018).

Summary and Future Directions

Strapdown inertial navigation is of fundamental importance for the control of aerospace vehicles. It is used either by itself or in conjunction with external measurements to provide the time propagation equations in estimation algorithms such as the extended Kalman filter. This entry presents both the high rate accumulation equations with coning and sculling corrections and the lower rate navigation equations with integration of position, velocity, and attitude of the vehicle. Only optimal algorithms for pure coning/sculling motion are presented. Other algorithms exist to optimize other types of motion, such as maneuvers or vibrations. The choice of coning/sculling algorithms depends heavily on the application of interest, and it is the subject of continuous research to optimize for desired performance.

Cross-References

- [Extended Kalman Filters](#)
- [Spacecraft Attitude Determination](#)

Bibliography

- Ignagni MB (1990) Optimal strapdown attitude integration algorithms. *AIAA J Guid Control Dynam* 13(2):363–369
- Ignagni MB (1996) Efficient class of optimized coning compensation algorithms. *AIAA J Guid Control Dynam* 19(2):424–429
- Ignagni MB (1998) Duality of optimal strapdown sculling and coning compensation algorithms. *J Inst Navig* 45(2):85–95. Summerl
- Ignagni MB (2018) Strapdown navigation systems theory and applications. Champlain Press, Minneapolis
- Markley FL, Crassidis JL (2014) Fundamentals of spacecraft attitude determination and control. Space technology library. Springer, New York
- Savage PG (1998) Strapdown inertial navigation integration algorithm design part 2: velocity and position algorithms. *AIAA J Guid Control Dynam* 21(2): 208–221
- Titterton D, Weston J (2004) Strapdown inertial navigation technology, 2nd edn. Radar, Sonar and Navigation Series, The Institution of Engineering and Technology. Stevenage, Hertfordshire

Infinite-Dimensional

► [Backstepping for PDEs](#)

Information and Communication Complexity of Networked Control Systems

Serdar Yüksel

Department of Mathematics and Statistics,
Queen's University, Kingston, ON,
Canada

Abstract

Information and communication complexity of a networked control system identifies the minimum amount of information exchange needed between the decision makers (such as encoders, controllers, and actuators) to achieve a certain objective, which may be in terms of reaching a target state or a set, achieving a given expected cost threshold, or minimizing a cost. This formulation does not impose any constraints on the computational requirements to perform the communication or control. Both stochastic and deterministic formulations are considered. We also provide an overview of some recent research results on information structures in decentralized stochastic control and networked control and provide an (admittedly incomplete) literature review.

Keywords

Communication complexity · Networked control · Decentralized stochastic control · Information and control · Information structure design

Introduction

Consider a decentralized stochastic control (or a dynamic team) problem with L control stations (these will be referred to as decision makers and denoted by DMs) under the following dynamics and measurement equations

$$x_{t+1} = f_t(x_t, u_t^1, \dots, u_t^L, w_t), \quad t = 0, 1, \dots \quad (1)$$

$$y_t^i = g_t^i(x_t, u_{t-1}^1, \dots, u_{t-1}^L; v_t^i), \quad (2)$$

where $i \in \{1, 2, \dots, L\} =: \mathcal{L}$ and $x_0, w_{[0, T-1]}, v_{[0, T-1]}^i$ are mutually independent random variables with specified probability distributions. Here, we use the notation $w_{[0, t]} := \{w_s, 0 \leq s \leq t\}$.

The DMs are allowed to exchange limited information, see Fig. 1. The information exchange is facilitated by an encoding protocol $\mathcal{E}$ which is a collection of admissible encoding functions described as follows. Let the information available to DM i at time t be:

$$\mathcal{I}_t^i = \{y_{[1, t]}^i, u_{[1, t-1]}^i, z_{[0, t]}^{j, i}, z_{[0, t]}^{j, i}, j \in \mathcal{L}\},$$

where $z_t^{i, j}$ takes values in some finite set $\mathcal{Z}_t^{i, j}$ and is the information variable transmitted from DM i to DM j at time t generated with

$$\mathbf{z}_t^i = \{z_t^{i, j}, j \in \mathcal{L}\} = \mathcal{E}_t^i(\mathcal{I}_{t-1}^i, u_{t-1}^i, y_t^i), \quad (3)$$

and for $t = 0$, $\mathbf{z}_0^i = \{z_0^{i, j}, j \in \mathcal{L}\} = \mathcal{E}_0^i(y_0^i)$. The control actions are generated with

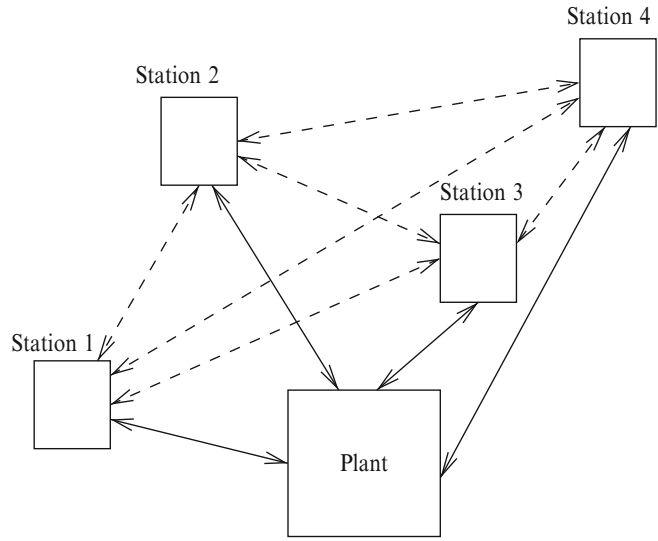
$$u_t^i = \gamma_t^i(\mathcal{I}_t^i),$$

for all DMs. Define $\log_2(|\mathcal{Z}_t^{i, j}|)$ to be the *communication rate from DM i to DM j at time t* and $\mathcal{R}(\mathbf{z}_{[0, T-1]}) = \sum_{t=0}^{T-1} \sum_{i, j \in \mathcal{L}} \log_2(|\mathcal{Z}_t^{i, j}|)$ to be the *(total) communication rate*. The minimum (total) communication rate over all coding and control policies subject to a design objective is called the *communication complexity* for this objective.

The above is a fixed-rate formulation for communication complexity, since for any two coder

Information and Communication Complexity of Networked Control Systems, Fig. 1

A decentralized networked control system with information exchange between decision makers



outputs, a fixed number of bits is used at any given time. One could also use variable-rate formulations, where, as the measure of the information rate, the cardinality of the message sets is replaced with the Shannon entropy of the exchanged messages. The variable-rate formulation exploits the probabilistic distribution of the system random variables to reduce the rate requirements; see Cover and Thomas (1991). This will be discussed further in section “Connections with Information Theory”.

Communication Complexity for Decentralized Dynamic Optimization

Let $\underline{\mathcal{E}}^i = \{\mathcal{E}_t^i, t \geq 0\}$ and $\underline{\gamma}^i = \{\gamma_t^i, t \geq 0\}$. Under a team-encoding policy $\underline{\mathcal{E}} = \{\underline{\mathcal{E}}^1, \underline{\mathcal{E}}^2, \dots, \underline{\mathcal{E}}^L\}$, and a team-control policy $\underline{\gamma} = \{\underline{\gamma}^1, \underline{\gamma}^2, \dots, \underline{\gamma}^L\}$, let the induced expected cost be

$$E^{\underline{\gamma}, \underline{\mathcal{E}}} \left[\sum_{t=0}^{T-1} c(x_t, u_t^1, u_t^2, \dots, u_t^L) \right]. \quad (4)$$

In networked control, the goal is to minimize (4) over all coding and control policies subject to information constraints and the information structure in the system. Let

$\mathbf{u}_t = \{u_t^1, u_t^2, \dots, u_t^L\}$. The following definition and example are from Yüksel and Başar (2013).

Definition 1 Given a decentralized control problem as above, *team cost-rate function* $C : \mathbb{R} \rightarrow \mathbb{R}$ is

$$C(R) := \inf_{\underline{\gamma}, \underline{\mathcal{E}}} \left\{ E^{\underline{\gamma}, \underline{\mathcal{E}}} \left[\sum_{t=0}^{T-1} c(x_t, \mathbf{u}_t) \right] : \frac{1}{T} \mathcal{R}(\mathbf{z}_{[0, T-1]}) \leq R \right\}.$$

We can define a dual function.

Definition 2 Given a decentralized control problem as above, *team rate-cost function* $R : \mathbb{R} \rightarrow \mathbb{R}$ is

$$R(C) := \inf_{\underline{\gamma}, \underline{\mathcal{E}}} \left\{ \frac{1}{T} \mathcal{R}(\mathbf{z}_{[0, T-1]}) : E^{\underline{\gamma}, \underline{\mathcal{E}}} \left[\sum_{t=0}^{T-1} c(x_t, \mathbf{u}_t) \right] \leq C \right\}.$$

The formulation here can be adjusted to include sequential (iterative) information exchange given a fixed ordering of actions, as opposed to a simultaneous (parallel) information exchange at any given time t . That is, instead of (3), we may have

$$\begin{aligned} \mathbf{z}_t^i &= \{z_t^{i,j}, j \in \{1, 2, \dots, L\}\} \\ &= \mathcal{E}_t^i(\mathcal{I}_{t-1}^i, u_{t-1}^i, y_t^i, \{z_t^{k,i}, k < i\}). \end{aligned} \quad (5)$$

Both to make the discussion more explicit and to show that a sequential (iterative) communication protocol may perform strictly better than an optimal parallel communication protocol given a total rate constraint, we state the following example: Consider the following setup with two DMs. Let x^1, x^2, p be uniformly distributed binary random variables, DM i have access to $y^i, i = 1, 2$, and

$$x = (p, x^1, x^2), \quad y^1 = p, \quad y^2 = (x^1, x^2),$$

and the cost function be

$$\begin{aligned} c(x, u^1, u^2) &= 1_{\{p=0\}}c(x^1, u^1, u^2) \\ &\quad + 1_{\{p=1\}}c(x^2, u^1, u^2), \end{aligned}$$

with

$$c(s, u^1, u^2) = (s - u^1)^2 + (s - u^2)^2.$$

Suppose that we wish to compute the minimum expected cost subject to a total rate of 2 bits that can be exchanged. Under a sequential scheme, if we allow DM 1 to encode y^1 to DM 2 with one bit, then a cost of 0 is achieved since DM 2 knows the relevant information that needs to be transmitted to DM 1, again with one bit: If $p = 0$, x^1 is the relevant random variable with an optimal policy $u^1 = u^2 = x^1$, and if $p = 1$, x^2 is relevant with an optimal policy $u^1 = u^2 = x^2$, and a cost of 0 is achieved. However, if the information exchange is parallel, then DM 2 does not know which state is the relevant one, and it can be shown through information theoretic arguments (Yüksel and Başar 2013, Prop. 12.5.1) that a cost of 0 cannot be achieved under any policy.

The formulation in Definition 1 can also be adjusted to allow for multiple rounds of communication per time stage. Keeping the total rate constant, having multiple rounds can enhance the performance for a class of team problems while keeping the total rate constant.

A formulation relevant to the one in Definition 1 has been considered in Teneketzis (1979) with mutual information constraints; please see section “Connections with Information Theory” for a discussion on such relaxations.

Structural Results on Optimal Policies

Even though obtaining explicit solutions for jointly optimal coding and control problems in its most general setup presented in section “Communication Complexity for Decentralized Dynamic Optimization” is very challenging, it is often very useful to obtain structural results on optimal coding and control policies since through such results one can reduce the search space for optimal policies to a smaller class of functions.

A particularly important special case of the above formulation is the setup with one sensor and one controller who receives information from the sensor, where a control-free or a controlled Markovian source is causally encoded. If the problem described in Definition 1 is for a real-time estimation problem for a Markov source, then the optimal causal fixed-rate coder minimizing any cost function (of the form studied here) uses only the last source symbol and the information at the controller’s memory; see Witsenhausen (1979) and with further refinements in Walrand and Varaiya (1983) which replaced the state with the conditional probability of the state given the information at the controller; a unified view is presented in Teneketzis (2006). These studies considered finite state Markov chains, with real state space and partially observed setup generalizations reported in Yüksel (2013). We refer the reader to Yüksel and Başar (2013, Chp. 10–11), and classical references such as Witsenhausen (1979), Walrand and Varaiya (1983), and more modern studies in Teneketzis (2006), Mahajan and Teneketzis (2009), Yüksel (2019) which presents a separation result involving quantization, estimation, and control, among many other recent references.

For dynamic team problems, such structural results typically follow from the construction of a controlled Markov chain (see Walrand and

Varaiya (1983) and applying tools from stochastic control theory to obtain structural results on optimal coding and control policies (see Nayyar et al. (2014) for a comprehensive construction and Yüksel and Başar (2013) for a review).

Communication Complexity in Decentralized Computation

Yao (1979) significantly influenced the research on communication complexity in distributed computation. This may be viewed as a special case of the setting considered earlier but with finite spaces and in a deterministic, control-free, and an error-free context: Consider two decision makers (DMs) who have access to local variables $x \in \{0, 1\}^n, y \in \{0, 1\}^n$. Given a function f of variables (x, y) , what is the maximum (over all input variables x, y) of the minimum amount of information exchange needed for at least one agent to compute the value of the function?: Let $s(x, y) = \{m_1, m_2, \dots, m_t\}$ be the communication symbols exchanged on input (x, y) during the execution of a communication protocol. Let m_i denote the i th binary message symbol with $|m_i|$ bits. The communication complexity for such a setup is defined as

$$R(f) = \min_{\gamma, \mathcal{E}} \max_{(x, y) \in \{0, 1\}^n \times \{0, 1\}^n} |s(x, y)|, \quad (6)$$

where $|s(x, y)| = \sum_{i=1}^t |m_i|$ and $\mathcal{E}$ is a protocol which dictates the iterative encoding functions as in (5) and γ is a decision policy.

For such problems, obtaining *good* lower bounds is in general challenging. One lower bound for such problems is obtained through the following reasoning: A subset of the form $A \times B$, where A and B are subsets of $\{0, 1\}^n$, is called an *f-monochromatic* rectangle if for every $x \in A, y \in B$, $f(x, y)$ is the same. It can be shown that given any finite message sequence $\{m_1, m_2, \dots, m_t\}$, the set $\{(x, y) : s(x, y) = \{m_1, m_2, \dots, m_t\}\}$ is an *f-monochromatic* rectangle. Hence, to minimize the number of messages, one needs to minimize the number of *f-monochromatic* rectangles which has led

to research in this direction. Upper bounds are typically obtained by explicit constructions. For a comprehensive review, see Kushilevitz and Nisan (2006).

The formulation above is on noise-free distributed computation. When one allows for approximation errors, the conditions are naturally more relaxed. We refer the reader to Tsitsiklis and Luo (1987) and Nemirovsky and Yudin (1983) for information complexity for optimization problems under various formulations and the detailed literature review present in Tsitsiklis and Luo (1987). A sequential setting with an information theoretic approach to the formulation of communication complexity has been considered in Raginsky and Rakhlin (2011). Giridhar and Kumar (2006) discuss distributed computation for a class of symmetric functions under information constraints and present a comprehensive review.

We note finally that for control systems, the discussion takes further aspects into account including a design objective, system dynamics, and the uncertainty in the system variables.

Communication Complexity in Reach-Control

Wong (2009) defines the communication complexity in networked control as follows. Consider a design specification where two DMs wish to steer the state of a dynamical system in finite time. This can be viewed as a setting in (1)–(2) with 4 DMs, where there is iterative communication between a sensor and a DM, and there is no stochastic noise in the system. Given a set of initial states $x_0 \in \mathcal{X}_0$, and finite sets of objective choices for each DM ($\mathcal{A}$ for DM 1, $\mathcal{B}$ for DM 2), the goal is to ensure that (i) there exists a finite time where both DMs know the final state of the system, (ii) the final state satisfies the choices of the DMs, and (iii) the finite time (when the objective is satisfied) is known by the DMs.

The communication complexity for such a system is defined as the infimum over all protocols of the supremum over the triple of initial states and choices of the DMs, such that the above

is satisfied. That is,

$$R(\mathcal{X}_0, \mathcal{A}, \mathcal{B}) = \inf_{\underline{\gamma}, \underline{\mathcal{E}}} \sup_{\alpha, \beta, x_0} R(\underline{\gamma}, \underline{\mathcal{E}}, \alpha, \beta, x_0),$$

where $R(\underline{\gamma}, \underline{\mathcal{E}}, \alpha, \beta, x_0)$ denotes the communication rate under the control and coding functions $\underline{\gamma}, \underline{\mathcal{E}}$, which satisfies the objectives given by the choices α, β and initial condition x_0 .

Wong obtains a cut-set type lower bound: Given a fixed initial state, a lower bound is given by $2D(f)$, where f is a function of the objective choices and $D(f)$ is a variation of $R(f)$ introduced in (6) with the additional property that both DMs know f at the end of the protocol. An upper bound is established by the exchange of the initial states and objective functions also taking into account signaling, that is, the communication through control actions, which is discussed further below in the context of stabilization. Wong and Baillieul (2012) consider a detailed analysis for a real-valued bilinear controlled decentralized system.

Connections with Information Theory

The field of information theory has made foundational contributions to such problems in a variety of contexts. An information theoretic setup typically entails settings where an unboundedly large sequence of messages are encoded and functions of which are to be computed. Such a setting is not applicable in a real-time setting but is very useful for obtaining performance bounds (i.e., good lower bounds on complexity) which can at certain instances be achievable even in a real-time setting. That is, instead of a single-realization of random variables in the setup of (1)–(2), the average performance for a large number of or infinitely many independent realizations/copies for such problems is typically considered.

In such a context, Definitions 1 and 2 can be adjusted so that the communication complexity is computed by mutual information (Cover and Thomas 1991). Replacing the fixed-rate or variable-rate (entropy) constraint in Definition 1 with a mutual information constraint leads to

desirable convexity properties for $C(R)$ and $R(C)$ which does not apply for fixed-rate formulations (Yüksel and Başar 2013, p. 169; György and Linder 2000; György et al. 2008). Such an information theoretic formulation can provide useful lower bounds and desirable analytical properties.

We note here the interesting discussion between decentralized computation and communication provided by Orlitsky and Roche (2001), as well as by Witsenhausen (1976) where a *probability-free* construction is considered and a *zero-error (non-asymptotic and error-free)* computation is considered in the same spirit as in Yao (1979).

Such decentralized computation problems can be viewed as multi-terminal source coding problems with a cost function aligned with the computation objective. Ma and Ishwar (2011) and Gamal and Kim (2012) provide a comprehensive treatment and review of information exchange requirements for computing. Essential in such constructions is the method of *binning* (Wyner and Ziv 1976), which is a key tool in distributed source and channel coding problems. Binning efficiently designates the enumeration of symbols (which can be confused at the receiver in the absence of coding) given the relevant information at the receiver DM from decentralized encoders.

Such problems involve interactive communications as well as multi-terminal coding problems. As mentioned earlier, it is also important to point out that multi-round protocols typically reduce the average rate requirements.

In the particular instance of the general setup noted at the end of section “[Communication Complexity for Decentralized Dynamic Optimization](#)”, where a controlled or control-free Markov source is to be coded across a noisy or noise-free communication channel under information constraints, information theoretic relaxations allow for lower bounds as well as computationally efficient algorithms to obtain such bounds. In the special case of linear systems driven by Gaussian noise, and typically across Gaussian channels, there has been a very significant research activity perhaps

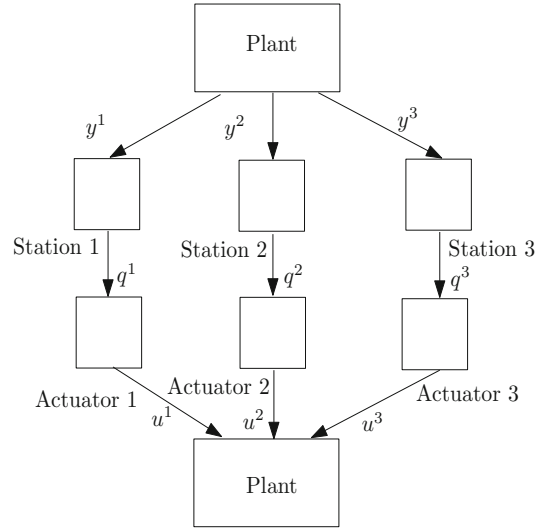
starting from Bansal and Başar (1989), with recent contributions including Tatikonda et al. (2004), Tanaka et al. (2016), Kostina and Hassibi (2019), and Stavrou et al. (2018), among many other references. We also note that information theoretic methods allow for useful lower and upper bounds even for non-Gaussian models; see, e.g., Derpich and Ostergaard (2012).

Communication Complexity in Decentralized Stabilization

An important relevant setting of reach-control is where the target final state is the zero vector: the system is to be stabilized. Consider the following special case of (1)–(2) for an LTI system

$$x_{t+1} = Ax_t + \sum_{j=1}^L B^j u_t^j, \quad y_t^i = C^i x_t \quad t = 0, 1, \dots \quad (7)$$

where $i \in \mathcal{L}$, and it is assumed that the joint system is stabilizable and detectable, but the individual pairs (A, B^i) may not be stabilizable or (A, C^i) may not be detectable. Here, $x_t \in \mathbb{R}^n$ is the state, $u_t^i \in \mathbb{R}^{m_i}$ is the control applied by station i , and $y_t^i \in \mathbb{R}^{p_i}$ is the observation available at station i , all at time t . The initial state x_0 is generated according to a probability distribution supported on a compact set $\mathcal{X}_0 \subset \mathbb{R}^n$. We denote controllable and unobservable subspaces at station i by K^i and N^i and refer to the subspace orthogonal to N^i as the observable subspace at the i th station, denoted by O^i . The information available to station i at time t is $I_t^i = \{y_{[0,t]}^i, u_{[0,t-1]}^i\}$. For such a system (see Fig. 2), it is possible for the controllers to communicate through the plant with the process known as *signaling* which can be used for communication of mode information among the decision makers. Denote by $i \rightarrow j$ the property that DM i can signal to DM j . This holds if and only if $C^j(A)^l B^i \neq 0$, for at least one l , $1 \leq l \leq n$. A directed graph $\mathcal{G}$ among the L stations can be constructed through such a communication relationship.



Information and Communication Complexity of Networked Control Systems, Fig. 2 Decentralized stabilization with multiple controllers

Suppose that A is such that in its Jordan form each Jordan block admits distinct real eigenvalues. Then, a lower bound on the communication complexity (per time stage) for stabilizability is given by $\sum_{|\lambda_i| > 1} \eta_{M_i} \log_2(|\lambda_i|)$, where

$$\eta_{M_i} = \min_{l, m \in \{1, 2, \dots, L\}} \{d(l, m) + 1 : l \rightarrow m, [x^i] \subset O^i \cup O^m, [x^i] \subset K^m\},$$

with $d(l, m)$ denoting the graph distance (number of edges in a shortest path) between DM l and DM m in $\mathcal{G}$ and $[x_i]$ denoting the subspace spanned by x_i . Furthermore, there exist stabilizing coding and control policies whose sum rate is arbitrarily close to this bound. When different Jordan blocks may admit repeated and possibly complex eigenvalues, variations of the result above are applicable. In the special case where there is a centralized controller which receives information from multiple sensors (under stabilizability and joint detectability), even in the presence of unbounded noise (such as Gaussian noise), to achieve asymptotic stability, it suffices to have the average total rate be greater than $\sum_{|\lambda_i| > 1} \log_2(|\lambda_i|)$ (see Johnston and Yüksel 2014, Matveev and Savkin 2008, and Yüksel and

Başar 2013). For the case with a single sensor and single controller, this result has been studied extensively in networked control [see the chapter on *Quantized Control and Data-Rate Constraints* in the Encyclopedia] under a plethora of setups (Nair et al. 2007; Franceschetti and Minero 2014; Kawan 2013; Yüksel 2016).

Information Structures and Their Design in Decentralized Stochastic Control and Some Topological Properties

As observed earlier, a decentralized control system, or a team, consists of a collection of decision makers/agents acting together to optimize a common cost function, but not necessarily sharing all the available information. Teams whose initial states, observations, cost function, or the evolution dynamics are random or are disturbed by some external noise processes are called *stochastic teams*. We now introduce the characterizations as laid out by Witsenhausen, through his *intrinsic model* (Witsenhausen 1975), with the presentation in Yüksel and Saldi (2017). Suppose there is a pre-defined order in which the decision makers act. Such systems are called *sequential systems*. The action and measurement spaces are standard Borel spaces, that is, Borel subsets of complete, separable, and metric spaces. The *intrinsic model* for sequential teams can be defined as follows:

- There exists a collection of *measurable spaces* $\{(\Omega, \mathcal{F}), (\mathbb{U}^i, \mathcal{U}^i), (\mathbb{Y}^i, \mathcal{Y}^i), i \in \mathcal{N}\}$, with $\mathcal{N} = \{1, 2, \dots, N\}$, specifying the system's distinguishable events, and control and measurement spaces. In this model (described in discrete time), any action applied at any given time is regarded as applied by an individual decision maker (DM), who acts only once. The pair $(\Omega, \mathcal{F})$ is a measurable space (on which an underlying probability may be defined). The pair $(\mathbb{U}^i, \mathcal{U}^i)$ denotes the measurable space from which the action, u^i , of decision maker i is selected.

The pair $(\mathbb{Y}^i, \mathcal{Y}^i)$ denotes the measurable observation/measurement space.

- There is a *measurement constraint* to establish the connection between the observation variables and the system's distinguishable events. The $\mathbb{Y}^i$ -valued observation variables are given by $y^i = h^i(\omega, \mathbf{u}^{[1, i-1]})$, where $\mathbf{u}^{[1, i-1]} := \{u^k, k \leq i-1\}$, h^i s are measurable functions and u^k denotes the action of DM k . Hence, y^i induces $\sigma(y^i)$ over $\Omega \times \prod_{k=1}^{i-1} \mathbb{U}^k$.
- The set of admissible control laws $\underline{\gamma} = \{\gamma^1, \gamma^2, \dots, \gamma^N\}$, also called *designs* or *policies*, are measurable control functions, so that $u^i = \gamma^i(y^i)$. Let Γ^i denote the set of all admissible policies for DM i and let $\Gamma = \prod_i \Gamma^i$.
- There is a *probability measure* $\mathbb{P}$ on $(\Omega, \mathcal{F})$ describing the probability space on which the system is defined.

Under this intrinsic model, a sequential team problem is *dynamic* if the information available to at least one DM is affected by the action of at least one other DM. A team problem is *static*, if for every decision maker, the information available is only affected by exogenous disturbances; that is, no other decision maker can affect the information at any given decision maker.

Information structures can additionally be categorized as *classical*, *quasi-classical*, or *nonclassical*. An information structure (IS) $\{y^i, i \in \mathcal{N}\}$ is *classical* if y^i contains all of the information available to DM k for $k < i$. An IS is *quasi-classical* or *partially nested*, if whenever u^k , for some $k < i$, affects y^i through the measurement function h^i , y^i contains y^k . An IS which is not partially nested is *nonclassical*.

Witsenhausen (1988) has shown that a large class of sequential team problems admit an equivalent information structure which is static, under an absolute continuity condition. For partially nested (or quasi-classical) information structures, static reduction of a dynamic team has been studied by Ho and Chu in the specific context of LQG systems (Ho and Chu 1972) and for a class of nonlinear systems satisfying restrictive invertibility properties (Ho and Chu 1973). Thus,

studying static teams is sufficient for investigating a large class of dynamic teams.

Let, as before, $\underline{\gamma} = \{\gamma^1, \dots, \gamma^N\}$, and let $\Gamma = \prod_i^N \Gamma^i$ be the set of admissible policies for the team with N DMs. Assume an expected cost function is defined as

$$J(\underline{\gamma}) = E^\gamma[c(\omega, \mathbf{u})], \quad (8)$$

for some Borel measurable cost function $c : \Omega \times \prod_{k=1}^N \mathbb{U}^k \rightarrow \mathbb{R}$ and where $E^\gamma[c(\omega, \mathbf{u})] := E[c(\omega, \gamma^1(y^1), \dots, \gamma^N(y^N))]$. Here, we have the notation $\mathbf{u} := \{u^i, i \in \mathcal{N}\}$.

Definition 3 For a given stochastic team problem with a given information structure, a policy (strategy) N -tuple $\underline{\gamma}^* := (\gamma^{1*}, \dots, \gamma^{N*}) \in \Gamma$ is *optimal (team-optimal solution)* if

$$J(\underline{\gamma}^*) = \inf_{\underline{\gamma} \in \Gamma} J(\underline{\gamma}) =: J^*.$$

Definition 4 Person-by-person optimal solution. For a given N -DM stochastic team with a fixed information structure, an N -tuple of strategies $\underline{\gamma}^* := (\gamma^{1*}, \dots, \gamma^{N*})$ constitutes a *person-by-person optimal* (pbp optimal) solution if, for all $\beta \in \Gamma^i$ and all $i \in \mathcal{N}$, the following inequalities hold:

$$J^* := J(\underline{\gamma}^*) \leq J(\underline{\gamma}^{-i*}, \beta),$$

where $(\underline{\gamma}^{-i*}, \beta) := (\gamma^{1*}, \dots, \gamma^{(i-1)*}, \beta, \gamma^{(i+1)*}, \dots, \gamma^{N*})$.

On teams with finitely many decision makers, Marschak (1955) studied optimal static teams, and Radner (1962) developed foundational results on optimality, establishing connections between person-by-person optimality, stationarity, and team optimality. Radner's results were generalized in Krainak et al. (1982) by relaxing optimality conditions. A summary of these results is that in the context of static team problems, convexity of the cost function, subject to minor regularity conditions such as continuous differentiability or local finiteness properties, may suffice for the global optimality of person-by-person-optimal solutions. In the particular case for LQG (Linear Quadratic Gaussian) static teams, this result leads to the optimality of

linear policies (Radner 1962), which also applies for dynamic LQG problems under specific information structures (to be discussed further below) (Ho and Chu 1972). These results are applicable for static teams with finite number of decision makers.

Information structures crucially affect whether there exists an optimal team policy or not. We refer the reader to Gupta et al. (2015), Yüksel and Saldi (2017), Charalambous and Ahmed (2017), and Yüksel (2018) for existence and associated geometric properties of information structures (and Yüksel and Başar (2013) for a nonexistence result), Saldi et al. (2017) for rigorous approximation results of optimal stochastic team problems with standard Borel spaces with those that are defined on finite spaces obtained through the quantization of measurement and control action spaces, and Lehrer et al. (2010) and Yüksel and Başar (2013, Chapter 4) for comparison of information structures in decentralized stochastic control.

We refer the reader to Yüksel and Linder (2012) and Yüksel and Başar (2013, Chapter 4) for further topological properties on information structures such as continuity of the optimal expected costs in the information structures and the resulting existence results on optimal information structures and channels/quantizers.

Similar to the discussion in section “**Connections with Information Theory**”, information theoretic relaxations have also been comprehensively studied, where as noted earlier, instead of single realizations, arbitrarily long sequences of random variables under conditional independence and marginal constraints on the empirical measures and mutual information constraints have been considered; these can be used to arrive at lower bounds for optimal decentralized stochastic control problems (Grover et al. 2015; Treust and Oechtering 2018). We refer the reader to Kulkarni and Coleman (2014), Anantharam and Borkar (2007) and Yüksel and Saldi (2017) for further properties on relaxations of information structures both under information theoretic or probabilistic constraints.

Along a further related research direction, information structures with respect to sparsity

constraints have been investigated in a number of publications including Rotkowitz and Lall (2006), Bamieh and Voulgaris (2005), Sabau and Martins (2012), Lin et al. (2012), Lessard and Lall (2011), and Jovanovic (2010). A detailed review for the literature and solution approaches for norm-optimal control as well as optimal stochastic dynamic teams is provided in Mahajan et al. (2012).

Large-Scale Systems and Mean-Field Theory

When the number of decision makers is large, often the analysis becomes significantly simpler under various technical conditions with respect to the structure of the cost function and the coupling among the decision makers. This phenomenon arises in *mean-field equilibrium* problems where the number of decision makers is unbounded (see Huang et al. 2007, 2006; Lasry and Lions 2007): Mean-field games can be viewed as limit models of symmetric non-zero-sum noncooperative N -player games with a mean-field interaction as $N \rightarrow \infty$. The mean-field approach designs policies for both cases of games with infinitely many players, as well as games with very large number of players where the equilibrium policies for the former are shown to be ϵ -equilibria for the latter.

These results, in the context of team problems, involve person-by-person-optimal policies, but these do not guarantee ϵ -global optimality among all policies in general. However, mean-field team problems have also been studied: social optima for mean-field LQG control problems have been considered in Huang et al. (2012) and Wang and Zhang (2017). In Arabneydi and Mahajan (2015), a setup is considered where the decision makers share some information on the mean field in the system, and through showing that the performance of a corresponding centralized system can be attained under a restricted decentralized information structure, global optimality is established. There exist contributions where games with finitely many players are studied, their equilibrium solutions are obtained, and the limit is taken (see, e.g., Fischer 2017; Biswas 2015; Apostathis et al. 2017; Lacker 2016). On a related

direction, Mahajan et al. (2013) and Sanjari and Yüksel (2019) have shown that, under sufficient conditions, a sequence of optimal policies for teams with N number of decision makers as $N \rightarrow \infty$ converges to a team optimal policy for the static team with countably infinite number of decision makers, where the latter establishes the optimality of symmetric (i.e., identical for each DM) policies as well as existence of optimal team policies for both finite and infinite DM setups.

Information Design for Stochastic Games and Networked Control with Non-aligned DMs

We do not have sufficient space to discuss the important subject of stochastic games; however, it would be essential in a short chapter such as this one to at least emphasize that the value of information in stochastic games is a very subtle subject: While the value of information in stochastic teams is always nonnegative, the value of additional information, in the sense of the impact of the additional information on the equilibria values, can be negative. This is due to the fact that equilibrium is a fragile notion arising from consistency conditions in fixed points under *joint* best-response maps and the unilateral best-response of a DM does not lead to a consequential answer on the impact of the additional information on the equilibrium values.

Thus, in the context of information design under information constraints for stochastic games, we note a crucial difference: for team problems, although it is difficult to obtain optimal solutions under general information structures, it is apparent that more information provided to any of the decision makers does not negatively affect the utility of the players. There is also a well-defined partial order of information structures as studied by Blackwell (1953), Lehrer et al. (2010) and Yüksel and Başar (2013). However, for general non-zero-sum game problems, informational aspects are challenging to address; more information can have negative effects on some or even all of the players in a system; see, e.g., Hirschleifer (1971) (see also Peski (2008) for a rather precise result

on comparison of information structures for zero-sum games, where a partial order is established.

In the context of information structure design for games under bit rate constraints, the problems of signaling games and cheap talk are concerned with a class of Bayesian games where a privately informed player (encoder or sender) transmits information (signal) to another player (decoder or receiver). In these problems, the objective functions of the players are not necessarily aligned, unlike the case in classical communication theoretic and control theoretic problems. The cheap talk problem was introduced in the economics literature in Crawford and Sobel (1982), who obtained the striking result that under some technical conditions on the cost functions, the cheap talk problem only admits equilibria that involve quantized encoding policies. This is in significant contrast to the usual communication/information theoretic case where the objective functions are aligned. Therefore, as indicated in Crawford and Sobel (1982), the similarity of players' interests (objective functions) indicates to which extent the information should be revealed by the encoder; Saritaş et al. (2017) extended the analysis for more general sources. We note that, unlike Crawford and Sobel's simultaneous Nash equilibrium formulation, if one considers a Stackelberg formulation (see Başar and Olsder (1999, p.133) for a definition), then the problem would reduce to a modified optimal communication problem since the encoder would be committed a priori; this is known as a *Bayesian persuasion game* (see Kamenica and Gentzkow (2011) and relevant contributions from the control theoretic literature includes Saritaş et al. 2017, Farokhi et al. 2017, Akyol et al. 2017, Sayın et al. 2017, and Treust and Tomala 2018).

Computational Complexity

The focus of this chapter is with regard to information and communication requirements and complexity, and not on computational complexity, which is a rather comprehensive field. Nonetheless, we refer the reader to the computational challenges that have been reflected in the celebrated counterexample of Witsenhausen

(1968). Tsitsiklis and Athans (1985) have observed that from a computational complexity viewpoint, obtaining optimal solutions for a class of such communication protocol design problems is non-tractable. We also note that although finding optimal solutions for finite models has been shown to be NP-complete in Papadimitriou and Tsitsiklis (1986), there exists a plethora of numerical results as well as results rigorously establishing asymptotic optimality of finite/quantized decentralized control policies converging to the optimal solutions (Saldi et al. 2017). A related recent contribution is Olshevsky (2019) which considers problem instances (with arbitrary distributions in the Witsenhausen's counterexample setup) with positive and negative results on complexity. Bernstein et al. (2002) studies the complexity of decentralized stochastic control problems under various setups, in particular, under a decentralized partially observed setup which is aligned with the models considered in this chapter.

Summary

In this text, we discussed the problem of communication complexity in networked control systems. Our analysis considered both cost minimization and controllability/reachability problems subject to information constraints. We also discussed communication complexity in distributed computing as has been studied in the computer science community and provided a brief discussion on the information theoretic approaches for such problems together with structural results. We also provided an overview of some recent research results on information structures in decentralized stochastic control.

Cross-References

- [Game Theory: A General Introduction and a Historical Overview](#)
- [Information-Based Control of Networked Systems](#)
- [Networked Control Systems: Estimation and Control over Lossy Networks](#)

► **Randomized Methods for Control of Uncertain Systems**

Bibliography

- Akyol E, Langbort C, Başar T (2017) Information-theoretic approach to strategic communication as a hierarchical game. *Proc IEEE* 105(2):205–218. <https://doi.org/10.1109/JPROC.2016.2575858>
- Anantharam V, Borkar VS (2007) Common randomness and distributed control: a counterexample. *Syst Control Lett* 56(7):568–572
- Arabneydi J, Mahajan A (2015) Team-optimal solution of finite number of mean-field coupled LQG subsystems. In: *IEEE 54th annual conference on decision and control (CDC)*, pp 5308–5313
- Arapostathis A, Biswas A, Carroll J (2017) On solutions of mean field games with ergodic cost. *Journal de Mathématiques Pures et Appliquées* 107(2):205–251
- Bamieh B, Voulgaris P (2005) A convex characterization of distributed control problems in spatially invariant systems with communication constraints. *Syst Control Lett* 54:575–583
- Bansal R, Başar T (1989) Simultaneous design of measurement and control strategies in stochastic systems with feedback. *Automatica* 45:679–694
- Başar T, Olsder G (1999) *Dynamic noncooperative game theory*. Classics in applied mathematics. SIAM, Philadelphia
- Bernstein DS, Givan R, Immerman N, Zilberstein S (2002) The complexity of decentralized control of Markov decision processes. *Math Oper Res* 27(4):819–840
- Biswas A (2015) Mean field games with ergodic cost for discrete time Markov processes. *arXiv preprint arXiv:151008968*
- Blackwell D (1953) Equivalent comparison of experiments. *Ann Math Stat* 24:265–272
- Charalambous CD, Ahmed NU (2017) Centralized versus decentralized optimization of distributed stochastic differential decision systems with different information structures-part I: A general theory. *IEEE Trans Autom Control* 62(3):1194–1209
- Cover TM, Thomas JA (1991) *Elements of information theory*. Wiley, New York
- Crawford VP, Sobel J (1982) Strategic information transmission. *Econometrica* 50:1431–1451
- Derpich MD, Ostergaard J (2012) Improved upper bounds to the causal quadratic rate-distortion function for gaussian stationary sources. *IEEE Trans Inf Theory* 58(5):3131–3152
- Farokhi F, Teixeira A, Langbort C (2017) Estimation with strategic sensors. *IEEE Trans Autom Control* 62(2):724–739. <https://doi.org/10.1109/TAC.2016.2571779>
- Fischer M (2017) On the connection between symmetric N-player games and mean field games. *Ann Appl Probab* 27(2):757–810
- Franceschetti M, Minero P (2014) Elements of information theory for networked control systems. In: *Information and control in networks*. Springer, New York, pp 3–37
- Gamal AE, Kim YH (2012) *Network information theory*. Cambridge University Press, Cambridge
- Giridhar A, Kumar P (2006) Toward a theory of in-network computation in wireless sensor networks. *IEEE Commun Mag* 44:98–107
- Grover P, Wagner AB, Sahai A (2015) Information embedding and the triple role of control. *IEEE Trans Inf Theory* 61(4):1539–1549
- Gupta A, Yüksel S, Başar T, Langbort C (2015) On the existence of optimal policies for a class of static and sequential dynamic teams. *SIAM J Control Optim* 53:1681–1712
- György A, Linder T (2000) Optimal entropy-constrained scalar quantization of a uniform source. *IEEE Trans Inf Theory* 46:2704–2711
- György A, Linder T, Lugosi G (2008) Tracking the best quantizer. *IEEE Trans Inf Theory* 54:1604–1625
- Hirshleifer J (1971) The private and social value of information and the reward to inventive activity. *Am Econ Rev* 61:561–574
- Ho YC, Chu KC (1972) Team decision theory and information structures in optimal control problems – part I. *IEEE Trans Autom Control* 17:15–22
- Ho YC, Chu KC (1973) On the equivalence of information structures in static and dynamic teams. *IEEE Trans Autom Control* 18(2):187–188
- Huang M, Caines PE, Malhamé RP (2006) Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. *Commun Inf Syst* 6:221–251
- Huang M, Caines PE, Malhamé RP (2007) Large-population cost-coupled LQG problems with nonuniform agents: individual-mass behavior and decentralized ϵ -Nash equilibria. *IEEE Trans Autom Control* 52:1560–1571
- Huang M, Caines PE, Malhamé RP (2012) Social optima in mean field LQG control: centralized and decentralized strategies. *IEEE Trans Autom Control* 57(7):1736–1751
- Johnston A, Yüksel S (2014) Stochastic stabilization of partially observed and multi-sensor systems driven by unbounded noise under fixed-rate information constraints. *IEEE Trans Autom Control* 59:792–798
- Jovanovic MR (2010) On the optimality of localized distributed controllers. *Int J Syst Control Commun* 2:82–99
- Kamenica E, Gentzkow M (2011) Bayesian persuasion. *American Economic Review* 101(6):2590–2615. <https://doi.org/10.1257/aer.101.6.2590>
- Kawan C (2013) *Invariance entropy for deterministic control systems*. Springer, New York
- Kostina V, Hassibi B (2019) Rate-cost tradeoffs in control. *IEEE Trans Autom Control* 64:4525–4540
- Krainak J, Speyer JL, Marcus S (1982) Static team problems – part I: Sufficient conditions and the exponential cost criterion. *IEEE Trans Autom Control* 27:839–848

- Kulkarni AA, Coleman TP (2014) An optimizer's approach to stochastic control problems with nonclassical information structures. *IEEE Trans Autom Control* 60(4):937–949
- Kushilevitz E, Nisan N (2006) Communication complexity, 2nd edn. Cambridge University Press, Cambridge
- Lacker D (2016) A general characterization of the mean field limit for stochastic differential games. *Probab Theory Relat Fields* 165(3–4):581–648
- Lasry JM, Lions PL (2007) Mean field games. *Jpn J Math* 2:229–260
- Lehrer E, Rosenberg D, Shmaya E (2010) Signaling and mediation in games with common interests. *Games Econ Behav* 68:670–682
- Lessard L, Lall S (2011) A state-space solution to the two-player decentralized optimal control problem. In: Annual Allerton conference on communication, control and computing, Monticello
- Lin F, Fardad M, Jovanovic MR (2012) Optimal control of vehicular formations with nearest neighbor interactions. *IEEE Trans Autom Control* 57:2203–2218
- Ma N, Ishwar P (2011) Some results on distributed source coding for interactive function computation. *IEEE Trans Inf Theory* 57:6180–6195
- Mahajan A, Teneketzis D (2009) Optimal design of sequential real-time communication systems. *IEEE Trans Inf Theory* 55:5317–5338
- Mahajan A, Martins N, Rotkowitz M, Yüksel S (2012) Information structures in optimal decentralized control. In: IEEE conference on decision and control, Hawaii
- Mahajan A, Martins NC, Yüksel S (2013) Static LQG teams with countably infinite players. In: 2013 IEEE 52nd annual conference on decision and control (CDC). IEEE, pp 6765–6770
- Marschak J (1955) Elements for a theory of teams. *Manag Sci* 1:127–137
- Matveev AS, Savkin AV (2008) Estimation and control over communication networks. Birkhäuser, Boston
- Nair GN, Fagnani F, Zampieri S, Evans JR (2007) Feedback control under data constraints: an overview. In: Proceedings of the IEEE, pp 108–137
- Nayyar A, Mahajan A, Teneketzis D (2014) The common-information approach to decentralized stochastic control. In: Como G, Bernhardsson B, Rantzer A (eds) Information and control in networks. Springer, Switzerland
- Nemirovsky A, Yudin D (1983) Problem complexity and method efficiency in optimization. Wiley-Interscience, New York
- Olshevsky A (2019) On the inapproximability of the discrete witsenhausen problem. *IEEE Control Syst Lett* 3:529–534
- Orlitsky A, Roche JR (2001) Coding for computing. *IEEE Trans Inf Theory* 47:903–917
- Papadimitriou C, Tsitsiklis J (1986) Intractable problems in control theory. *SIAM J Control Optim* 24(4):639–654
- Peski M (2008) Comparison of information structures in zero-sum games. *Games Econ Behav* 62(2):732–735
- Radner R (1962) Team decision problems. *Ann Math Stat* 33:857–881
- Raginsky M, Rakhlin A (2011) Information-based complexity, feedback and dynamics in convex programming. *IEEE Trans Inf Theory* 57:7036–7056
- Rotkowitz M, Lall S (2006) A characterization of convex problems in decentralized control. *IEEE Trans Autom Control* 51:274–286
- Sabau S, Martins NC (2012) Stabilizability and norm-optimal control design subject to sparsity constraints. arXiv preprint arXiv:12091123
- Saldi N, Yüksel S, Linder T (2017) Finite model approximations and asymptotic optimality of quantized policies in decentralized stochastic control. *IEEE Trans Autom Control* 62:2360–2373
- Sanjari S, Yüksel S (2019) Optimal solutions to infinite-player stochastic teams and mean-field teams. *IEEE Trans Autom Control*
- Sarıtaş S, Yüksel S, Gezici S (2017) Quadratic multidimensional signaling games and affine equilibria. *IEEE Trans Autom Control* 62(2):605–619. <https://doi.org/10.1109/TAC.2016.2578843>
- Sayın MO, Akyol E, Başar T (2017) Hierarchical multi-stage Gaussian signaling games: strategic communication and control. arXiv preprint arXiv:160909448
- Stavrou PA, Østergaard J, Charalambous CD (2018) Zero-delay rate distortion via filtering for vector-valued Gaussian sources. *IEEE J Sel Top Signal Process* 12(5):841–856
- Tanaka T, Kim KK, Parrilo PA, Mitter SK (2016) Semidefinite programming approach to gaussian sequential rate-distortion trade-offs. *IEEE Trans Autom Control* 62(4):1896–1910
- Tatikonda S, Sahai A, Mitter S (2004) Stochastic linear control over a communication channels. *IEEE Trans Autom Control* 49:1549–1561
- Teneketzis D (1979) Communication in decentralized control. PhD Dissertation, M.I.T.
- Teneketzis D (2006) On the structure of optimal real-time encoders and decoders in noisy communication. *IEEE Trans Inf Theory* 52:4017–4035
- Trust ML, Oechtering TJ (2018) Optimal control designs for vector-valued witsenhausen counterexample setups. In: 56th annual Allerton conference on communication, control, and computing (Allerton). IEEE, pp 532–537
- Trust ML, Tomala T (2018) Information-theoretic limits of strategic communication. arXiv preprint arXiv:180705147
- Tsitsiklis J, Athans M (1985) On the complexity of decentralized decision making and detection problems. *IEEE Trans Autom Control* 30:440–446
- Tsitsiklis J, Luo ZQ (1987) Communication complexity of convex optimization. *J Complexity* 3:231–243
- Walrand JC, Varaiya P (1983) Optimal causal coding-decoding problems. *IEEE Trans Inf Theory* 19:814–820
- Wang BC, Zhang JF (2017) Social optima in mean field linear-quadratic-Gaussian models with Markov jump parameters. *SIAM J Control Optim* 55(1):429–456

- Witsenhausen H (1968) A counterexample in stochastic optimal control. *SIAM J Control* 6:131–147
- Witsenhausen H (1975) The intrinsic model for discrete stochastic control: some open problems. *Lect Notes Econ Math Syst* 107:322–335. Springer
- Witsenhausen HS (1976) The zero-error side information problem and chromatic numbers. *IEEE Trans Inf Theory* 22:592–593
- Witsenhausen HS (1979) On the structure of real-time source coders. *Bell Syst Tech J* 58:1437–1451
- Witsenhausen H (1988) Equivalent stochastic control problems. *Math Control Signals Syst* 1:3–11
- Wong WS (2009) Control communication complexity of distributed control systems. *SIAM J Control Optim* 48:1722–1742
- Wong WS, Baillieul J (2012) Control communication complexity of distributed actions. *IEEE Trans Autom Control* 57:2731–2745
- Wyner AD, Ziv J (1976) The rate-distortion function for source coding with side information at the decoder. *IEEE Trans Inf Theory* 22:1–11
- Yao ACC (1979) Some complexity questions related to distributive computing. In: *Proceedings of the of the 11th annual ACM symposium on theory of computing*
- Yüksel S (2013) On optimal causal coding of partially observed Markov sources in single and multi-terminal settings. *IEEE Trans Inf Theory* 59:424–437
- Yüksel S (2016) Stationary and ergodic properties of stochastic non-linear systems controlled over communication channels. *SIAM J Control Optim* 54:2844–2871
- Yüksel S (2018) Witsenhausen's standard form revisited: a general dynamic program for decentralized stochastic control and existence results. In: *2018 IEEE conference on decision and control (CDC)*. IEEE, pp 5051–5056
- Yüksel S (2019) A note on the separation of optimal quantization and control policies in networked control. *SIAM J Control Optim* 57(1):773–782
- Yüksel S, Başar T (2013) *Stochastic networked control systems: stabilization and optimization under information constraints*. Springer, New York
- Yüksel S, Linder T (2012) Optimization and convergence of observation channels in stochastic control. *SIAM J Control Optim* 50:864–887
- Yüksel S, Saldi N (2017) Convex analysis in decentralized stochastic control, strategic measures and optimal solutions. *SIAM J Control Optim* 55:1–28

Information-Based Multi-agent Systems

Wing Shing Wong¹ and John Baillieul²

¹Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, China

²College of Engineering, Boston University, Boston, MA, USA

Abstract

Multi-agent systems are encountered in nature (animal groups), in various domains of technology (multi-robot networks, mixed robot-human teams), and in various human activities (such as dance and team athletics). Information exchange among agents ranges from being incidentally important to crucial in such systems. Several systems in which information exchange among the agents is either a primary goal or a primary enabler are discussed briefly below. Specific topics include power management in wireless communication networks, data rate constraints, the complexity of distributed control, robotics networks and formation control, action-mediated communication, and multi-objective distributed systems.

Keywords

Distributed control · Information constraints · Multi-agent systems

Synonyms

[Information-Based Control of Networked Systems](#)

Information-Based Control of Networked Systems

► Information-Based Multi-agent Systems

Introduction

The role of information patterns in the decentralized control of multi-agent systems has been

studied in different theoretical contexts for more than five decades. The paper Ho (1972) provides references to early work in this area. While research on distributed decision-making has continued, a large body of recent research on robotic networks has brought new dimensions of geometric aspects of information patterns to the forefront (Bullo et al. 2009). At the same time, machine intelligence, machine learning, machine autonomy, and theories of operation of mixed teams of humans and robots have considerably extended the intellectual frontiers of information-based multi-agent systems (Baillieul et al. 2012). A further important development has been the study of action-mediated communication and the recently articulated theory of *control communication complexity* (Wong and Baillieul 2012). These developments may shed light on nonverbal forms of communication among biological organisms (including humans) and on the intrinsic energy requirements of information processing.

In conventional decentralized control, the control objective is usually well-defined and known to all agents. Multi-agent information-based control encompasses a broader scenario where the objective can be agent-dependent and is not necessarily explicitly announced to all. For illustration, consider the power control problem in wireless communication – one of the earliest engineering systems that can be regarded as multi-agent information-based. It is common that multiple transmitter-receiver communication pairs share the same radio-frequency band and the transmission signals interfere with each other. The power control problem searches for feedback control for each transmitter to set its power level. The goal is for each transmitter to achieve targeted signal-to-interference ratio (SIR) level by using information of the observed levels at the intended receiver only.

A popular version of the power control problem (Foschini and Miljanic 1993) defines each individual objective target level by means of a requirement threshold, known only to the intended transmitter. As SIR measurements naturally reside on a receiver, the observed

SIR needs to be communicated back to the transmitter. For obvious reasons, the bandwidth for such communication is limited. The resulting model fits the bill of multi-agent information-based control. In Sung and Wong (1999), a tri-state power control strategy is proposed so that the power control outputs are either increased or decreased by a fixed dB or no change at all. Convergence of the feedback algorithm was shown using a Lyapunov-like function.

This encyclopedia entry surveys key topics related to multi-agent information-based control systems, including control complexity, control with data rate constraints, robotic networks and formation control, action-mediated communication, and multi-objective distributed systems.

Control Complexity

In information-based distributed control systems, how to efficiently share computational and communication resources is a fundamental issue. One of the earliest investigations on how to schedule communication resources to support a network of sensors and actuators is discussed in Brockett (1995). The concept of *communication sequencing* was introduced to describe how the communication channel is utilized to convey feedback control information in a network consisting of interacting subsystems. In Brockett (1997) the concept of *control attention* was introduced to provide a measure of the complexity of a control law against its performance. As attention is a shared, limited resource, the goal is to find minimum attention control. Another approach to gauge control complexity in a distributed system is by means of the minimum amount of communicated data required to accomplish a given control task.

Control with Data Rate Constraints

A fundamental challenge in any control implementation in which system components communicate with each other over communication links is ensuring that the channel capacity is large enough to deal with the fastest time constants among the system components. In a single

agent system, the so-called data rate theorem has been formulated in various ways to understand the constraints imposed between the sensor and the controller and between the controller and the actuator. Extensions to this fundamental result have been focused on addressing similar problems in the network control system context. Information on such extensions in the distributed control setting can be found in Nair and Evans (2004) and Yüksel and Başar (2007).

Robotic Networks and Formation Control

The defining characteristic of robotic networks within the larger class of multi-agent systems is the centrality of spatial relationships among network nodes. Graph theory has been shown to provide a generally convenient mathematical language in which to describe spatial concepts, and it is the key to understanding spatial rigidity related to the control of formations of autonomous vehicles (Anderson et al. 2008), or in flocking systems (Leonard et al. 2012), or in consensus problems (Su and Huang 2012), or in rendezvous problems (Cortés et al. 2006). For these distributed control research topics, readers can consult other sections in this encyclopedia for a comprehensive reference list.

Much of the recent work on formation control has included information limitation considerations. For consensus problems, for example, Olfati-Saber and Murray (2004) introduced a sensing cost constraint, and in Ren and Beard (2005) information exchange constraints are considered, and in Yu and Wang (2010) communication delays are explicitly modeled.

Action-Mediated Communication

Biological organisms communicate through motion. Examples of this include prides of lions or packs of wolves whose pursuit of prey is a cooperative effort and competitive team athletics in the case of humans. Recent research has been aimed at developing a theoretical foundation of action-mediated communication. Communication protocols for motion-based signaling between mobile robots have been developed (Raghuathan and Baillieul 2009),

and preliminary steps toward a theory of artistic expression through controlled movement in dance have been reported in Baillieul and Özcimder (2012). Motion-based communication of this type involves specially tailored motion description languages in which sequences of motion primitives are assembled with the objective of conveying artistic intent while minimizing the use of limited energy resources in carrying out the movement. These motion primitives constitute the *alphabet* that enables communication, and physical constraints on the motions define the grammatical rules that govern the ways in which motion sequences may be constructed.

Research on action-mediated communication helps illustrate the close connection between control and information theory. Further discussion of the deep connection between the two can be found, for example, in Park and Sahai (2011), which argues for the equivalence between the stabilization of a distributed linear system and the capacity characterization in linear network coding.

Multi-objective Distributive Systems

In a multi-agent system, agents may aim to carry out individual objectives. These objectives can either be cooperatively aligned (such as in a cooperative control setting) or may contend antagonistically (such as in a zero-sum game setting). In either case, a common assumption is that the objective functions are a priori known to all agents. However, in many practical applications, agents do not know the objectives of other agents, at least not precisely. For example, in the power control problem alluded to earlier, the signal-to-interference requirement of a user may be unknown to other users. Yet this does not prevent the possibility of deriving convergent algorithms to allow the joint goals be achieved.

The issue of unknown objective in a multi-agent system is formally analyzed in Wong (2009) via the introduction of choice-based actions. In an open access network, objectives of an individual agent may be known only partially, via the form of a random distribution in some cases. In order to achieve a joint control objective

in general, some communication via the system is required if there is no side communication channel. A basic issue is how to measure the minimum amount of information exchange that is required to perform a specific control task. Motivated by the idea of communication complexity in computer science, the idea of control communication complexity was introduced in Wong (2009), which can provide such a measure. The idea was further applied to nonlinear systems known as the Heisenberg system (or Brockett Integrator) in Wong and Baillieul (2009).

In some special cases, control objectives can be achieved without any communication among the agents. For systems with bilinear input-output mapping, including the Brockett Integrator, it is possible to derive conditions that guarantee this property (Wong and Baillieul 2012). Moreover, for quadratic type of control cost, it is possible to compute the optimal control cost. Similar results can be extended to linear systems as discussed in Liu et al. (2013). This circle of ideas is connected to the so-called *standard parts* problem as investigated in Baillieul and Wong (2009). Another connection is to *correlated equilibrium* problems that have been recently studied by game theorists (Shoham and Leyton-Brown 2009).

Summary and Future Directions

Information based control of multi-agent systems is concerned with the design and effective use of information channels for enabling multiple agents to collaboratively achieve objectives that no single agent could accomplish alone. It focuses in part on systems in which objectives are not explicitly known to individual agents but are nevertheless achievable through the dynamic interactions of the agents. Specific topics discussed above include control communication complexity, control with information constraints, robotic networks, action mediated control, and multi-agent distributed systems. Protocols for information sharing agents can be either cooperative or adversarial or mixtures of both. Research is now needed to develop new theories

of dynamic games that focus on information flows in multi-agent systems.

Cross-References

- Consensus of Complex Multi-agent Systems
- Information and Communication Complexity of Networked Control Systems
- Networked Systems

Bibliography

- Altman E, Avrachenkov K, Menache I, Miller G, Prabhu BJ, Schwartz A (2009) Dynamic discrete power control in cellular networks. *IEEE Trans Autom Control* 54(10):2328–2340
- Anderson B, Yu C, Fidan B, Hendrickx J (2008) Rigid graph control architectures for autonomous formations. *IEEE Control Syst Mag* 28(6):48–63. <https://doi.org/10.1109/MCS.2008.929280>
- Baillieul J, Özcimder K (2012) The control theory of motion-based communication: problems in teaching robots to dance. In: *Proceedings of the American control conference (ACC)*, 27–29 June 2012, Montreal, pp 4319–4326
- Baillieul J, Wong WS (2009) The standard parts problem and the complexity of control communication. In: *Proceedings of the 48th IEEE conference on decision and control*, held jointly with the 28th Chinese control conference, 16–18 Dec 2009, Shanghai, pp 2723–2728
- Baillieul J, Leonard NE, Morgansen KA (2012) Interaction dynamics: the interface of humans and smart machines. *Proc IEEE* 100(3):776–801. <https://doi.org/10.1109/JPROC.s011.2180055>
- Brockett RW (1995) Stabilization of motor networks. In: *Proceedings of the 34th IEEE conference on decision and control*, 13–15 Dec 1995, New Orleans, pp 1484–1488
- Brockett RW (1997) Minimum attention control. In: *Proceedings of the 36th IEEE conference on decision and control*, 10–12 Dec 1997, San Diego, pp 2628–2632
- Bullo F, Cortés J, Martínez S (2009) *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, Princeton
- Cortés J, Martínez S, Bullo F (2006) Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Trans Autom Control* 51(8):1289–1298
- Foschini GJ, Miljanic Z (1993) A simple distributed autonomous power control algorithm and its convergence. *IEEE Trans Veh Technol* 42(4):641–646
- Ho YC (1972) Team decision theory and information structures in optimal control problems—Part I. *IEEE Trans Autom Control* 17(1):15–22

- Leonard NE, Young G, Hochgraf K, Swain D, Chen W, Marshall S (2012) In the dance studio: analysis of human flocking. In: Proceedings of the 2012 American control conference, 27–29 June 2012, Montreal, pp 4333–4338
- Liu ZC, Wong WS, Guo G (2013) Cooperative control of linear systems with choice actions. In: Proceedings of the American control conference, 17–19 June 2013, Washington, D.C., pp 5374–5379
- Nair GN, Evans RJ (2004) Stabilisability of stochastic linear systems with finite feedback data rates. *SIAM J Control Optim* 43(2):413–436
- Olfati-Saber R, Murray RM (2004) Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans Autom Control* 49(9):1520–1533
- Park SY, Sahai A (2011) Network coding meets decentralized control: capacity-stabilizability equivalence. In: Proceedings of the 50th IEEE conference on decision and control and European control conference, 12–15 Dec 2011, Orlando, pp 4817–4822
- Ragunathan D, Baillieul J (2009) Motion based communication channels between mobile robots—a novel paradigm for low bandwidth information exchange. In: Proceeding of the IEEE/RSJ international conference on intelligent robots and systems, 11–15 Oct 2009, St. Louis, pp 702–708
- Ren W, Beard RW (2005) Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Trans Autom Control* 50(5):655–661
- Shoham Y, Leyton-Brown K (2009) Multiagent systems: algorithmic, game-theoretic, and logical foundations. Cambridge University Press, New York. ISBN 978-0-521-89943-7
- Su Y, Huang J (2012) Two consensus problems for discrete-time multi-agent systems with switching network topology. *Automatica* 48(9):1988–1997
- Sung CW, Wong WS (1999) A distributed fixed-step power control algorithm with quantization and active link quality protection. *IEEE Trans Veh Technol* 48(2):553–562
- Wong WS (2009) Control communication complexity of distributed control systems. *SIAM J Optim Control* 48(3):1722–1742
- Wong WS, Baillieul J (2009) Control communication complexity of nonlinear systems. *Commun Inf Syst* 9(1):103–140
- Wong WS, Baillieul J (2012) Control communication complexity of distributed actions. *IEEE Trans Autom Control* 57(11):2731–2745. <https://doi.org/10.1109/TAC.2012.2192357>
- Yu J, Wang L (2010) Group consensus in multi-agent systems with switching topologies and communication delays. *Syst Control Lett* 59(6):340–348
- Yüksel S, Başar T (2007) Optimal signaling policies for decentralized multicontroller stabilizability over communication channels. *IEEE Trans Autom Control* 52(10):1969–1974

Information-Theoretic Approaches for Non-classical Information Patterns

Pulkit Grover

Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

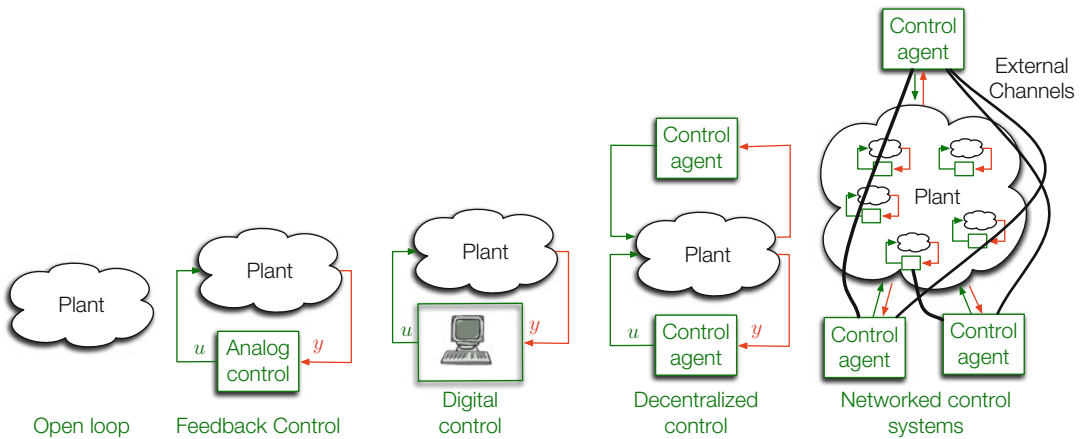
The concept of “information structures” in decentralized control is a formalization of the notion of “who knows what and when do they know it.” Even seemingly simple problems with simply stated information structures can be extremely hard to solve. Perhaps the simplest of such unsolved problem is the celebrated Witsenhausen counterexample, formulated by Hans Witsenhausen in 1968. This article discusses how the information structure of the Witsenhausen counterexample makes it hard and how an information-theoretic approach, which explores the knowledge gradient due to a non-classical information pattern, has helped obtain insights into the problem. Applications of these tools to modern cyber-physical systems are also discussed.

Keywords

Team decision theory · Decentralized control · Information theory · Implicit communication

Introduction

Modern control systems often comprise of multiple decentralized control agents that interact over communication channels. What characteristic distinguishes a centralized control problem from a decentralized one? One fundamental difference is a “knowledge gradient”: agents in a decentralized team often observe, and hence know, different things. This observation leads to the idea of *information patterns* (Witsenhausen 1971), a formalization of the notion of “who



Information-Theoretic Approaches for Non-classical Information Patterns, Fig. 1 The evolution of control systems. Modern “networked control systems” (also

called “cyber-physical systems”) are decentralized and networked using communication channels

knows what and when do they know it” (Ho et al. 1978; Mitter and Sahai 1999).

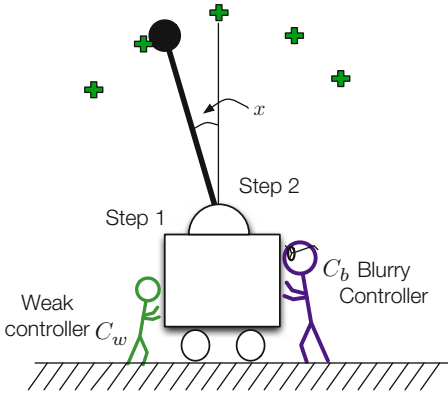
The information pattern is said to be *classical* if all agents in the team receive the same information and have perfect recall (so they do not forget it). What is so special about classical information patterns? For these patterns, the presence of external communication links has no effect on the optimal costs! After all, what could the agents use the communication links for, when there is no knowledge gradient? More interesting, therefore, are the problems for which the information pattern is *non-classical* (Fig. 1). These problems sit at the intersection of communication and control: *communication between agents* can help reduce the knowledge differential that exists between them, helping them perform the *control* task. Intellectually and practically, the concept of non-classical information patterns motivates a lot of formulations at the control-communication intersection. Many of these formulations – including some discussed in this encyclopedia (e.g., Nair 2014; Kawan 2014; Bushnell 2014; Hespanha 2014; Yuksel 2014) – intellectually ask the question: for a realistic channel that is constrained by noise, bandwidth, and speed, what is the optimal communication *and* control strategy?

One could ask the question of optimal control strategy even for decentralized control problems where no external channel is available to bridge

this knowledge gradient. Why could these problems be of interest? First, these problems are limiting cases of control with communication constraints. Second, and perhaps more importantly, they bring out an interesting possibility that can allow the agents to “communicate,” i.e., exchange information, *even when the external channel is absent*. It is possible to use control actions to *communicate* through changing the system state! We now introduce this form of communication using a simple toy example.

Communicating Using Actions: An Example

To gain intuition into when communication using actions could be useful, consider the inverted pendulum example shown in Fig. 2. The goal of the two agents is to bring the pendulum as close to the origin as possible. Both controllers have their strengths and weaknesses. The “weak” controller C_w has little energy but has perfect state observations. On the other hand, the “blurry” controller C_b has infinite energy but noisy observations. They act one after the other, and their goal is to move the pendulum close to the center from any initial state. The information structure of the problem is non-classical: the C_w , but not C_b , knows the initial state of the pendulum, and C_w



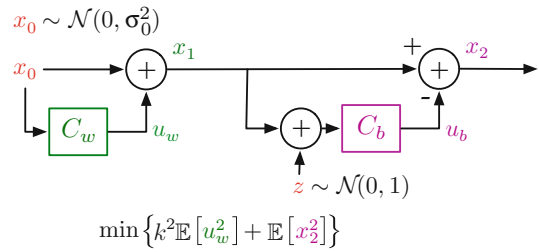
Information-Theoretic Approaches for Non-classical Information Patterns, Fig. 2 Two controllers, with their respective strengths and weaknesses, attempting to bring an inverted pendulum close to the center. Also shown (using green “+” signs) are possible quantization points chosen by the controllers for a quantization-based control strategy

does not know the precise (noisy) observation of C_b using which C_b takes actions.

A possible strategy: A little thought reveals an interesting strategy – the weak controller, having perfect observations, can move the state to the closest of some pre-decided points in space, effectively *quantizing* the state. If these quantization points are sufficiently far from each other, they can be estimated accurately (through the noise) by the blurry controller, which can then use its energy to push the pendulum all the way to zero. In this way, the weak controller expends little energy but is able to “communicate” the state through the noise to the blurry controller, by making it take values on a finite set. Once the blurry controller has received the state through the noise, it can use its infinite energy to push the state to zero.

The Witsenhausen Counterexample

The above two-controller inverted pendulum example is, in fact, motivated by what is now known as “the Witsenhausen counterexample,” formulated by Hans Witsenhausen in 1968 (Witsenhausen 1968) (see Fig. 3). In the counterexample, two controllers (denoted

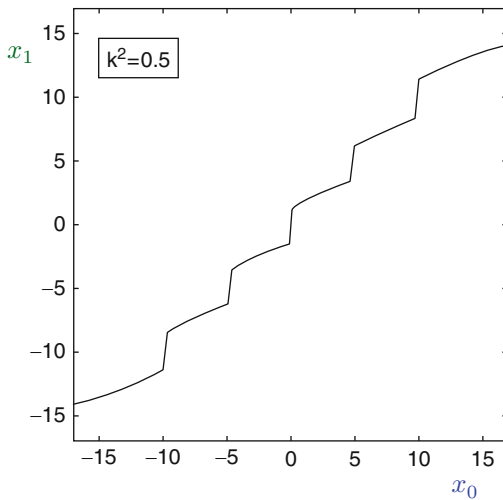


Information-Theoretic Approaches for Non-classical Information Patterns, Fig. 3 The Witsenhausen counterexample is a deceptively simple two-time-step two-controller decentralized control problem. The weak and the blurry controllers, C_w and C_b , act in a sequential manner

here by C_w for “weak” and C_b for “blurry”) act one after the other in two time-steps to minimize a quadratic cost function. The system state is denoted by x_t , where t is the time index. u_w and u_b denote the inputs generated by the two controllers. The cost function is $k^2 \mathbb{E}[u_w^2] + \mathbb{E}[x_2^2]$ for some constant k . The initial state x_0 and the noise z at the input of the blurry controller are assumed to be Gaussian distributed and independent, with variances σ_0^2 and 1, respectively. The problem is a “linear-quadratic-Gaussian” (LQG) problem, i.e., the state evolution is linear, the costs are quadratic, and the primitive random variables are Gaussian.

Why is the problem called a “counterexample”? The traditional “certainty-equivalence” principle (Bertsekas 1995) shows that for all centralized LQG problems, linear control laws are optimal. Witsenhausen (1968) provided a nonlinear strategy for the Witsenhausen problem which outperforms all linear strategies. Thus, the counterexample showed that the certainty-equivalence doctrine does not extend to decentralized control.

What is this strategy of Witsenhausen that outperforms all linear strategies? It is, in fact, a quantization-based strategy, as suggested in our inverted pendulum story above. Further, it was shown by Mitter and Sahai (1999) that multi-point quantization strategies can outperform linear strategies by an arbitrarily large factor. Considering the simplicity of the counterexample, this observation makes the problem very



Information-Theoretic Approaches for Non-classical Information Patterns, Fig. 4 The optimization solution of Baglietto et al. (1997) for $k^2 = 0.5$, $\sigma_0^2 = 5$. The information-theoretic strategy of “dirty-paper coding” (Costa 1983) also yields the same curve (Grover and Sahai 2010a)

important in decentralized control. This simple two-time-step two-controller LQG problem needs to be understood to have any hope of understanding larger and more complex problems.

While the optimal costs for the problem are still unknown, (Even though it is known that an optimal strategy exists (Witsenhausen 1968; Wu and Verdú 2011)) there exists a wealth of understanding of the counterexample that has helped address more complicated problems. A body of work, starting with that of Baglietto et al. (1997), numerically obtained solutions that could be close to optimal (although there is no mathematical proof thereof; indeed, there are many local minima that make this search difficult Subramanian et al. 2018). All these solutions have a consistent form (illustrated in Fig. 4), with slight improvements in the optimal cost. Because the discrete version of the problem, appropriately relaxed, is known to be NP-complete (Papadimitriou and Tsitsiklis 1986), this approach cannot be used to understand the entire parameter space, and hence has focused on one point: $k^2 = 0.5$, $\sigma_0^2 = 5$. Nevertheless, the approach has proven to be insightful: a recent

information-theoretic body of work shows that the strategies of Fig. 4 can be thought of as information-theoretic strategies of “dirty-paper coding” (Costa 1983) that is related to the idea of embedding information in the state. The question here is: how do we embed the information about the state *in the state itself*?

An information-theoretic view of the counterexample This information-theoretic approach that culminated in Grover et al. (2013) also obtained the first approximately optimal solutions to the Witsenhausen counterexample, as well as its vector extensions. The result is established by analyzing information flows in the counterexample that work toward minimizing the knowledge gradient, effectively an information pattern in which C_w can predict the observation of C_b more precisely. The analysis provides *an information-theoretic lower bound on cost that holds irrespective of what strategy is used*. For the original problem, this characterizes the optimal costs (with associated strategies) within a factor of 8 for all problem parameters (i.e., k and σ_0^2). For any finite-length extension, uniform-finite-ratio approximations also exist (Grover et al. 2013). The asymptotically infinite-length extension has been solved *exactly* (Choudhuri and Mitra 2012).

The information-theoretic strategies build on the idea of “dirty-paper coding” (DPC) (Costa 1983) in information theory and utilize a combination of DPC and linear strategies. These strategies, termed “slopy quantization”, were recently shown to be locally optimal (Ajorlou and Jadbabaie 2016) and resemble the strategies obtained numerically in Baglietto et al. (1997).

The problem has also driven delineation of decentralized LQG control problems with optimal linear solutions and those with nonlinear optimal solutions. This led to development and understanding of many variations of the counterexample (Bansal and Başar 1987; Ho et al. 1978; Rotkowitz et al. 2006; Başar 2008) and understanding that can extend to larger decentralized control problems. More recent work shows that the promise of the Witsenhausen counterexample was not a misplaced one: the

information-theoretic approach that provides approximately optimal solutions to the counterexample (Grover et al. 2013) yields solutions to other more complex (e.g., multi-controller, more time-steps) problems as well (Park and Sahai 2012; Grover 2010).

Applications, Summary, and Future Directions

One might ask if this two-agent problem already provides insights on practical applications. As the era of cyber-physical systems arrives, we believe that the applications of decentralized control will only increase. One particular issue of interest is synchronization, which, abstractly, requires the state of two systems to be aligned in state space and time. This requires accurate estimates of states of each agents to be available to other agents – if the agents have noisy or blurry observations, limited capacity channels connecting them, or a limited resource clock synchronizing them. For example, in Shi et al. (2016), the authors observe that hardware implementation of certain neuromorphic architectures suffers from phase-synchronization issues. They arrive at quantization-based strategies, in which the phase itself is quantized, to attain reliable estimation. Similarly, in a robotics company called “Kiva Systems”, locations of robots are quantized to enable multiple robots to work in the same warehouse without collisions (). In general, wherever extremely precise estimates of states are required, and state editing is possible, quantization of the state might be a useful approach but can be easily overlooked. It might also be possible to improve it using dirty-paper coding-type strategies developed in the literature. When communication channels are present, one can optimize use of both implicit and explicit channels (Grover and Sahai 2010b).

This information-theoretic approach is promising for larger decentralized control problems as well. It is now important to explore what is the simplest decentralized control problem that cannot be solved (exactly or approximately) using ideas developed for the

counterexample. In this manner, the Witsenhausen counterexample can provide a platform to unify the more modern (i.e., external-channel centric approaches; see Nair (2014), Kawan (2014), Bushnell (2014), Hespanha (2014), and Yuksel (2014) in the encyclopedia) with the more classical decentralized LQG problems, leading to intellectually enriching and practically useful formulations.

Cross-References

- [Data Rate of Nonlinear Control Systems and Feedback Entropy](#)
- [Information and Communication Complexity of Networked Control Systems](#)
- [Networked Control Systems: Architecture and Stability Issues](#)
- [Networked Control Systems: Estimation and Control Over Lossy Networks](#)
- [Quantized Control and Data Rate Constraints](#)

Bibliography

- Ajorlou A, Jadbabaie A (2016) Slope quantizers are locally optimal for witsenhausen’s counterexample. In: 2016 IEEE 55th conference on decision and control (CDC). IEEE, pp 3566–3571
- Baglietto M, Parisini T, Zoppoli R (1997) Nonlinear approximations for the solution of team optimal control problems. In: Proceedings of the IEEE conference on decision and control (CDC), pp 4592–4594
- Bansal R, Başar T (1987) Stochastic teams with nonclassical information revisited: when is an affine control optimal? IEEE Trans Autom Control 32:554–559
- Başar T (2008) Variations on the theme of the Witsenhausen counterexample. In: Proceedings of the 47th IEEE conference on decision and control (CDC), pp 1614–1619
- Bertsekas D (1995) Dynamic programming. Athena Scientific, Belmont
- Bushnell L (2014) Networked control systems: architecture and stability issues. In: Encyclopedia of systems and control
- Choudhuri C, Mitra U (2012) On Witsenhausen’s counterexample: the asymptotic vector case. In: IEEE information theory workshop (ITW), pp 162–166
- Costa M (1983) Writing on dirty paper. IEEE Trans Inf Theory 29(3):439–441
- Grover P (2010) Actions can speak more clearly than words. Ph.D. dissertation, UC Berkeley, Berkeley

- Grover P, Sahai A (2010a) Vector Witsenhausen counterexample as assisted interference suppression. In: Special issue on information processing and decision making in distributed control systems of the international journal on systems, control and communications (IJSCC), vol 2, pp 197–237
- Grover P, Sahai A (2010b) Implicit and explicit communication in decentralized control. In: Proceedings of the allerton conference on communication, control, and computing, Monticello, Oct 2010. [Online]. Available: <http://arxiv.org/abs/1010.4854>
- Grover P, Park S-Y, Sahai A (2013) Approximately-optimal solutions to the finite-dimensional Witsenhausen counterexample. *IEEE Trans Autom Control* 58(9):2189–2204
- Hespanha J (2014) Networked control systems: estimation and control over lossy networks. In: *Encyclopedia of systems and control*
- Ho YC, Kastner MP, Wong E (1978) Teams, signaling, and information theory. *IEEE Trans Autom Control* 23(2):305–312
- Kawan C (2014) Data-rate of nonlinear control systems and feedback entropy. In: *Encyclopedia of systems and control*
- Kiva Systems (acquired by amazon robotics). [Online]. Available: <https://www.amazonrobotics.com/>
- Mitter SK, Sahai A (1999) Information and control: Witsenhausen revisited. In: Yamamoto Y, Hara S (eds) *Learning, control and hybrid systems: lecture notes in control and information sciences*, vol 241. Springer, New York, pp 281–293
- Nair G (2014) Quantized control and data-rate constraints. In: *Encyclopedia of systems and control*
- Papadimitriou CH, Tsitsiklis JN (1986) Intractable problems in control theory. *SIAM J Control Optim* 24(4):639–654
- Park SY, Sahai A (2012) It may be “easier to approximate” decentralized infinite-horizon LQG problems. In: 51st IEEE conference on decision and control (CDC), pp 2250–2255
- Rotkowitz M (2006) Linear controllers are uniformly optimal for the Witsenhausen counterexample. In: Proceedings of the 45th IEEE conference on decision and control (CDC), pp 553–558, Dec 2006
- Shi R, Jackson TC, Swenson B, Kar S, Pileggi L (2016) On the design of phase locked loop oscillatory neural networks: mitigation of transmission delay effects. In: 2016 international joint conference on neural networks (IJCNN), pp 2039–2046
- Subramanian V, Brink L, Jain N, Vodrahalli K, Jalan A, Shinde N, Sahai A (2018) Some new numeric results concerning the witsenhausen counterexample. In: 2018 56th annual allerton conference on communication, control, and computing (Allerton). IEEE, pp 413–420
- Witsenhausen HS (1968) A counterexample in stochastic optimum control. *SIAM J Control* 6(1):131–147
- Witsenhausen HS (1971) Separation of estimation and control for discrete time systems. *Proc IEEE* 59(11):1557–1566

Wu Y, Verdú S (2011) Witsenhausen’s counterexample: a view from optimal transport theory. In: *IEEE conference on decision and control and European control conference (CDC-ECC)*, pp 5732–5737

Yuksel S (2014) Information and communication complexity of networked control systems. In: *Encyclopedia of systems and control*

Input-to-State Stability

Eduardo D. Sontag

Departments of Bioengineering and Electrical and Computer Engineering, Northeastern University, Boston, MA, USA

Abstract

The notion of input-to-state stability (ISS) qualitatively describes stability of the mapping from initial states and inputs to internal states (and more generally outputs). The entry focuses on the definition of ISS and a discussion of equivalent characterizations.

Keywords

Input-to-state stability · Lyapunov functions · Dissipation · Asymptotic stability

Introduction

We consider here systems with inputs in the usual sense of control theory:

$$\dot{x}(t) = f(x(t), u(t))$$

(the arguments “ t ” is often omitted). There are n state variables and m input channels. States $x(t)$ take values in Euclidean space $\mathbb{R}^n$, and the inputs (also called “controls” or “disturbances” depending on the context) are measurable locally essentially bounded maps $u(\cdot) : [0, \infty) \rightarrow \mathbb{R}^m$. The map $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is assumed to be locally Lipschitz with $f(0, 0) = 0$. The solution, defined on some maximal interval

$[0, t_{\max}(x^0, u))$, for each initial state x^0 and input u , is denoted as $x(t, x^0, u)$, and in particular, for systems with no inputs $\dot{x}(t) = f(x(t))$, just as $x(t, x^0)$. The *zero-system* associated with $\dot{x} = f(x, u)$ is by definition the system with no inputs $\dot{x} = f(x, 0)$. Euclidean norm is written as $|x|$. For a function of time, typically an input or a state trajectory, $\|u\|$, or $\|u\|_\infty$ for emphasis, is the (essential) supremum or “sup” norm (possibly $+\infty$, if u is not bounded). The norm of the restriction of a signal to an interval I is denoted by $\|u_I\|_\infty$ (or just $\|u_I\|$).

Input-to-State Stability

It is convenient to introduce “comparison functions” to quantify stability. A *class $\mathcal{K}_\infty$ function* is a function $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ which is continuous, strictly increasing, and unbounded and satisfies $\alpha(0) = 0$, and a *class $\mathcal{KL}$ function* is a function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ such that $\beta(\cdot, t) \in \mathcal{K}_\infty$ for each t and $\beta(r, t)$ decreases to zero as $t \rightarrow \infty$, for each fixed r .

For a system with no inputs $\dot{x} = f(x)$, there is a well-known notion of global asymptotic stability (for short from now on, *GAS* or “*0-GAS*” when referring to the zero-system $\dot{x} = f(x, 0)$ associated with a given system with inputs $\dot{x} = f(x, u)$) due to Lyapunov, and usually defined in “ ε - δ ” terms. It is an easy exercise to show that this standard definition is in fact equivalent to the following statement:

$$(\exists \beta \in \mathcal{KL}) \quad |x(t, x^0)| \leq \beta(|x^0|, t) \quad \forall x^0, \forall t \geq 0.$$

The notion of input-to-state stability (ISS) was introduced in Sontag (1989), and it provides theoretical concepts used to describe stability features of a mapping $(u(\cdot), x(0)) \mapsto x(\cdot)$ that sends initial states and input functions into states (or, more generally, outputs). Prominent among these features are that inputs that are bounded, “eventually small,” “integrally small,” or convergent, should lead to outputs with the respective property. In addition, ISS and related notions quantify in what

manner initial states affect transient behavior. The formal definition is as follows.

A system is said to be *input to state stable* (ISS) if there exist some $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$ such that

$$|x(t)| \leq \beta(|x^0|, t) + \gamma(\|u\|_\infty) \quad (\text{ISS})$$

holds for all solutions (meaning that the estimate is valid for all inputs $u(\cdot)$, all initial conditions x^0 , and all $t \geq 0$). Note that the supremum $\sup_{s \in [0, t]} \gamma(|u(s)|)$ over the interval $[0, t]$ is the same as $\gamma(\|u_{[0, t]}\|_\infty) = \gamma(\sup_{s \in [0, t]} |u(s)|)$, because the function γ is increasing, so one may replace this term by $\gamma(\|u\|_\infty)$, where $\|u\|_\infty = \sup_{s \in [0, \infty)} \gamma(|u(s)|)$ is the sup norm of the input, because the solution $x(t)$ depends only on values $u(s)$, $s \leq t$ (so, one could equally well consider the input that has values $\equiv 0$ for all $s > t$).

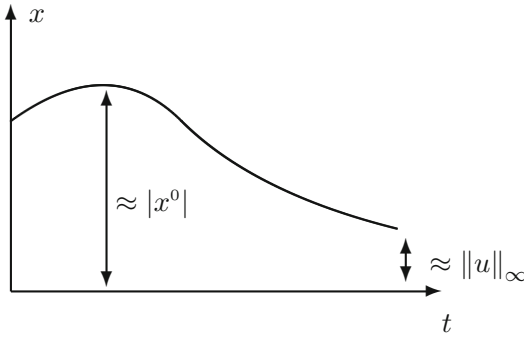
Since, in general, $\max\{a, b\} \leq a + b \leq \max\{2a, 2b\}$, one can restate the ISS condition in a slightly different manner, namely, asking for the existence of some $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$ (in general different from the ones in the ISS definition) such that

$$|x(t)| \leq \max\{\beta(|x^0|, t), \gamma(\|u\|_\infty)\}$$

holds for all solutions. Such redefinitions, using “max” instead of sum, are also possible for each of the other concepts to be introduced later.

Intuitively, the definition of ISS requires that, for t large, the size of the state must be bounded by some function of the sup norm – that is to say, the amplitude – of inputs (because $\beta(|x^0|, t) \rightarrow 0$ as $t \rightarrow \infty$). On the other hand, the $\beta(|x^0|, 0)$ term may dominate for small t , and this serves to quantify the magnitude of the transient (overshoot) behavior as a function of the size of the initial state x^0 (Fig. 1). The *ISS superposition theorem*, discussed later, shows that ISS is, in a precise mathematical sense, the conjunction of two properties, one of them dealing with asymptotic bounds on $|x^0|$ as a function of the magnitude of the input and the other one providing a transient term obtained when one ignores inputs.

For internally stable linear systems $\dot{x} = Ax + Bu$, the variation of parameters formula



Input-to-State Stability, Fig. 1 ISS combines overshoot and asymptotic behavior

gives immediately the following inequality:

$$|x(t)| \leq \beta(t) |x^0| + \gamma \|u\|_\infty,$$

where

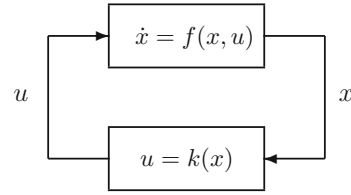
$$\beta(t) = \|e^{tA}\| \rightarrow 0 \text{ and}$$

$$\gamma = \|B\| \int_0^\infty \|e^{sA}\| ds < \infty.$$

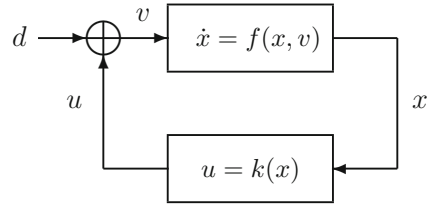
This is a particular case of the ISS estimate, $|x(t)| \leq \beta(|x^0|, t) + \gamma (\|u\|_\infty)$, with linear comparison functions.

Feedback Redesign

The notion of ISS arose originally as a way to precisely formulate, and then answer, the following question. Suppose that, as in many problems in control theory, a system $\dot{x} = f(x, u)$ has been stabilized by means of a feedback law $u = k(x)$ (Fig. 2), that is to say, k was chosen such that the origin of the closed-loop system $\dot{x} = f(x, k(x))$ is globally asymptotically stable. (See, e.g., Sontag 1999 for a discussion of mathematical aspects of state feedback stabilization.) Typically, the design of k was performed by ignoring the effect of possible *input disturbances* $d(\cdot)$ (also called *actuator disturbances*). These “disturbances” might represent true noise or perhaps errors in the calculation of the value $k(x)$



Input-to-State Stability, Fig. 2 Feedback stabilization, closed-loop system $\dot{x} = f(x, k(x))$



Input-to-State Stability, Fig. 3 Actuator disturbances, closed-loop system $\dot{x} = f(x, k(x) + d)$

by a physical controller or modeling uncertainty in the controller or the system itself. What is the effect of considering disturbances? In order to analyze the problem, d is incorporated into the model, and one studies the new system $\dot{x} = f(x, k(x) + d)$, where d is seen as an input (Fig. 3). One may then ask what is the effect of d on the behavior of the system. Disturbances d may well destabilize the system, and the problem may arise even when using a routine technique for control design, feedback linearization. To appreciate this issue, take the following very simple example. Given is the system

$$\dot{x} = f(x, u) = x + (x^2 + 1)u.$$

In order to stabilize it, substitute $u = \frac{\tilde{u}}{x^2 + 1}$ (a preliminary feedback transformation), rendering the system linear with respect to the new input $\tilde{u}$: $\dot{x} = x + \tilde{u}$, and then use $\tilde{u} = -2x$ in order to obtain the closed-loop system $\dot{x} = -x$. In other words, in terms of the original input u , the feedback law is

$$k(x) = \frac{-2x}{x^2 + 1}$$

so that $f(x, k(x)) = -x$. This is a GAS system. The effect of the disturbance input d is analyzed

as follows. The system $\dot{x} = f(x, k(x) + d)$ is:

$$\dot{x} = -x + (x^2 + 1)d.$$

This system has solutions which diverge to infinity even for inputs d that converge to zero; moreover, the constant input $d \equiv 1$ results in solutions that explode in finite time. Thus $k(x) = \frac{-2x}{x^2+1}$ was not a good feedback law, in the sense that its performance degraded drastically once actuator disturbances were taken into account.

The key observation for what follows is that if one adds a correction term “ $-x$ ” to the above formula for $k(x)$, so that now:

$$\tilde{k}(x) = \frac{-2x}{x^2 + 1} - x$$

then the system $\dot{x} = f(x, \tilde{k}(x) + d)$ with disturbance d as input becomes, instead:

$$\dot{x} = -2x - x^3 + (x^2 + 1)d$$

and this system is much better behaved: it is still GAS when there are no disturbances (it reduces to $\dot{x} = -2x - x^3$) but, in addition, it is ISS (easy to verify directly, or appealing to some of the characterizations mentioned later). Intuitively, for large x , the term $-x^3$ serves to dominate the term $(x^2 + 1)d$, for all bounded disturbances $d(\cdot)$, and this prevents the state from getting too large.

This example is an instance of a general result, which says that, whenever there is some feedback law that stabilizes a system, there is also a (possibly different) feedback so that the system with external input d is ISS.

Theorem 1 (Sontag 1989) *Consider a system affine in controls*

$$\begin{aligned} \dot{x} &= f(x, u) \\ &= g_0(x) + \sum_{i=1}^m u_i g_i(x) \quad (g_0(0) = 0) \end{aligned}$$

and suppose that there is some differentiable feedback law $u = k(x)$ so that

$$\dot{x} = f(x, k(x))$$

has $x = 0$ as a GAS equilibrium. Then, there is a feedback law $u = \tilde{k}(x)$ such that

$$\dot{x} = f(x, \tilde{k}(x) + d)$$

is ISS with input $d(\cdot)$

The reader is referred to the book Krstić et al. (1995), and the references given later, for many further developments on the subjects of recursive feedback design, the “back-stepping” approach, and other far-reaching extensions.

Equivalences for ISS

This section reviews results that show that ISS is equivalent to several other notions, including asymptotic gain, existence of robustness margins, dissipativity, and an energy-like stability estimate.

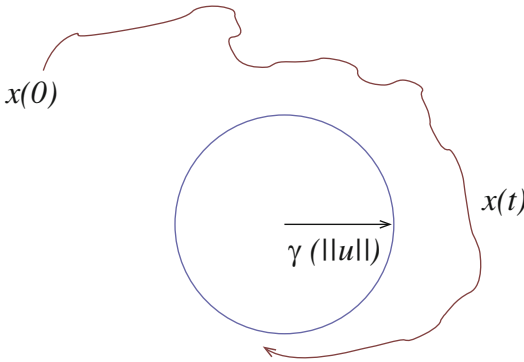
Nonlinear Superposition Principle

Clearly, if a system is ISS, then the system with no inputs $\dot{x} = f(x, 0)$ is GAS: the term $\|u\|_\infty$ vanishes, leaving precisely the GAS property. In particular, then, the system $\dot{x} = f(x, u)$ is *0-stable*, meaning that the origin of the system without inputs $\dot{x} = f(x, 0)$ is stable in the sense of Lyapunov: for each $\varepsilon > 0$, there is some $\delta > 0$ such that $|x^0| < \delta$ implies $|x(t, x^0)| < \varepsilon$. (In comparison-function language, one can restate 0-stability as there is some $\gamma \in \mathcal{K}$ such that $|x(t, x^0)| \leq \gamma(|x^0|)$ holds for all small x^0 .)

On the other hand, since $\beta(|x^0|, t) \rightarrow 0$ as $t \rightarrow \infty$, for t large one has that the first term in the ISS estimate $|x(t)| \leq \max\{\beta(|x^0|, t), \gamma(\|u\|_\infty)\}$ vanishes. Thus an ISS system satisfies the following *asymptotic gain property* (“AG”): there is some $\gamma \in \mathcal{K}_\infty$ so that:

$$\overline{\lim}_{t \rightarrow +\infty} |x(t, x^0, u)| \leq \gamma(\|u\|_\infty) \quad \forall x^0, u(\cdot) \quad (\text{AG})$$

(see Fig. 4). In words, for all large enough t , the trajectory exists, and it gets arbitrarily close to a sphere whose radius is proportional, in a



Input-to-State Stability, Fig. 4 Asymptotic gain property

possibly nonlinear way quantified by the function γ , to the amplitude of the input. In the language of robust control, the estimate (AG) would be called an “ultimate boundedness” condition; it is a generalization of attractivity (all trajectories converge to zero, for a system $\dot{x} = f(x)$ with no inputs) to the case of systems with inputs; the “limsup” is required since the limit of $x(t)$ as $t \rightarrow \infty$ may well not exist. From now on (and analogously when defining other properties), we will just say “the system is AG” instead of the more cumbersome “satisfies the AG property”.

Observe that, since only large values of t matter in the limsup, one can equally well consider merely tails of the input u when computing its sup norm. In other words, one may replace $\gamma(\|u\|_\infty)$ by $\gamma(\lim_{t \rightarrow +\infty} |u(t)|)$, or (since γ is increasing), $\lim_{t \rightarrow +\infty} \gamma(|u(t)|)$.

The surprising fact is that these two necessary conditions are also sufficient. This is summarized by the *ISS superposition theorem*:

Theorem 2 (Sontag and Wang 1996) *A system is ISS if and only if it is 0-stable and AG.*

A minor variation of the above superposition theorem is as follows. Let us consider the *limit property (LIM)*:

$$\inf_{t \geq 0} |x(t, x^0, u)| \leq \gamma(\|u\|_\infty) \quad \forall x^0, u(\cdot) \quad (\text{LIM})$$

(for some $\gamma \in \mathcal{K}_\infty$).

Theorem 3 (Sontag and Wang 1996) *A system is ISS if and only if it is 0-stable and LIM.*

Robust Stability

In this entry, a system is said to be *robustly stable* if it admits a *margin of stability* ρ , that is, a smooth function $\rho \in \mathcal{K}_\infty$ so the system

$$\dot{x} = g(x, d) := f(x, d\rho(|x|))$$

is GAS uniformly in this sense: for some $\beta \in \mathcal{KL}$,

$$|x(t, x^0, d)| \leq \beta(|x^0|, t)$$

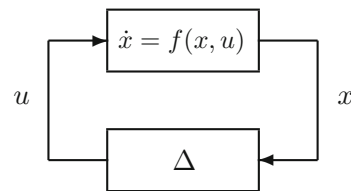
for all possible $d(\cdot) : [0, \infty) \rightarrow [-1, 1]^m$. An alternative way to interpret this concept (cf. Sontag and Wang 1995) is as uniform global asymptotic stability of the origin with respect to all possible time-varying feedback laws Δ bounded by ρ : $|\Delta(t, x)| \leq \rho(|x|)$. In other words, the system

$$\dot{x} = f(x, \Delta(t, x))$$

(Fig. 5) is stable uniformly over all such perturbations Δ . In contrast to the ISS definition, which deals with all possible “open-loop” inputs, the present notion of robust stability asks about all possible closed-loop interconnections. One may think of Δ as representing uncertainty in the dynamics of the original system, for example.

Theorem 4 (Sontag and Wang 1995) *A system is ISS if and only if it is robustly stable.*

Intuitively, the ISS estimate $|x(t)| \leq \max\{\beta(|x^0|, t), \gamma(\|u\|_\infty)\}$ says that the β term dominates as long as $|u(t)| \ll |x(t)|$ for all t , but $|u(t)| \ll |x(t)|$ amounts to $u(t) = d(t) \cdot \rho(|x(t)|)$ with an appropriate function ρ . This is an instance of a “small-gain” argument; see below.



Input-to-State Stability, Fig. 5 Margin of robustness

One analog for linear systems is as follows: if A is a Hurwitz matrix, then $A + Q$ is also Hurwitz, for all small enough perturbations Q ; note that when Q is a nonsingular matrix, $|Qx|$ is a $\mathcal{K}_\infty$ function of $|x|$.

Dissipation

Another characterization of ISS is as a dissipation notion stated in terms of a Lyapunov-like function. A continuous function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be a *storage function* if it is positive definite, that is, $V(0) = 0$ and $V(x) > 0$ for $x \neq 0$, and proper, that is, $V(x) \rightarrow \infty$ as $|x| \rightarrow \infty$. This last property is equivalent to the requirement that the sets $V^{-1}([0, A])$ should be compact subsets of $\mathbb{R}^n$, for each $A > 0$, and in the engineering literature, it is usual to call such functions *radially unbounded*. It is an easy exercise to show that $V : \mathbb{R}^n \rightarrow \mathbb{R}$ is a storage function if and only if there exist $\underline{\alpha}, \bar{\alpha} \in \mathcal{K}_\infty$ such that

$$\underline{\alpha}(|x|) \leq V(x) \leq \bar{\alpha}(|x|) \quad \forall x \in \mathbb{R}^n$$

(the lower bound amounts to properness and $V(x) > 0$ for $x \neq 0$, while the upper bound guarantees $V(0) = 0$). For convenience, $\dot{V} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ is the function:

$$\dot{V}(x, u) := \nabla V(x) \cdot f(x, u)$$

which provides, when evaluated at $(x(t), u(t))$, the derivative $dV(x(t))/dt$ along solutions of $\dot{x} = f(x, u)$.

An *ISS-Lyapunov function* for $\dot{x} = f(x, u)$ is by definition a smooth storage function V for which there exist functions $\gamma, \alpha \in \mathcal{K}_\infty$ so that

$$\dot{V}(x, u) \leq -\alpha(|x|) + \gamma(|u|) \quad \forall x, u. \quad (\text{L-ISS})$$

Integrating, an equivalent statement is that, along all trajectories of the system, there holds the following dissipation inequality:

$$V(x(t_2)) - V(x(t_1)) \leq \int_{t_1}^{t_2} w(u(s), x(s)) ds$$

where, using the terminology of Willems (1976), the “supply” function is $w(u, x) = \gamma(|u|) - \alpha(|x|)$. For systems with no inputs, an ISS-Lyapunov function is precisely the same object as a Lyapunov function in the usual sense.

Theorem 5 (Sontag and Wang 1995) *A system is ISS if and only if it admits a smooth ISS-Lyapunov function.*

Since $-\alpha(|x|) \leq -\alpha(\bar{\alpha}^{-1}(V(x)))$, the ISS-Lyapunov condition can be restated as

$$\dot{V}(x, u) \leq -\tilde{\alpha}(V(x)) + \gamma(|u|) \quad \forall x, u$$

for some $\tilde{\alpha} \in \mathcal{K}_\infty$. In fact, one may strengthen this a bit (Praly and Wang 1996): for any ISS system, there is always a smooth ISS-Lyapunov function satisfying the “exponential” estimate $\dot{V}(x, u) \leq -V(x) + \gamma(|u|)$.

The sufficiency of the ISS-Lyapunov condition is easy to show and was already in the original paper (Sontag 1989). A sketch of proof is as follows, assuming for simplicity a dissipation estimate in the form $\dot{V}(x, u) \leq -\alpha(V(x)) + \gamma(|u|)$. Given any x and u , either $\alpha(V(x)) \leq 2\gamma(|u|)$ or $\dot{V} \leq -\alpha(V)/2$. From here, one deduces by a comparison theorem that, along all solutions,

$$V(x(t)) \leq \max \left\{ \beta(V(x^0), t), \alpha^{-1}(2\gamma(\|u\|_\infty)) \right\},$$

where the $\mathcal{KL}$ function $\beta(s, t)$ is the solution $y(t)$ of the initial value problem

$$\dot{y} = -\frac{1}{2}\alpha(y) + \gamma(u), \quad y(0) = s.$$

Finally, an ISS estimate is obtained from $V(x^0) \leq \bar{\alpha}(x^0)$.

The proof of the converse part of the theorem is based upon first showing that ISS implies robust stability in the sense already discussed and then obtaining a converse Lyapunov theorem for robust stability for the system $\dot{x} = f(x, d\rho(|x|)) = g(x, d)$, which is asymptotically stable uniformly on all Lebesgue-measurable functions $d(\cdot) : \mathbb{R}_{\geq 0} \rightarrow B(0, 1)$. This last theorem was given in Lin et al. (1996) and is basically a theorem on Lyapunov functions for

differential inclusions. The classical result of Massera (1956) for differential equations (with no inputs) becomes a special case.

Using “Energy” Estimates Instead of Amplitudes

In linear control theory, H_∞ theory studies $L^2 \rightarrow L^2$ induced norms, which under coordinate changes leads to the following type of estimate:

$$\int_0^t \alpha(|x(s)|) ds \leq \alpha_0(|x^0|) + \int_0^t \gamma(|u(s)|) ds$$

along all solutions, and for some $\alpha, \alpha_0, \gamma \in \mathcal{K}_\infty$. Just for the statement of the next result, a system is said to *satisfy an integral-integral estimate* if for every initial state x^0 and input u , the solution $x(t, x^0, u)$ is defined for all $t > 0$ and an estimate as above holds. (In contrast to ISS, this definition explicitly demands that $t_{\max} = \infty$.)

Theorem 6 (Sontag 1998) *A system is ISS if and only if it satisfies an integral-integral estimate.*

This theorem is quite easy to prove, in view of previous results. A sketch of proof is as follows. If the system is ISS, then there is an ISS-Lyapunov function satisfying $\dot{V}(x, u) \leq -V(x) + \gamma(|u|)$, so, integrating along any solution:

$$\begin{aligned} \int_0^t V(x(s)) ds &\leq \int_0^t V(x(s)) ds + V(x(t)) \\ &\leq V(x(0)) + \int_0^t \gamma(|u(s)|) ds \end{aligned}$$

and thus an integral-integral estimate holds. Conversely, if such an estimate holds, one can prove that $\dot{x} = f(x, 0)$ is stable and that an asymptotic gain exists.

Integral Input-to-State Stability

A concept of nonlinear stability that is truly distinct from ISS arises when considering a mixed notion which combines the “energy” of the input with the amplitude of the state. A system is said to be *integral input to state stable (iISS)* provided

that there exist $\alpha, \gamma \in \mathcal{K}_\infty$, and $\beta \in \mathcal{KL}$ such that the estimate

$$\begin{aligned} \alpha(|x(t)|) &\leq \beta(|x^0|, t) \\ &+ \int_0^t \gamma(|u(s)|) ds \quad (\text{iISS}) \end{aligned}$$

holds along all solutions. Just as with ISS, one could state this property merely for all times $t \in t_{\max}(x^0, u)$. Since the right-hand side is bounded on each interval $[0, t]$ (because, recall, inputs are by definition assumed to be bounded on each finite interval), it is automatically true that $t_{\max}(x^0, u) = +\infty$ if such an estimate holds along maximal solutions. So forward completeness (solution exists for all $t > 0$) can be assumed with no loss of generality.

One might also consider the following type of “weak integral to integral” mixed estimate:

$$\begin{aligned} \int_0^t \underline{\alpha}(|x(s)|) ds &\leq \kappa(|x^0|) \\ &+ \alpha \left(\int_0^t \gamma(|u(s)|) ds \right) \end{aligned}$$

for appropriate $\mathcal{K}_\infty$ functions (note the additional “ α ”).

Theorem 7 (Angeli et al. 2000) *A system satisfies a weak integral to integral estimate if and only if it is iISS.*

Another interesting variant is found when considering mixed *integral/supremum* estimates:

$$\begin{aligned} \underline{\alpha}(|x(t)|) &\leq \beta(|x^0|, t) \\ &+ \int_0^t \gamma_1(|u(s)|) ds + \gamma_2(\|u\|_\infty) \end{aligned}$$

for suitable $\beta \in \mathcal{KL}$ and $\underline{\alpha}, \gamma_i \in \mathcal{K}_\infty$. One then has:

Theorem 8 (Angeli et al. 2000) *A system satisfies a mixed estimate if and only if it is iISS.*

Dissipation Characterization of iISS

A smooth storage function V is an *iISS-Lyapunov function* for the system $\dot{x} = f(x, u)$ if there are a $\gamma \in \mathcal{K}_\infty$ and an $\alpha : [0, +\infty) \rightarrow [0, +\infty)$

which is merely *positive definite* (i.e., $\alpha(0) = 0$ and $\alpha(r) > 0$ for $r > 0$) such that the inequality:

$$\dot{V}(x, u) \leq -\alpha(|x|) + \gamma(|u|) \quad (\text{L-iISS})$$

holds for all $(x, u) \in \mathbb{R}^n \times \mathbb{R}^m$. To compare, recall that an ISS-Lyapunov function is required to satisfy an estimate of the same form but where α is required to be of class $\mathcal{K}_\infty$; since every $\mathcal{K}_\infty$ function is positive definite, an ISS-Lyapunov function is also an iISS-Lyapunov function.

Theorem 9 (Angeli et al. 2000) *A system is iISS if and only if it admits a smooth iISS-Lyapunov function.*

Since an ISS-Lyapunov function is also an iISS one, ISS implies iISS. However, iISS is a strictly weaker property than ISS, because α may be bounded in the iISS-Lyapunov estimate, which means that V may increase, and the state become unbounded, even under bounded inputs, so long as $\gamma(|u(t)|)$ is larger than the range of α . This is also clear from the iISS definition, since a constant input with $|u(t)| = r$ results in a term in the right-hand side that grows like rt .

An interesting general class of examples is given by *bilinear* systems

$$\dot{x} = \left(A + \sum_{i=1}^m u_i A_i \right) x + Bu$$

for which the matrix A is Hurwitz. Such systems are always iISS (see Sontag 1998), but they are not in general ISS. For instance, in the case when $B = 0$, boundedness of trajectories for all constant inputs already implies that $A + \sum_{i=1}^m u_i A_i$ must have all eigenvalues with nonpositive real part, for all $u \in \mathbb{R}^m$, which is a condition involving the matrices A_i (e.g., $\dot{x} = -x + ux$ is iISS but it is not ISS).

The notion of iISS is useful in situations where an appropriate notion of detectability can be verified using LaSalle-type arguments. There follow two examples of theorems along these lines.

Theorem 10 (Angeli et al. 2000) *A system is iISS if and only if it is 0-GAS and there is a smooth storage function V such that, for some*

$\sigma \in \mathcal{K}_\infty$:

$$\dot{V}(x, u) \leq \sigma(|u|)$$

for all (x, u) .

The sufficiency part of this result follows from the observation that the 0-GAS property by itself already implies the existence of a smooth and positive definite, but not necessarily proper, function V_0 such that $\dot{V}_0 \leq \gamma_0(|u|) - \alpha_0(|x|)$ for all (x, u) , for some $\gamma_0 \in \mathcal{K}_\infty$ and positive definite α_0 (if V_0 were proper, then it would be an iISS-Lyapunov function). Now one uses $V_0 + V$ as an iISS-Lyapunov function (V provides properness).

Theorem 11 (Angeli et al. 2000) *A system is iISS if and only if there exists an output function $y = h(x)$ (continuous and with $h(0) = 0$) which provides zero detectability ($u \equiv 0$ and $y \equiv 0 \Rightarrow x(t) \rightarrow 0$) and dissipativity in the following sense: there exists a storage function V and $\sigma \in \mathcal{K}_\infty$, α positive definite, so that:*

$$\dot{V}(x, u) \leq \sigma(|u|) - \alpha(h(x))$$

holds for all (x, u) .

The paper Angeli et al. (2000) contains several additional characterizations of iISS.

Superposition Principles for iISS

There are also asymptotic gain characterizations for iISS. A system is *bounded energy weakly converging state (BEWCS)* if there exists some $\sigma \in \mathcal{K}_\infty$ so that the following implication holds:

$$\begin{aligned} \int_0^{+\infty} \sigma(|u(s)|) ds < +\infty \\ \Rightarrow \liminf_{t \rightarrow +\infty} |x(t, x^0, u)| = 0 \quad (\text{BEWCS}) \end{aligned}$$

(more precisely, if the integral is finite, then $t_{\max}(x^0, u) = +\infty$, and the liminf is zero). It is *bounded energy frequently bounded state (BEFBS)* if there exists some $\sigma \in \mathcal{K}_\infty$ so that the following implication holds:

$$\begin{aligned} \int_0^{+\infty} \sigma(|u(s)|) ds < +\infty \\ \Rightarrow \liminf_{t \rightarrow +\infty} |x(t, x^0, u)| < +\infty \quad (\text{BEFBS}) \end{aligned}$$

(again, meaning that $t_{\max}(x^0, u) = +\infty$ and the $\liminf$ is finite).

Theorem 12 (Angeli et al. 2004) *The following three properties are equivalent for any given system $\dot{x} = f(x, u)$:*

- *The system is iISS.*
- *The system is BEWCS and zero stable.*
- *The system is BEFBS and zero GAS.*

Summary, Extensions, and Future Directions

This entry focuses on stability notions relative to steady states, but a more general theory is also possible, that allows consideration of more arbitrary attractors, as well as robust and/or adaptive concepts. Much else has been omitted from this entry. Most importantly, one of the key results is the *ISS small-gain theorem* due to Jiang, Teel, and Praly (Jiang et al. 1994), which provides a powerful sufficient condition for the interconnection of ISS systems being itself ISS. The extension of ISS to PDEs, especially to PDEs with inputs entering the boundary conditions, long remained elusive because of the need to consider unbounded input operators. However, ISS theory is now well-established for both parabolic and hyperbolic PDEs (Karafyllis and Krstić 2019), including stability results, obtained via small-gain arguments, for interconnections or parabolic and/or hyperbolic systems, in various physically meaningful norms.

Other topics not treated include, among many others, all notions involving outputs; ISS properties of time-varying (and in particular periodic) systems; ISS for discrete-time systems; questions of sampling, relating ISS properties of continuous and discrete-time systems; ISS with respect to a closed subset K ; stochastic ISS; applications to tracking, vehicle formations (“leader to followers” stability); and averaging of ISS systems. The paper Sontag (2006) may also be consulted for further references, a detailed development of some of these ideas, and citations to the literature for others. In addition, the textbooks Isidori (1999), Krstić et al. (1995), Khalil (1996), Sepul-

chre et al. (1997), Krstić and Deng (1998), Freeman and Kokotović (1996), Isidori et al. (2003) contain many extensions of the theory as well as applications.

Cross-References

► [Input-to-State Stability for PDEs](#)

Bibliography

- Angeli D, Ingalls B, Sontag ED, Wang Y (2004) Separation principles for input-output and integral-input-to-state stability. *SIAM J Control Optim* 43(1):256–276
- Angeli D, Sontag ED, Wang Y (2000) A characterization of integral input-to-state stability. *IEEE Trans Autom Control* 45(6):1082–1097
- Angeli D, Sontag ED, Wang Y (2000) Further equivalences and semiglobal versions of integral input to state stability. *Dyn Control* 10(2):127–149
- Freeman RA, Kokotović PV (1996) Robust nonlinear control design, state-space and Lyapunov techniques. Birkhauser, Boston
- Isidori A (1999) Nonlinear control systems II. Springer, London
- Isidori A, Marconi L, Serrani A (2003) Robust autonomous guidance: an internal model-based approach. Springer, London
- Jiang Z-P, Teel A, Praly L (1994) Small-gain theorem for ISS systems and applications. *Math Control Signals Syst* 7:95–120
- Karafyllis I, Krstić M (2019) Input-to-state stability for PDEs. Springer, London
- Khalil HK (1996) Nonlinear systems, 2nd edn. Prentice-Hall, Upper Saddle River
- Krstić M, Deng H (1998) Stabilization of uncertain nonlinear systems. Springer, London
- Krstić M, Kanellakopoulos I, Kokotović PV (1995) Nonlinear and adaptive control design. John Wiley & Sons, New York
- Lin Y, Sontag ED, Wang Y (1996) A smooth converse Lyapunov theorem for robust stability. *SIAM J Control Optim* 34(1):124–160
- Massera JL (1956) Contributions to stability theory. *Ann Math* 64:182–206
- Praly L, Wang Y (1996) Stabilization in spite of matched unmodelled dynamics and an equivalent definition of input-to-state stability. *Math Control Signals Syst* 9:1–33
- Sepulchre R, Jankovic M, Kokotović PV (1997) Constructive nonlinear control. Springer, New York
- Sontag ED (1989) Smooth stabilization implies coprime factorization. *IEEE Trans Autom Control* 34(4):435–443
- Sontag ED (1998) Comments on integral variants of ISS. *Syst Control Lett* 34(1–2):93–100

- Sontag ED (1999) Stability and stabilization: discontinuities and the effect of disturbances. In: *Nonlinear analysis, differential equations and control* (Montreal, QC, 1998). NATO Science Series C, Mathematical and physical sciences, vol 528. Kluwer Academic Publishers, Dordrecht, pp 551–598
- Sontag ED (2006) Input to state stability: basic concepts and results. In: Nistri P, Stefani G (eds) *Nonlinear and optimal control theory*. Springer, Berlin, pp 163–220
- Sontag ED, Wang Y (1995) On characterizations of the input-to-state stability property. *Syst Control Lett* 24(5):351–359
- Sontag ED, Wang Y (1996) New characterizations of input-to-state stability. *IEEE Trans Autom Control* 41(9):1283–1294
- Willems JC (1976) Mechanisms for the stability and instability in feedback systems. *Proc IEEE* 64:24–35

Input-to-State Stability for PDEs

Iasson Karafyllis¹ and Miroslav Krstic²

¹Department of Mathematics, National Technical University of Athens, Athens, Greece

²Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA, USA

Abstract

This chapter reviews the challenges for the extension of the Input-to-State Stability (ISS) property for systems described by Partial Differential Equations (PDEs). The methodologies that have been used in the literature for the derivation of ISS estimates, are presented. Examples are also provided and possible directions of future research on ISS for PDEs are given.

Keywords

Input-to-State Stability · Partial Differential Equations · Infinite-Dimensional Systems

Introduction

Input-to-State Stability (ISS) is a stability notion that has played a key role in the development

of modern control theory. ISS combines internal and external stability notions and has been heavily used in the finite-dimensional case for the stability analysis of control systems as well as for the construction of feedback stabilizers. The present chapter is devoted to the extension of the ISS property to infinite-dimensional systems and more specifically, to control systems that involve Partial Differential Equations (PDEs).

Problems and Methods for ISS for PDEs

The extension of input-to-state stability (ISS) to systems described by partial differential equations (PDEs) is a challenging topic that has required extensive research efforts by many researchers. The main reason for the difficulty of such an extension is the fact that there are key features which are absent in finite-dimensional systems but play an important role for the study of ISS in PDE systems. Such key features are:

- (1) The nature of the disturbance inputs. There are two cases for PDE systems where a disturbance input may appear: a disturbance may appear in the PDE itself (distributed disturbance) and/or in the boundary conditions (boundary disturbance). The transformation of a boundary disturbance to a distributed disturbance does not solve the problem because in this way time derivatives of the disturbance appear in the PDE. This happens because the boundary input operator is unbounded. This is a fundamental obstacle which cannot be avoided.
- (2) Systems described by PDEs require various mathematical tools. For example, the mathematical theory and techniques for parabolic PDEs are completely different from the mathematical theory and techniques for hyperbolic PDEs. Moreover, there are PDEs which may not fall into one of these two categories (hyperbolic and parabolic).
- (3) When nonlinear PDEs are studied, the place where the nonlinearity appears is very impor-

tant. PDEs with nonlinear differential operators are much more demanding than PDEs with linear differential operators and nonlinear reaction terms (semilinear PDEs).

- (4) A variety of spatial norms may be used for the solution of a PDE. However, the ISS property written in different state norms has a different meaning. For example, the ISS property in certain L^p norm with $p < +\infty$ cannot give pointwise estimates of the solution. On the other hand, the ISS property in the sup spatial norm can indeed give pointwise estimates of the state.
- (5) Different notions of a solution are used in the theory of PDEs. The notion of the solution of a PDE has a tremendous effect on the selection of the state space and the set of allowable disturbance inputs. So, if weak solutions are used for a PDE, then the state space is usually an L^p space with $p < +\infty$, and it makes no sense to talk about the ISS property in the sup norm. However, it should be noticed that the state space does not necessarily determine the (spatial) norm in the ISS property: for example, if the state space is $H^1(0, 1)$, then the ISS property may be written in an $L^p(0, 1)$ norm with $p \leq +\infty$.
- (6) While the main tool of proving the ISS property for finite-dimensional systems is the ISS Lyapunov function, there are important cases for PDEs where ISS Lyapunov functionals are not available (e.g., the heat equation with boundary disturbances and Dirichlet boundary conditions).
- (7) A PDE may or may not include nonlocal terms. Moreover, the nonlocal terms may appear in the boundary conditions of a PDE system. The latter case is particularly important for control theory, because it is exactly the case that appears when boundary feedback control is applied to a PDE system. However, it should be noticed that PDEs with nonlocal terms are not always covered by standard results of the theory of PDEs.

All the above features do not appear in systems described by ordinary differential equations (ODEs), and this has led the research community

to create novel mathematical tools which can be used for the derivation of an ISS estimate for the solution of a PDE. So far research has mostly focused on 1-D linear or semilinear PDEs. The reader can consult the recent book (Karafyllis and Krstic 2019b) for a detailed description. More specifically, for hyperbolic PDEs researchers have used:

- (a) ISS Lyapunov functionals, which usually provide ISS estimates in L^p norms with $p < +\infty$, and
- (b) the strong relation between hyperbolic PDEs and delay equations (see Bekiaris-Liberis and Krstic (2013) and Karafyllis and Krstic 2014, 2017b). This relation is most useful in 1-D hyperbolic PDEs and provides ISS estimates in the sup spatial norm.

For parabolic PDEs researchers have used numerically inspired methods which relate the solution of a PDE to the solution of a finite-dimensional discrete-time system. Such methods were used in Karafyllis and Krstic (2017a, 2019b), where ISS estimates in various spatial state norms were derived (including the sup spatial norm). Researchers have also used:

- (a) ISS Lyapunov functionals,
- (b) Eigenfunction expansions of the solution, which can be used for the L^2 spatial norm as well as for the spatial H^1 norm,
- (c) Semigroup theory, which is particularly useful for linear PDEs, and
- (d) Monotonicity methods, i.e., methods that exploit the monotonicity of a particular PDE system (see Mironchenko et al. 2019).

Examples of Results of ISS for PDEs

As an example, consider the 1-D heat equation

$$\frac{\partial u}{\partial t}(t, x) = p \frac{\partial^2 u}{\partial x^2}(t, x) + f(t, x),$$

for all $(t, x) \in (0, +\infty) \times (0, 1)$ (1)

where $p > 0$ is a constant, subject to the boundary conditions:

$$u(t, 0) - d_0(t) = u(t, 1) = 0 \quad (2)$$

where $d_0 : \mathfrak{R}_+ \rightarrow \mathfrak{R}$ is a given locally bounded boundary disturbance input and $f : \mathfrak{R}_+ \times [0, 1] \rightarrow \mathfrak{R}$ is a locally bounded distributed input. Using the results in Chapter 5 and Chapter 6 of Karafyllis and Krstic (2019a), we have that:

- every solution $u \in C^0(\mathfrak{R}_+; L^2(0, 1))$ with $u \in C^1((0, +\infty) \times [0, 1])$ satisfying $u[t] \in C^2([0, 1])$, (2) for $t > 0$ and (1) satisfies the following ISS estimate for all $t > 0$

$$\begin{aligned} \|u[t]\|_2 &\leq \exp\left(-p\pi^2 t\right) \|u[0]\|_2 \\ &\quad + \frac{1}{\sqrt{3}} \sup_{0 < s < t} (|d_0(s)|) \\ &\quad + \frac{1}{p\pi^2} \sup_{0 < s < t} (\|f[s]\|_2) \end{aligned} \quad (3)$$

- every solution $u \in C^0(\mathfrak{R}_+ \times [0, 1])$ with $u \in C^1((0, +\infty) \times [0, 1])$ satisfying $u[t] \in C^2([0, 1])$ for $t > 0$, $u[0] \in H^2(0, 1)$, (2) for $t \geq 0$ and (1) satisfies the following ISS estimate for all $t \geq 0$

$$\begin{aligned} \|u[t]\|_\infty &\leq \sqrt{2} \max \\ &\quad \times \left(\exp\left(-\frac{p\pi^2}{4} t\right) \|u[0]\|_\infty, \right. \\ &\quad \left. \sup_{0 \leq s \leq t} (|d_0(s)|) \right) \\ &\quad + \frac{4\sqrt{2}}{p\pi^2} \sup_{0 \leq s \leq t} (\|f[s]\|_\infty) \end{aligned} \quad (4)$$

where $\|u[t]\|_2 = \left(\int_0^1 u^2(t, x) dx \right)^{1/2}$ and $\|u[t]\|_\infty = \max \{ |u(t, x)| : x \in [0, 1] \}$. The reader can notice the difference between estimates (3), (4). As remarked above the selection of different spatial (state) norms leads to completely different ISS estimates.

Small-gain results are also useful for the derivation of ISS estimates for various kinds of PDE systems. In the book Karafyllis and Krstic (2019a), small-gain methods were used for the study of ISS in a wide variety of PDE loops (i.e., interconnected PDEs). Moreover, PDE-ODE loops were also studied. It should be noticed that small-gain results rely on accurate estimations of the gain functions (or gain coefficients), and consequently, it is usually the case that small-gain results cannot be used in conjunction with qualitative characterizations of the ISS property.

Summary and Future Directions

The effort for a complete qualitative characterization of ISS for PDEs continues; the reader should consult (Mironchenko 2016) and notice that many characterizations of the ISS property for ODEs are not valid for PDEs. It should be also noted that at this point in time, there has been very little research devoted to the Input-to-Output Stability (IOS) property.

It is expected that ISS theory for PDEs will enable researchers to solve important feedback design problems and observer design problems. ISS was recently used in Karafyllis and Krstic (2019b) for boundary feedback design in 1-D parabolic semilinear PDEs with nonlinear reaction terms that satisfy a linear growth condition. Future research may address feedback stabilization problems for systems of PDEs as well as more complicated mathematical models, such as free-boundary parabolic PDEs. The ISS property for a boundary controlled Stefan problem was recently shown in Koga (2018). Since PDEs are used for the mathematical modeling of many phenomena, it is also expected that ISS theory for PDEs will have important applications to all sciences.

Cross-References

► [Input-to-State Stability](#)

Bibliography

- Bekiaris-Liberis N, Krstic M (2013) Nonlinear control under nonconstant delays. SIAM, Philadelphia
- Karafyllis I, Krstic M (2014) On the relation of delay equations to first-order hyperbolic partial differential equations. *ESAIM Control Optim Calc Var* 20: 894–923
- Karafyllis I, Krstic M (2017a) ISS in different norms for 1-D parabolic PDEs with boundary disturbances. *SIAM J Control Optim* 55:1716–1751
- Karafyllis I, Krstic M (2017b) Stability of integral delay equations and stabilization of age-structured models. *ESAIM Control Optim Calc Var* 23(4):1667–1714
- Karafyllis I, Krstic M (2019a) Input-to-state stability for PDEs. *Communications and control engineering*. Springer, London
- Karafyllis I, Krstic M (2019b) Small-gain-based boundary feedback design for global exponential stabilization of 1-D semilinear parabolic PDEs. *SIAM J Control Optim* 57(3):2016–2036
- Koga S, Karafyllis I, Krstic M (2018) Input-to-state stability for the control of Stefan problem with respect to heat loss at the interface. In: *Proceedings of the 2018 American control conference*
- Mironchenko A (2016) Local input-to-state stability: characterizations and counterexamples. *Syst Control Lett* 87:23–28
- Mironchenko A, Karafyllis I, Krstic M (2019) Monotonicity methods for input-to-state stability of nonlinear parabolic PDEs with boundary disturbances. *SIAM J Control Optim* 57:510–532

Interaction Patterns as Units of Analysis in Hierarchical Human-in-the-Loop Control

Bérénice Mettler

International Computer Science Institute (ICSI),
Berkeley, CA, USA

Department of Aerospace Engineering and
Mechanics, University of Minnesota,
Minneapolis, MN, USA

Abstract

Interaction patterns (IPs) are coordinated, sensory-motor behaviors used by human operators to effectively interact with complex task environments. They take different forms depending on the domain of activity. In spatial control tasks, IPs take the form of guidance

primitives that connect discrete environment features with the continuous control and guidance behavior. In particular they can be used to explain the emergence of subgoals and the partitioning of a spatial control problem's workspace.

IPs are significant also because they explain the functional integration of sensory, perceptual, guidance, and control processes. They also represent a keystone in the formation of hierarchical control strategies, where they provide an interface between the low-level sensory and control functions and the high-level cognitive functions. Finally – and more generally – they provide a critical insight into the acquisition of human operators' higher-level control functions.

Keywords

Meta control · Sensory-motor patterns ·
Motion primitives · Perception

Introduction

Motivation

Machines have been climbing the ladder of human skills, and as a result, human operators are expected to interact with machines, at both a higher level and across a more comprehensive control hierarchy. This has been the trend in process control, and it is now underway in spatial control applications, as is already manifest in various autonomous guidance and robotic applications.

The acquisition of skills for complex tasks depends on the ability to form abstractions, including meta control such as motion primitives (see so-called options in reinforcement learning (Sutton et al. 1998)). In general, these are not just simple control abstractions or concatenations of inputs, but correspond to comprehensive input-output patterns associated with agent-environment dynamics. These sensory-motor patterns are critical for enabling more effective interactions with the task and environment elements.

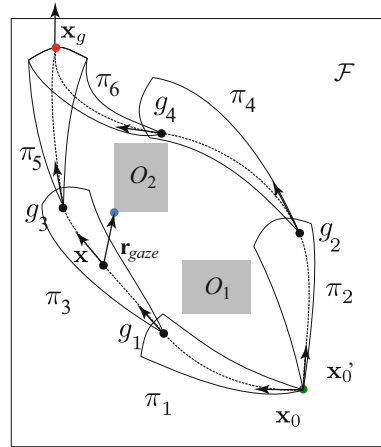
Human control has been primarily investigated at the low-level control functions, such

as tracking and stabilization (see, e.g., McRuer and Krendel 1974). One of the most influential hierarchical control frameworks in human systems modeling and design is Rasmussen's model of human information processing and performance Rasmussen (1983). The framework delineates performance according to skills, rules, and knowledge (signals, signs, and symbols). However, the delineation between levels of processing has not been formally researched in complex tasks which rely on adaptive, agile behaviors. Hierarchical frameworks for such systems are challenging to formalize because they require deeper understanding of the integration between sensing, control, guidance, and planning functions.

Linking the symbolic and signal domains requires an interface that doesn't impose artificial constraints on the system and behavior. Consider the example in Fig. 1, which shows the workspace of a relatively simple navigation problem. A human or artificial agent with dynamics $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ (where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state of the agent and $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control) has to reach the goal x_g from a starting location x_0 while avoiding the obstacles O_1 and O_2 . In such navigation and guidance problems, waypoint-based algorithms provide useful abstractions; however, many such algorithms define waypoints top-down, for example, using geometrical heuristics that don't account for the dynamic interactions with the environment as illustrated in Fig. 1.

As illustrated in Fig. 1, it is possible to define a type of guidance primitive that describes the behavior through subgoals g_i leading to the goal. These types of abstractions have been identified in human guidance experiments (Kong and Mettler 2013) and have been shown to be related to patterns in the human guidance behavior, emerging from the closed-loop dynamics combining the effect of the environment on the dynamic response and, reciprocally, the effect of the dynamics on the agent's environment. Therefore, they have been called interaction patterns (IPs).

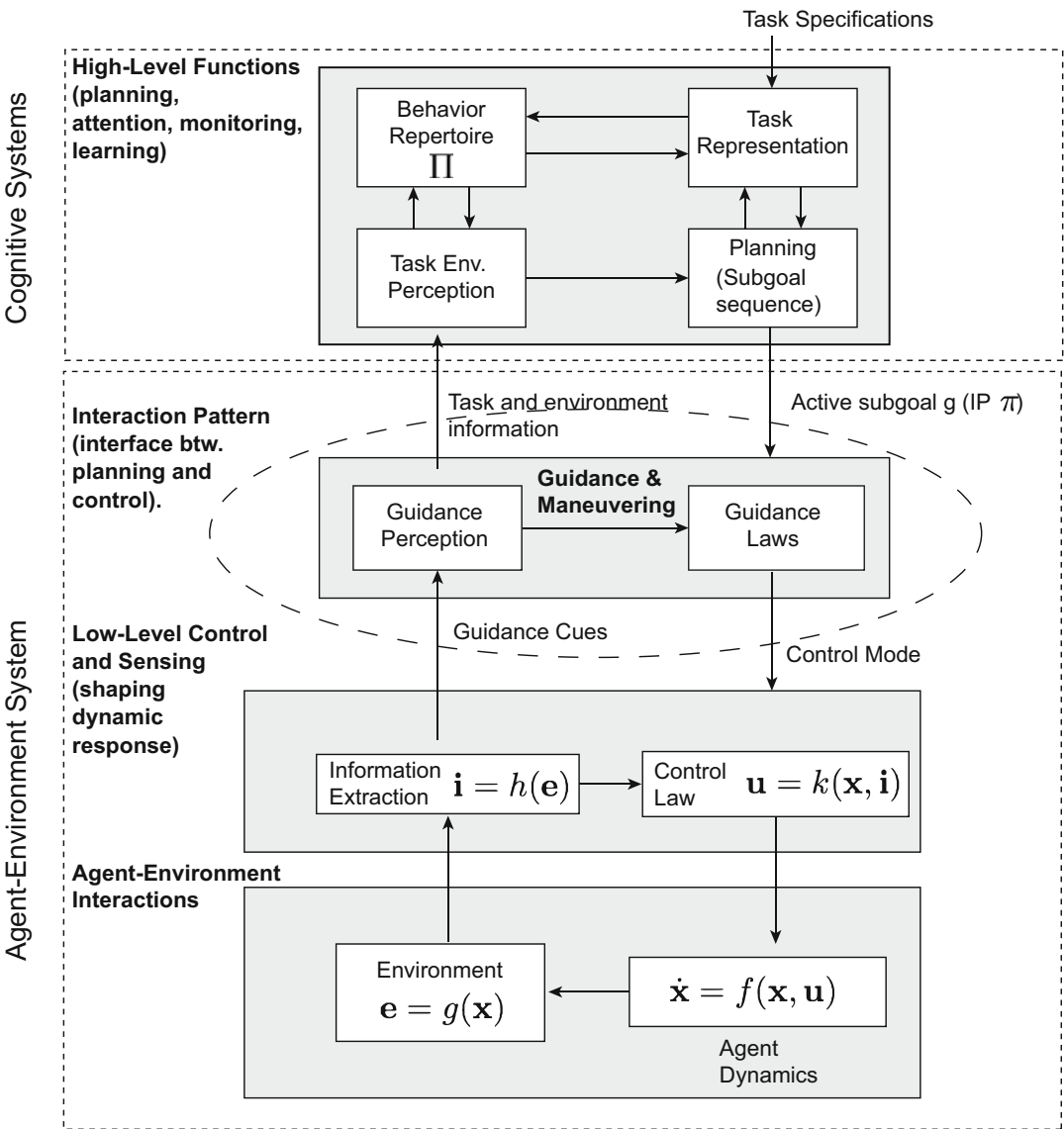
In the most general sense, IPs are patterns in the dynamics of the agent's interactions with the environment. IPs result from several factors,



Interaction Patterns as Units of Analysis in Hierarchical Human-in-the-Loop Control, Fig. 1 Toy navigation problem example highlighting interaction patterns $\pi_1, \pi_2, \dots$ acquired by the operator to effectively navigate through the environment. The figure also shows how these patterns delineate the behavior and define subgoals $g_1, g_2, \dots$ that can be used to abstract the dynamic behavior. The gaze vector r_{gaze} is added to underscore that the behavior results from a perception-action process used to support the interaction with the environment

including guidance skills that build on the natural agent-environment dynamics and, at the same time, planning skills that can exploit the larger patterns. The relationship between the agent dynamics and control and the structure of the larger behavior explains why understanding appropriate units of analysis for complex skills has been elusive. IPs, like patterns in other domains, are interesting because the regularities in the local behavior, as well as the larger organization and structure they confer, can be used to understand unobservable processes and principles related to human skills (Mumford and Desolneux 2010).

For example, IPs illustrate that there are units of organization beyond the basic trajectories and that these have to be considered in other human in the loop control applications. Such units are relevant in many applications because they result in a hierarchical description of behavior. As illustrated in Fig. 2, IPs link the domain of physical implementation, which is dictated by agent dynamics (including the effects of low-level



Interaction Patterns as Units of Analysis in Hierarchical Human-in-the-Loop Control, Fig. 2 Functional hierarchical model based on the IPs and the subgoal

abstraction. Notice the role of the IPs as the interface between low-level and the high-level control, i.e., cognitive control

control and sensing), with the abstract domain of reasoning and planning (Mettler et al. 2015). To summarize, finding control abstractions and what may be considered appropriate units of behavior that are derived from natural properties of the application domain represents one of the central problems for understanding higher-level skills such as learning, knowledge transfer, and

adaptation. These units of behavior also play a critical role in the system-wide organization of processes such as for hierarchical modeling. The following provides general background and summarizes the system-level foundations needed to understand and analyze such patterns. The entry also describes the larger significance for hierarchical modeling of human-in-the-loop control.

Background

Units of Behavior

A strategy to overcome human information processing limitations is chunking, which is the process of collecting multiple pieces of information under a single unit. This concept has primarily been studied in the domain of perception and memory (see, e.g., the early work on perceptual chunking in chess (Chase and Simon 1973)). In engineering and computer sciences, motion primitives (MPs) have been popular abstractions of dynamics. For example, in Frazzoli et al. (2002), MPs are formulated as quantization of the dynamics, i.e., they approximate the trajectories of the nonlinear dynamics through a set of equilibrium trajectories and a set of maneuvers needed to transition between the former. MPs emphasize the control – or output side of the performance – but neglect the input side and larger environment interactions. Motion primitives have also been proposed and studied in neurosciences, in particular motor control (see Flash and Hochner (2005) for a review).

In spatial control domains, chunking implies some form of coordination mechanisms combining sensory stimuli and control, leading to some behavior patterns that are useful for the task performance. From a behavioral standpoint, these patterns are a form of functional unit supporting key interactions with the environment, such as circumventing the obstacles in Fig. 1.

From a systems perspective, units of behavior must satisfy the constraints of the hierarchical system. As depicted in Fig. 2, the behavioral chunks have to provide an interface between low-level and high-level control and higher-level planning (and attention), therefore linking the continuous and discrete elements of behavior delineated in Rasmussen's model.

The concept of IPs thus generalizes that of MPs by incorporating inputs and the environment interaction. And, as can be seen above, IPs can be seen as a chunking approach for spatial control systems that also help elucidate the hierarchical system organization.

Organizational Units and Invariants

A central question is what principles govern the formation of IPs and the larger behavior organization such as shown in Fig. 1. It is expected that agents or human operators can exploit a range of invariants in how they sense and respond to situations and how they coordinate behavior. The following considers two types of invariants: the first are properties of trajectories that remain invariant to transformations, and the other are more general properties of trajectories that are invariant with respect to equivalence relations (see below).

In motion guidance, Tau theory has been validated in many perceptual guidance tasks (Lee 1998). Tau theory stipulates that an agent uses invariants in the perceptual field to sense and regulate the time-to-closure between the desired and current state (motion gap). Behaviors resulting from tau guidance produce perception-action patterns. Patterns from Tau guidance, or similar guidance laws, therefore can be viewed as a form of sensory-motor chunks (Mettler et al. 2015, 2017). However, there has been little research on their role in the macroscopic organization of behavior and larger control hierarchy.

It is well established that human problem-solving in general exploits invariants in the problem domain (Simon 1990). The intuition about abstractions and subgoals such as in Fig. 1 is that, while the specific tasks and their environments may be different, the subtasks are generally transferable between different tasks within the same problem domain. Therefore, these subtasks must satisfy invariant properties. The invariants are expected to be specific to the larger task or problem domain and not the specific problem, which explains why skills can transfer within similar domains. In spatial control, several invariants are available to the agents as described next using a systems formulation.

Agent-Environment Systems Formulation

To characterize the principles that help explain the formation of spatial control abstractions, the

first step is to formulate an agent-environment system (for details see Kong and Mettler 2013 and Mettler et al. 2015).

Dynamic Systems and Agent-Environment Dynamics

In traditional control applications, spatial control is often formulated as a trajectory generation problem. The main problem elements (see Fig. 1) include the vehicle dynamics with state-space domain $\mathcal{X} \subset \mathbb{R}^n$, evolving in a workspace $\mathcal{W} = \mathcal{O}_E \cup \mathcal{F}$, consisting of free space ($\mathcal{F}$) and obstacles ($\mathcal{O}_E = \bigcup_{i=1}^k \mathcal{O}_i$). The closed-loop agent-environment dynamics emphasize the effects of coupling between the agent and the environment mediated by the control sensing functions. They are given as:

$$\dot{\mathbf{x}} = f(\mathbf{x}, k(\mathbf{x}, h(g(\mathbf{x}))))). \quad (1)$$

Please refer to Fig. 2 for the definition of the symbols, and note that letter g is also used to designate subgoals.

Guidance Behavior and Interaction Patterns

The *guidance behavior* is defined as the collection of closed-loop state trajectories, $\mathbf{x}^u(t; \mathbf{x}_0)$, together with the corresponding environment state trajectories, $\mathbf{e}(t) = g(\mathbf{x}(t))$, resulting from the closed-loop dynamics (Eq. 1). The combination of state, input, and environment defines the extended state space $\tilde{\mathcal{X}} = \mathcal{X} \times \mathcal{U} \times \mathcal{E}$, which represents the domain of valid guidance behaviors.

Closed-loop interactions are typically produced by a combination of guidance and planning processes (see Fig. 2). Given additional constraints on their operation, the combination of processes results in guidance behavior forming patterns $\pi_i \in \tilde{\mathcal{X}}$ organized around subgoals g_i such as π_1, π_2, π_3 in Fig. 1.

IPs π_i are the sets of trajectories associated with a guidance and planning process. Referring to these patterns as interaction patterns (IPs) emphasizes that they arise from emergent closed-loop interactions of the agent-environment system (Fig. 2). The patterns can be delineated

into different subsets based on their characteristics, forming a library $\Pi := \{\pi_1, \pi_2, \dots\}$. The general idea, illustrated by Fig. 2, is that the IPs should provide structural features that can be exploited by the high-level perception and planning functions. For example, IPs identified in Kong and Mettler (2013) have provided evidence for operators' exploiting invariants in the closed-loop agent-environment dynamics.

Interaction Patterns and Invariant Properties

Invariants are significant because they represent one way of exploiting structural characteristics in the behavior (Eq. 1). Invariants can be captured through equivalence relations. An equivalence relation $\sim_e$ is a binary relation between elements of a set, in this case, the set of trajectory segments $s \in \Pi$. Two types of equivalence relation were defined in Kong and Mettler (2013) following the properties found in human guidance behavior.

First, guidance laws are generally invariant to rigid-body translation and rotation transformations, which is similar to the invariants underlying MPs, but a key difference here is that the invariant has to be extended to environment trajectories $\mathbf{e}(t)$. This *first equivalence*, described by $\sim_g$ for group action equivalence, derives from the symmetry group M (group action $\Psi : M \times \mathcal{X} \rightarrow \mathcal{X}$).

Second, IPs describe coordinated motions relative to specific environment features, and since IPs break up the task into subtasks, they define subgoals. The trajectories within IPs, e.g., π_1 in Fig. 1, are related through their subgoal g_1 , i.e., once trajectories pass through the subgoal, they share the same trajectory. This relationship has been described by the so-called subgoal equivalence $\sim_s$.

Equivalence Relations and Behavior Organization

Behaviors segments s that satisfy some equivalence relationship $\sim_e$ define an equivalence class:

$$[s]_{\sim_e} = \{s_2 \in \tilde{\mathcal{X}} | s_1 \sim_e s_2\}. \quad (2)$$

Equivalence classes partition the state space $\tilde{\mathcal{X}}$, which explains their role in reducing representational complexity. For example, the IPs in Fig. 1 partition the problem space into distinct regions, which explains their role in the macroscopic organization and planning of spatial behavior.

Finally, with guidance that produces adequate IPs (satisfy the group and subgoal equivalences), the environment can be abstracted by the subgoals, and the agent-environment interactions can be simplified to a guidance behavior relative to subgoals. The set of all interaction patterns for an agent and its environment is called an *interaction pattern library*, which can be denoted Π , and enables a parsimonious representation of the extended state space $\tilde{\mathcal{X}}$:

$$\tilde{\mathcal{X}} \rightarrow \Pi := \{\pi_1, \pi_2, \dots\} \quad (3)$$

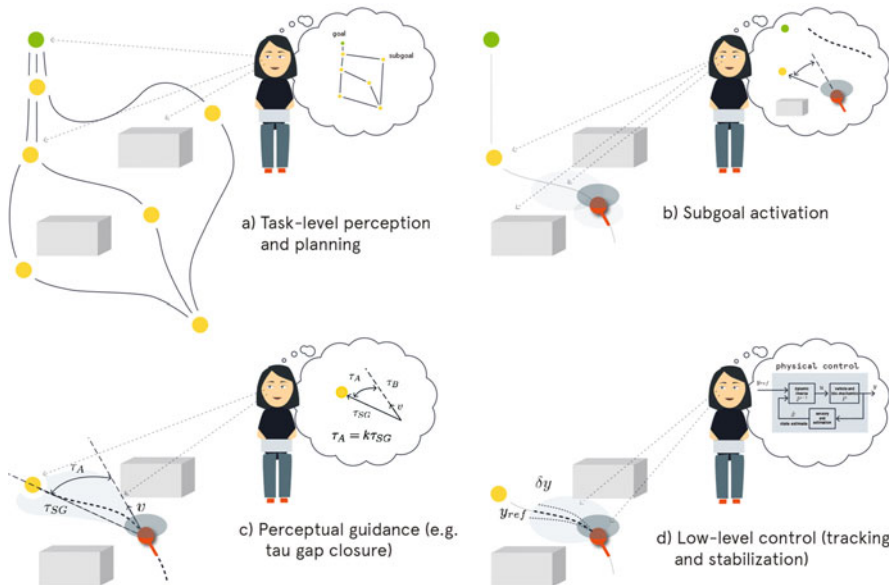
With this abstraction, the spatial behavior can be described and represented using the symbols belonging to Π .

Interaction Patterns and Systems Modeling

IPs are interesting from a system's perspective, because they combine functional and behavioral dimensions (Mettler et al. 2017). The functional properties arise from the IPs' grounding in the closed-loop perception and guidance mechanisms. As shown in Fig. 2, the IPs are produced by the guidance process and represent the basic behavioral elements leveraged by the planning and control hierarchy. They also make it possible to delineate the hierarchical control functions, their mechanisms, and associated information.

Functional Hierarchical Model Overview

Figure 3 elaborates these ideas using Rasmussen's three levels: symbol, cue, and signal. At the top level, the behavior is described by a sequence of IPs and corresponding subgoals relative to the environment elements and task objectives (Fig. 3a). This allows formalization of



Interaction Patterns as Units of Analysis in Hierarchical Human-in-the-Loop Control, Fig. 3 Illustration of the levels of operator's interactions for the spatial control task in Fig. 1. The levels are organized based on the functional hierarchical model, highlighting the parallels with Rasmussen's levels of information processing. The

bubbles indicate the mental models associated with the different levels of control. (a) Task-level perception and planning. (b) Subgoal activation. (c) Perceptual guidance (e.g., tau gap closure). (d) Low-level control (tracking and stabilization)

the task representation, which can, for example, be described using a subgoal graph (Verma and Mettler 2016). The plan is provided as a sequence of subgoals $g_1, g_2, \dots, x_g$. The perceptual function at this level is responsible for the identification of task elements and designation of subgoals (Fig. 3a).

At the intermediate level, the system component is conceptualized as the interaction pattern IP. The control actions correspond to the settings that configure the physical control/perceptual system to produce the desired agent-environment interaction (e.g., Tau or other guidance in Fig. 3c). The guidance process produces the trajectory needed to reach the active subgoal. The perceptual function at this level is responsible for extracting necessary cues (e.g., Tau motion gaps in Fig. 3b).

At the lowest level, the behavior is produced by a control process shaping the dynamic response needed to track the trajectory profiles provided by the guidance law (Fig. 3d). Typical models at this level include nonlinear feed-forward inverse controller, together with feedback control actions, which produce the control reference trajectories necessary to implement the behavior (control input u and reference y_{ref}) Mettler and Kong (2013).

Interaction Patterns in Applications

The IPs can be used in modeling and analysis of a range of spatial control applications for human-in-the-loop systems and artificial agents.

Modeling Applications

Specifically, the subgoal equivalence provides the basis for the decomposition of behavior into hierarchical elements. The decomposition proceeds according to the hierarchical model, starting with the subgoal equivalence, which enables segmentation according to the IPs. This “lawful” decomposition makes it possible to tear the behavior apart along functionally meaningful segments (see Mettler and Kong 2012). The dynamic behavior and trajectories associated with the identified IPs can subsequently be used to analyze and model the behavior, including:

1. Modeling the guidance and control mechanisms underlying the IPs π_i .
2. Modeling of the higher-level functions, including planning, visual attention, memory representation, and learning.

To illustrate this approach, Feit and Mettler (2016) provides an example of functional modeling of the control and perceptual processes of guidance primitives. The functional properties were captured using structure learning techniques (Bayesian graph). Subsequently, trajectories within control modes were used to identify perceptual and guidance mechanisms.

Verma and Mettler (2016) provides another example, here an investigation of learning task environments using IPs as the unit of behavior. Using the IPs for the analysis helps raise the level of analysis into visual attention and memory structure associated with the learning process.

Engineering Design Applications

In parallel to modeling and analysis, the hierarchical framework can be used to design and implement control augmentations in human-machine systems. The general idea of this work is that the hierarchical framework enables design of operator interactions based on the functional organization of the information processing. Tseng and Mettler (2016) demonstrates how IPs can be used to model human strategies in search applications with a tele-operated mobile robot. In this example, the subgoals are used as control abstractions to augment the operator (Tseng and Mettler 2020). In Andersh and Mettler (2018), the opposite strategy is applied, the human visual gaze inputs provide the high-level subgoal information, and an artificial low-level controller assists with the guidance and control processes.

Future Research and Applications

Modeling and analysis of humans’ hierarchical control architecture is critical to enabling more rigorous and comprehensive human skill analysis, as well as the design of natural and effective human-machine interfaces. The interaction patterns and functional hierarchical model provide a holistic, system-level understanding of behavior

in artificial and natural agents. The improved knowledge about the principles of hierarchical control is also critical for designing computationally efficient and versatile guidance algorithms for autonomous vehicles. In particular, the idea would be to continue leveraging concepts from ecological psychology, such as direct perception and the concept of affordances (Gibson 1977; Stoffregen 2003). This understanding can also be helpful for the elaboration of new design principles for machine vision, for example, by integrating the functions into complete units of operation, such as the coordination of perceptual and control dimensions leading to IPs.

Skill acquisition can be conceptualized as the problem of learning hierarchical control architecture. The hierarchical functional model can be used for human learning, skill analysis, and machine learning approaches. Hierarchical control, and more specifically, the definition of subgoals, has been central to more efficient reinforcement learning algorithms, including deep hierarchical learning. Many demonstrations, however, have so far focused on video games which don't replicate the complexity of natural environment interactions. Finally, the functional understanding enabled by a more detailed hierarchical model can be useful to formulate stronger hypotheses in cognitive neuroscience.

Cross-References

- [Human Decision-Making in Multi-agent Systems](#)
- [Physical Human-Robot Interaction](#)
- [Trajectory Generation for Aerial Multicopters](#)
- [Trust Models for Human-Robot Interaction and Control](#)

Bibliography

- Andersh J, Mettler B (2018) Modeling the human visuomotor system to support remote-control operation. *Sensors* 18(9):2979
- Chase WG, Simon HA (1973) Perception in chess. *Cogn Psychol* 4(1):55–81
- Feit A, Mettler B (2016) Extraction and deployment of human guidance policies. In: Submitted
- Flash T, Hochner B (2005) Motor primitives in vertebrates and invertebrates. *Curr Opin Neurobiol* 15(6):660–666
- Frazzoli E, Dahleh MA, Feron E (2002) Real-time motion planning for agile autonomous vehicles. *J Guid Control Dyn* 25(1):116–129
- Gibson JJ (1977) The theory of affordances. *Perceiving, acting, and knowing: toward an ecological psychology*, pp 67–82
- Kong Z, Mettler B (2013) Modeling human guidance behavior based on patterns in agent–environment interactions. *IEEE Trans Hum-Mach Syst* 43(4):371–384
- Lee DN (1998) Guiding movement by coupling taus. *Ecol Psychol* 10(3–4):221–250
- McRuer D, Krendel E (1974) Mathematical models of human pilot behavior. Technical Report AGARD-AG-188, AGARD
- Mettler B, Kong Z (2012) Hierarchical model of human guidance performance based on interaction patterns in behavior. In: 2nd international conference on application and theory of automation in command and control systems, ATACCS, London
- Mettler B, Kong Z (2013) Mapping and analysis of human guidance performance from trajectory ensembles. *IEEE Trans Hum-Mach Syst* 43(1):32–45
- Mettler B, Kong Z, Li B, Andersh J (2015) Systems view on spatial planning and perception based on invariants in agent–environment dynamics. *Front Neurosci* 8:439
- Mettler B, Verma A, Feit A (2017) Emergent patterns in agent–environment interactions and their roles in supporting agile spatial skills. *Ann Rev Control* 44:252–273. <https://doi.org/10.1016/j.arcontrol.2017.09.001>. <http://www.sciencedirect.com/science/article/pii/S1367578817301219>
- Mumford D, Desolneux A (2010) Pattern theory: the stochastic analysis of real-world signals. A K Peters
- Rasmussen J (1983) Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Trans Syst Man Cybern* (3):257–266
- Simon HA (1990) Invariants of human behavior. *Ann Rev Psychol* 41(1):1–20
- Stoffregen TA (2003) Affordances as properties of the animal–environment system. *Ecol Psychol* 15(2):115–134
- Sutton RS, Precup D, Singh SP (1998) Intra-option learning about temporally abstract actions. In: *ICML*, vol. 98, pp 556–564
- Tseng KS, Mettler B (2016) Human planning and coordination in spatial search problems. *IFAC-PapersOnLine* 49(32):222–227
- Tseng KS, Mettler B (2020) Analysis and augmentation of human performance on telerobotic search problems. *IEEE Access* 8
- Verma A, Mettler B (2016) Investigating human learning and decision-making in navigation of unknown environments. In: 1st IFAC conference on cyber-physical and human-systems, Brazil

Interactive Environments and Software Tools for CACSD

Vasile Sima

National Institute for Research and Development
in Informatics, Bucharest, Romania

Abstract

The main functional and support facilities offered by interactive environments and tools for Computer-Aided Control System Design (CACSD) and reference examples of such software systems are presented, from both a user and a developer perspective. The essential functions these environments should possess and requirements which should be satisfied are discussed. The importance of reliability and efficiency is highlighted, besides the desired friendliness and flexibility of the user interface. Widely used environments and software tools for CACSD, including MATLAB, Mathematica, Maple, and the SLICOT Library, serve as illustrative examples.

Keywords

Automatic control · Controller design ·
Numerical algorithms · Simulation · User
interface

Introduction

The complexity of many processes or systems to be controlled and the strong performance requirements to be fulfilled nowadays make it very difficult or even impossible to design suitable control laws without resorting to dedicated software tools. Computer-Aided Control System Design (CACSD) is the use of computer programs to support the creation, analysis, evaluation, or optimization of a control system design. CACSD is a specialization of Computer-Aided Design (CAD) for control systems. CAD is used in many domains, to enhance designer's productivity and

the design quality and to manage the design versions and documentation.

The interactive environments and tools for CACSD have evolved significantly during the last decades, in parallel with the developments of numerical linear algebra, scientific computations, and computer hardware and software, including programming and networking capabilities. Starting from simple collections of specialized tools for solving well-defined system analysis and design problems, CACSD became increasingly more sophisticated and powerful, allowing complicated tasks to be orchestrated for fully covering the control engineering design, prototyping, testing, and deployment. Modelling, system analysis and synthesis, and control system assessment are activities which are assisted by the nowadays advanced CACSD environments and software tools. The main aim is to help the designer to concentrate on the design problem itself, not on theoretical approaches, numerical algorithms, and computational details. Moreover, CACSD environments allow the developers and users to do conceptual thinking but also programming and debugging at a higher level of abstraction, in comparison with standard programming languages, like Fortran, C/C++, or JavaTM.

There are both commercial or free and open-source CACSD environments and tools. State-of-the-art CACSD systems exist for several platforms (Windows, Linux/UNIX, and Mac OS X). Multiple high speed CPUs, graphics cards, and large amounts of RAM are well-suited to perform graphically and computationally intensive tasks. A common feature is the presence of a "friendly" graphical user interface, but often a dedicated command language is also available. The user interacts with the CACSD environment either by specifying the model or control structure, the design requirements, and the values of essential parameters or by selecting and combining the tools to be used. The process can be repeated until a satisfactory behavior is obtained.

Usually, the underlying computational tools, on which the interactive environments are based on, are hidden to the user. Moreover, software for extensive testing is not normally provided, but demonstrators running few examples are offered.

Unfortunately, even mathematically simple problems of small dimension can conduct to wrong results when using unsuitable algorithms. Illustrative control-related examples are given, e.g., in Van Huffel et al. (2004). Since system analysis and design tasks usually involve sequential or iterative solution of complex subproblems, it follows that the quality of the intermediate results is of utmost importance. Consequently, the interactive environments for CACSD should be based on reliable, efficient, and thoroughly tested computational building blocks, which are called at the lower layers of calculations. These blocks constitute the computational engine of an interactive environment.

Figure 1 gives a typical hierarchy of the software components incorporated in an interactive CACSD environment.

Interactive Environments and Software Tools

The interactive environments can be divided into general purpose numerical systems, such as MATLAB® from The MathWorks, Inc., Octave, Scilab, Julia, or OML, and computer algebra systems, such as Mathematica from Wolfram, or Maple from Waterloo Maple Inc. (Maplesoft).

The software tools for CACSD are also formally divided into computational and support tools. The SLICOT Library and Dymola serve as illustrative examples.

Main Functionality

A comprehensive set of functions of, and requirements for interactive environments for control engineering, is described in MacFarlane et al. (1989), but such a set has probably not yet been covered by any single environment. State-of-the-art interactive environments and software tools for CACSD include many attractive functional features:

- define or find (via first principles or system identification) various system models (e.g., state-space models or transfer-function matrices), and convert between different representations;
- find reduced order (or simplified) models, which can more economically be used for simulation, control, prediction, etc.;
- analyze basic system properties, like stability, controllability, observability, stabilizability, detectability, minimality, properness, etc.;
- analyze interactively the behavior of a control system for various scenarios;

Interactive CACSD environment (e.g., MATLAB, Mathematica, Maple) (for modelling, simulation, analysis, synthesis, etc.)
Toolboxes or packages with executables or functions written in the environment language
(Graphical) User interface, Interactive language, Graphical functions, API
CACSD subroutine libraries (e.g., SLICOT)
Mathematical subroutine libraries (e.g., LAPACK, ARPACK, IMSL, NAG)
Computer-optimized mathematical libraries or their generators (e.g., BLAS, MKL, ATLAS)
Libraries of intrinsic functions (e.g., in Fortran or C/C++)

Interactive Environments and Software Tools for CACSD, Fig. 1 Hierarchy of the software components incorporated in an interactive CACSD environment

- provide alternative tools for different categories of users, from novice to expert and from classical to “modern” or advanced analysis and synthesis techniques, in time domain or frequency domain;
- provide a wide range of tools, covering modelling, system identification, filtering, control system design, simulation, real-time behavior, hardware-in-the-loop simulation, and code generation for easy deployment; ensure their interoperability;
- allow the user to add extensions at various levels: new functions, interfaces, or even toolboxes or packages, which can be made available to a general community; allow customization.

In addition to the functional and computational tools, essential components of an interactive environment are the user interface, the application program interface (API), and the support tools, which enable to easily specify, document and store a design solution, visualize and interpret the results, export the results to other applications for further processing, generate reports, etc.

It is a common feature of interactive environments for CACSD to address the requirements of a large diversity of users, in various stages of familiarity with the environment. This feature is expressed, e.g., by the option to use either a graphical user interface or a command language to call and sequence various computational procedures. In addition, tools for easy building new computational or graphical procedures, or for managing the codes and results, are often included. The command language often operates both on low-level data constructs, such as a matrix, and on high-level ones (e.g., system objects) and allows operator overloading (e.g., taking $G_1 * G_2$ as the result of a series interconnection of the systems represented by G_1 and G_2).

An environment for CACSD should integrate advanced user interfaces and API, a collection of problem-solvers based on reliable and efficient numerical and possibly symbolic algorithms, and tools for visualizing and interpreting the results.

Widely used such environments are, for instance, MATLAB, Mathematica, or Maple, presented in the next subsection. There are also environments dedicated to modelling and simulation, which cover a broad range of technical and engineering computations, including those for mechanical, electrical, thermodynamic, hydraulic, pneumatic, or thermal systems. An example is Dymola, presented in a subsequent subsection, or COMSOL Multiphysics®.

Reference Interactive Environments

MATLAB (MATrix LABoratory) is an integrated, interactive environment for technical computing, visualization, and programming (MathWorks® 2013). Based on a powerful high-level interpreter language and development tools, an easy-to-use, flexible, and customizable graphical user interface, complemented with attractive visualization capabilities, and open for extensions with new toolkits, MATLAB can be used for solving intricate scientific and engineering problems, as well as for development and deployment of applications.

MATLAB® and Simulink® are registered trademarks of The MathWorks, Inc. MATLAB, Simulink, and several toolboxes, including System Identification Toolbox, Control System Toolbox, and Robust Control Toolbox, are suitable for solving various control engineering problems; other toolboxes, such as Signal Processing Toolbox, Optimization Toolbox, and Symbolic Math Toolbox, offer additional useful facilities. See <https://www.mathworks.com/products/>.

Simulink is a high-level implementation of the engineering approach, based on block diagrams, to analyze and design control systems. It is also a powerful modelling and multi-domain simulation and model-based design tool for dynamic systems, which supports hierarchical system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink offers a graphical editor, customizable block libraries, and solvers for modelling and simulation. The models may include MATLAB algorithms, and the simulation results may be further processed in MATLAB.

Managing projects (files, components, data), connecting to hardware for real-time testing, and deploying the designed system are additional, useful Simulink features. Real-Time Workshop code generation allows to speed up the design and implementation, by generating syntactically and semantically correct code which can be uploaded to the target machine. The capabilities for rapid prototyping are complemented by capabilities for developing and testing new mathematical or control theories and algorithms.

MATLAB offers support for object-oriented programming and interfacing with other languages, or connecting to similar environments. When using the command line interface, MATLAB helps the user, e.g., by showing the arguments of the typed MATLAB functions; also, it allows execution profiling, for increasing the computational efficiency, and its editor can suggest changes in the user M-functions for improving the performance. MATLAB users may upload their own contributions to the MATLAB Central website or may download tools developed by other people. User feedback is used by the MATLAB developers to improve its functionality, reliability, and efficiency.

Commercial competitors to MATLAB include Mathematica, Maple, and IDL; free open-source alternatives are, e.g., GNU Octave, FreeMat, and Scilab, intended to be mostly compatible with the MATLAB language. For instance, a set of free CACSD tools for GNU Octave version 3.6.0 or beyond has been recently developed (see <https://octave.sourceforge.io/control/>). The Octave extension package called *control* is based on the SLICOT Library and includes functionalities for system identification, system analysis, control system design (including H_∞ synthesis), and model reduction, which are the basic steps of the control engineer design workflow.

Mathematica is an interactive environment which supports complete computational workflows, making it suitable for a convenient endeavor from ideas to deployed solutions (see <https://www.wolfram.com/mathematica/>). Mathematica offers, e.g., tools for 2D and 3D data and function visualization and animation,

numeric and symbolic tools for discrete and continuous calculus, a toolkit for adding user interfaces to applications, control systems libraries, tools for parallel programming, etc. High-performance computing capabilities include the use of packed and sparse arrays, multi-precision arithmetic, automatic multi-threading on multicore computers (based on processor specific optimized libraries), hardware accelerators, support for grid technology, and CUDA and OpenCL GPU hardware. Mathematica and SystemModeler (based on Modelica[®] language) offer numerous built-in functions which allow to design, analyze, and simulate continuous- and discrete-time control systems, simplify models, interactively test controllers, and document the design. Both classical and modern techniques are provided. A powerful symbolic-numeric computation engine and highly efficient numerical algorithms are used. Mathematica allows to define the system models in a natural form. It can analyze not only numeric systems but also symbolic ones, represented by state-space or transfer-function models. The computational precision and algorithms can be automatically controlled and selected, respectively.

Maple is a computer algebra system, which combines a powerful engine for mathematical calculations with an intuitive user interface (see <https://www.maplesoft.com/>). Classical mathematical notation can be used, and the interface is customizable. Arbitrary precision numerical computations, as well as symbolic computations, can be performed. The Maple language is provided by a small kernel. NAG Numerical Libraries, ATLAS libraries, and other libraries written in this language are used for numerical calculations. Symbolic expressions are stored as directed acyclic graphs. Maple includes some CACSD tools for linear and nonlinear dynamic systems. For instance, the built-in package DynamicSystems (available since Maple 12 release) covers the analysis of linear time-invariant systems. Numerical solvers for Sylvester and Lyapunov equations have been added to the Linear Algebra Packages in Maple 13, and solvers for algebraic Riccati equations – based on SLICOT Library routines –

have been included in Maple 14 (available in multiple precision arithmetic since Maple 15). Moreover, the MapleSim environment, based on Modelica, enables symbolic simplification, numerical solution of the differential-algebraic equations (DAEs), and model post-processing (sensitivity analysis, linearization, parameter optimization, code generation, etc.). Its Control Design Toolbox provides solutions for optimal control, Kalman filtering, pole assignment, etc. Bidirectional communication with MATLAB is possible.

Compose, OML, and Activate is a new software environment for modelling and simulation (Campbell and Nikoukhah 2019). It is a mixture of open-source and freely distributed software, including Altair ComposeTM, Altair ActivateTM, and Open Matrix Language (OML). Continuous-time, discrete-time, and hybrid systems can be dealt with.

Computational Tools

The computational tools for CACSD implement the main numerical algorithms of the systems and control theory and should satisfy several strong requirements:

- reliability, or guaranteed accuracy, implies the use of numerically stable algorithms as much as possible and the estimation of the problem sensitivity (conditioning) and of the solution accuracy; backward numerical stability ensures that the computed results are exact for slightly perturbed original data;
- robustness, mainly ensured by avoiding overflows, harmful underflows, and unacceptable accumulation of rounding errors; scaling the data may be essential;
- computational efficiency, important for large-scale engineering design problems or for real-time control;
- ease of use, achieved by simplified user interface (hiding the details), and default values for algorithmic parameters, such as tolerances;
- wide scope and rich functionality, which address the range of problems and system representations that can be handled;

- portability to various platforms, in the sense of functional correctness;
- reusability, in building several dedicated software systems or environments.

SLICOT Library (Benner et al. 1999; Van Huffel et al. 2004) is one of the most comprehensive libraries for control theory numerical computations, containing over 550 subroutines which cover system analysis, benchmark and test problems, data analysis, filtering, identification, mathematical routines, some capabilities for nonlinear systems, synthesis, system transformation, and utility routines (see <http://www.slicot.org/>). The requirements above have been taken into account in the SLICOT Library development. Some of the SLICOT components are used in several interactive environments for CACSD, including MATLAB, Maple, Scilab, and Octave *control* package. The library is still under development. It is worth mentioning the recent focus on structure-preserving algorithms, which offer increased accuracy, reliability, and efficiency, in comparison with standard solvers. Many procedures for optimal control and filtering, model reduction, etc. can benefit from using the “structured” solvers. There are also separate SLICOT-based toolboxes for MATLAB (Benner et al. 2010).

SLICOT Library routines, and many functions offered by interactive environments for CACSD, call components from the Basic Linear Algebra Subprograms (BLAS; see Dongarra et al. 1990 and the references therein) and Linear Algebra Package (LAPACK, Anderson et al. 1999). This approach enhances portability and efficiency, since optimized BLAS and LAPACK Libraries are provided for major computer platforms.

Support Tools

The support software tools for CACSD offer additional capabilities compared to computational tools. They may include alternative algorithms, symbolic computations (usually, for low-dimensional problems), extended functionality, e.g., for modelling/simulation of nonlinear systems, code generation, etc. The

support tools can be used by software developers of CACSD environments or computational tools or directly by other users. For instance, symbolic calculations are useful for checking the accuracy of numerical algorithms. The code generation facility offers a safe and convenient support for deploying a design solution to the control hardware. A reference support software tool is briefly presented below.

Dymola (Dynamic modeling Laboratory), from Dassault Systemes (see <https://www.3ds.com/products/catia/portfolio/dymola>), deals with high-fidelity modelling and simulation of complex systems from various domains, like aerospace, automotive, robotics, process control and other applications. Compatible and comprehensive model libraries, developed by leading experts, exist for many engineering branches. The users may create their own libraries or adapt existing libraries. This flexibility and openness is provided by the use of the open, object-oriented modelling language Modelica®, currently further developed by the Modelica Association.

Equation-oriented models, based on DAEs, and symbolic manipulation are used, stimulating the reuse of components and enhancing the reliability and efficiency of the calculations. This approach enables to simplify generating the equations which result from interconnecting various subsystems and to deal with algebraic loops and structurally singular models. Algebraic loops are encountered when some auxiliary variables depend algebraically upon each other in a mutual way (Cellier and Elmqvist 1992). Structural singularities are related to DAE of index higher than 1.

Dymola allows to perform hardware-in-the-loop simulation and real-time 3D animation. A model can be built by graphical composition, connecting components from various libraries using drag-and-drop operations. The parameters a model depends on can be tuned either by *parameter estimation* (also called *model calibration*), which minimizes the error between the physical measurements and simulation results, or by optimization, which minimizes certain performance criteria. Sometimes, e.g., when designing certain controllers, the criteria values are

obtained by simulation. Dymola offers also facilities for model management, including checking, testing, encrypting, or comparing models, and version control.

Summary and Future Directions

The main functional and support facilities offered by interactive environments and software tools for CACSD and reference examples have been presented. Their remarkable evolution during the past decades, combined with the importance of the design solutions they offer, are strong arguments that the CACSD software arsenal will continue to evolve, and more reliable, efficient, and powerful systems will come into place. Progress is expected at all levels, including basic algorithms and numerical and symbolic libraries, but also command languages, user interfaces, human-machine communication, and associated hardware. Tools for adaptive, nonlinear, and distributed control systems design should be developed and integrated. Artificial intelligence support might be required to add expert capabilities to the forthcoming interactive environments. It is expected that deep neural networks will be increasingly used for modelling complex processes.

Cross-References

- [Computer-Aided Control Systems Design: Introduction and Historical Overview](#)
- [Descriptor System Techniques and Software Tools](#)
- [Fault Detection and Diagnosis: Computational Issues and Tools](#)
- [Model Order Reduction: Techniques and Tools](#)
- [Multi-domain Modeling and Simulation](#)
- [Optimization-Based Control Design Techniques and Tools](#)
- [Robust Synthesis and Robustness Analysis Techniques and Tools](#)
- [System Identification: An Overview](#)
- [Validation and Verification Techniques and Tools](#)

Recommended Reading

CACSD is well presented in many textbooks. A recent one is Chin (2012), which covers modelling, control system design, implementation, and testing and describes practical applications using MATLAB and Simulink. Many IFAC (International Federation of Automatic Control) and IEEE (The Institute for Electrical and Electronics Engineers, Inc.) international conferences and symposia have been dedicated to CACSD, going back more than three decades. A wealth of material is available, e.g., on the IEEE Xplore (<https://ieeexplore.ieee.org>), containing the proceedings of many of the IEEE CACSD events. A recent event has been the 2016 IEEE Conference on CACSD. Similar IEEE events were regularly held between 1989 and 2013.

Bibliography

- Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J, Du Croz J, Greenbaum A, Hammarling S, McKenney A, Sorensen D (1999) LAPACK Users' guide, 3rd edn. SIAM, Philadelphia
- Benner P, Mehrmann V, Sima V, Van Huffel S, Varga A (1999) SLICOT – a subroutine library in systems and control theory. In: Datta BN (ed) Applied and computational control, signals, and circuits, vol 1. Birkhäuser, Boston, pp 499–539
- Benner P, Kressner D, Sima V, Varga A (2010) Die SLICOT-Toolboxen für Matlab. at – Automatisierungstechnik 58:15–25
- Campbell SL, Nikoukhah R (2019) Compose, OML, and activate: a new software suite for modeling and simulation. In: Proceedings of 2019 annual IEEE international conference (SysCon 2019), Orlando, 8–11 Apr 2019
- Cellier FE, Elmqvist H (1992) The need for automated formula manipulation in object-oriented continuous-system modeling. In: Proceedings of the 1992 IEEE symposium on computer-aided control system design, Tucson, 17–19 Mar 1992, pp 1–8
- Chin CS (2012) Computer-aided control systems design: practical applications using MATLAB® and Simulink®. CRC Press, Boca Raton
- Dongarra JJ, Du Croz J, Duff IS, Hammarling S (1990) Algorithm 679: a set of level 3 basic linear algebra subprograms. ACM Trans Math Softw 16:1–17, 18–28
- MacFarlane AGJ, Grübel G, Ackermann J (1989) Future design environments for control engineering. Automatica 25:165–176
- MathWorks® (2013) MATLAB® Primer. R2013a. The MathWorks, Inc., Natick
- Van Huffel S, Sima V, Varga A, Hammarling S, Delebecque F (2004) High-performance numerical software for control. IEEE Control Syst Mag 24:60–76

Intermittent Image-Based Estimation

Warren E. Dixon

Department of Mechanical and AeroMechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA

Abstract

Image feedback can be used to estimate Euclidean distances between features in an image and/or relative motion between an image feature and the camera. Typical current methods assume that the image feature is continuously observed; yet, in most practical scenarios, the image feature can be occluded. Image occlusion/intermittency segregates the image dynamics into two subsystems: when the feature is visible, feedback is available and the estimator/observer is stabilizable, otherwise it's unstable. This entry discusses the use of switched/hybrid systems methods including the development of sufficient dwell-time conditions to ensure the stability of such estimators/observers.

Keywords

Nonlinear estimation · Image estimation · Hybrid system · Lyapunov methods · Intermittent sensing

Introduction

Numerous advances have led to widespread adoption of image-based sensing. One of the technical challenges of image-based feedback for navigation and visual servo control for autonomous systems is the lack of depth information (i.e., the distance from the camera to an image feature along

the focal axis), resulting in scale ambiguity. Typical solutions to recover the depth information use multiple views of a feature. Multiple views can be obtained using multiple physical cameras (e.g., a stereo pair) with calibrated relative position and orientation (pose) to determine the properly scaled Euclidean coordinates of image features (i.e., structure estimation) and/or the relative motion between the camera and the feature(s) (i.e., motion estimation) (Faugeras 1993; Hartley and Zisserman 2000; Chaumette and Hutchinson 2006a). Alternatively, monocular solutions use multiple images taken by a moving camera to determine the structure from known motion of the camera (structure from motion (SfM)), or motion from known structure, either using image geometry, provided a sufficient number of features on an object are tracked, or using an estimator/observer (cf. Chaumette et al. 1996; Chaumette and Hutchinson 2006b; Dani et al. 2012; Hu et al. 2008; Dixon et al. 2003). A common assumption among these strategies is that the image features are continuously available.

In many realistic scenarios, image feedback may be intermittent due to failures in feature tracking due to feature occlusions or features leaving the camera field-of-view (FoV) due to motion constraints of the camera or even purposeful motion by the camera. Various path planning and control methods have been motivated by the need to ensure uninterrupted feedback, where the desired trajectory or behavior of such systems is inherently constrained. For example, systems with nonholonomic constraints may experience non-ideal, sharp-angled, or non-smooth trajectories to keep a navigational landmark in the FoV. Moreover, such approaches are still not robust to arbitrary occlusions of the image feature that result from environmental conditions. Typically when features are temporarily not visible, the estimator/observer is reinitialized with the previous state estimate when the feature is reacquired; however, as shown in Liberzon (2003), such switching is insufficient: the estimates may not converge.

In contrast to the strategy of altering the trajectory of the camera to keep the image feature in the FoV, this entry examines recent and

emerging efforts that use switched/hybrid systems methods to examine the stability of image-based estimates in the presence of feature intermittency. These methods include the development of dwell-time conditions that provide sufficient (and potentially conservative) timing conditions which indicate how long features can be occluded, relative to the amount of time the features are visible. A challenge is that image occlusion/intermittency segregates the image dynamics into two subsystems: when the feature is visible, feedback is available and the estimator/observer is stabilizable, otherwise it is unstable. Despite this challenge, tolerance to image feedback intermittency provides inherent robustness to the observer and enables new dwell-time-aware path-planning strategies that ease trajectory generation constraints that require the image feature to remain visible.

Image-Based Estimation Dynamics

The unknown Euclidean coordinates, $X, Y, Z \in \mathbb{R}$, of a tracked feature point can be related to the states of an image-based observer by the normalization $x = [x_1, x_2, x_3]^T \triangleq \left[\frac{X}{Z}, \frac{Y}{Z}, \frac{1}{Z} \right]^T \in \mathbb{R}^3$. Using projective geometry, the measurable image coordinates, $p \in \mathbb{R}^3$, of the feature point can be related to the normalized Euclidean coordinates as $p = A \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$, where $A \in \mathbb{R}^{3 \times 3}$ is a known, invertible camera intrinsic parameter matrix (Faugeras 1993; Hartley and Zisserman 2000; Chaumette and Hutchinson 2006a). The perspective state dynamics can be written as $\dot{x} = g(t, x)$, where $g(t, x) : [0, \infty) \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a nonlinear function that depends on the partially measurable states (i.e., x_3 is not measurable).

A common objective in image-based navigation and control research is the SfM problem, where the goal of estimating the Euclidean coordinates of the target position can be quantified by the state estimate error

$$e \triangleq x - \hat{x}, \quad (1)$$

where $\hat{x} \in \mathbb{R}^3$ denotes the state estimate provided by an observer. If the image feature is intermit-

tently tracked, then the evolution of e is defined by the family of systems

$$\dot{e} = f_p(t, x, \hat{x}) \quad (2)$$

where $f_p : [0, \infty) \times \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ and $p \in \{s, u\}$, where s is an index for the resulting subsystem when the target is visible and u is an index for the subsystem when the target is not visible. When the target is in view, the first two elements of the state vector x (i.e., x_1 and x_2) are measurable, and the closed-loop error dynamics are

$$f_s = g(t, x) - \dot{\hat{x}}, \quad (3)$$

where $\dot{\hat{x}}$ is defined by an observer. For the error dynamics in (3), numerous observers have been developed that achieve a variety of stability results. However, when the feature is not visible, the state estimates cannot depend on x , and the resulting error system is unstable. For example, a zero order hold (ZOH) could be used (i.e., $\dot{\hat{x}} = 0$), and the unstable error dynamics would be given by

$$f_u = g(t, x). \quad (4)$$

Switched Systems Analysis

Switched systems methods (cf. Liberzon 2003; Goebel et al. 2012) provide an analysis framework to evaluate the stability of state estimation error dynamics in (2) after a series of instances when the image feature is visible or not. The underlying concept is that if the image feature is visible long enough and the instances without feedback are short enough, then the state estimate over the entire trajectory could remain stable. To understand the relative timing for the estimate to be stable, switched systems analysis includes the development of a dwell-time. The dwell-time is typically expressed as a ratio of the time spent in each subsystem, along with the convergence rate in the stable subsystem and the divergence rate in the unstable subsystem. The intermittent

image-based estimation problem is challenging because it inherently involves switching between stabilizable and unstable subsystems, and if the image feature is not visible long enough then stability may never be achieved. Therefore, the concept of an average dwell-time condition (a weaker dwell-time condition where the time in the stable subsystem is longer than the unstable region on average) is difficult to develop. Moreover, when switching between stable and unstable subsystems, two different dwell times are often required, to determine a minimum amount of time to dwell in the stable subsystem and the maximum amount of time in the unstable that can be tolerated. Typically, such dwell-time conditions provide a sufficient rule for a system to follow to ensure stability; however, often for image-based estimation, the ratios of time in each subsystem are arbitrary. In these cases, the dwell-time conditions provide a sufficient condition for a decision-maker to check as a means to trust or not trust the estimated states.

Because the dwell-time conditions are inherently linked to the ratios of the rates of convergence and divergence, exponential stability of the observer is desired. While various exponentially convergent observers have been developed (cf. Chen and Kano 2002; Dani et al. 2012), open questions remain for developing dwell-time conditions for switching between stable and unstable subsystems for asymptotic convergence rates that are more common for the inherently nonlinear image dynamics. Moreover, exponential divergence can also be difficult to determine. For example, in Parikh et al. (2018), the ZOH scenario yielded a faster than exponential divergence rate, which resulted in complex trigonometric dwell-time conditions. However, results such as Parikh et al. (2017) illustrate that by replacing the ZOH observer with a model-based state-predictor during the periods when feedback is unavailable, the divergence rate can be bounded by an exponential envelope, facilitating the development of traditional (and less restrictive) dwell-time conditions. That is, the dwell-time conditions result in the intuitive conclusion that strategies with better state prediction allow the image feature to be occluded for longer periods; however, such

predictors often require more stringent assumptions on knowledge about the camera and image feature relative motion.

Summary and Future Directions

Various strategies are available to analyze image-based estimators in the presence of intermittent visibility of image features. An alternative to the aforementioned switched systems methods is to treat the availability of feedback as a stochastic system with a known distribution and the goal of proving the observer converges in expectation. Another alternative is to develop relationships between multiple visible features, which are assumed to have a constant relative pose, so that when a subset of the features are occluded, the remaining visible features can be used to approximate the trajectory of the hidden features (cf. Mehta et al. 2010). Opportunities also exist to develop such methods within a switched systems framework.

Switched systems methods provide dwell-time conditions that can be checked to determine the stability of image-based observers, with arbitrary loss of features. These conditions also provide design guidelines for potential new advances in image-guided autonomy. Specifically, new image-based path planners could be developed that are informed by the dwell-time conditions to allow the image features to purposefully leave the FoV for controlled periods of time as a means to yield more favorable trajectories. New state-prediction results to minimize the error growth when features are not visible also present new opportunities. With continued development and maturity of switched systems methods (e.g., stochastic switched systems methods), more practical and robust image-based estimation strategies are likely to result.

Cross-References

► [Image-Based Estimation for Robotics and Autonomous Systems](#)

- [Image-Based Formation Control of Mobile Robots](#)
- [Image-Based Robot Control](#)
- [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)
- [Linear Systems: Discrete-Time Impulse Response Descriptions](#)
- [Lyapunov's Stability Theory](#)
- [Optimal Control and the Dynamic Programming Principle](#)
- [Quantitative Feedback Theory](#)
- [Sampled-Data Systems](#)
- [Switching Adaptive Control](#)
- [2.5D Vision-Based Estimation](#)

Bibliography

- Chaumette F, Hutchinson S (2006a) Visual servo control part I: basic approaches. *IEEE Rob Autom Mag* 13(4):82–90
- Chaumette F, Hutchinson S (2006b) Visual servo control part II: advanced approaches. *IEEE Rob Autom Mag* 14(1):109–118
- Chaumette F, Boukir S, Bouthemy P, Juvin D (1996) Structure from controlled motion. *IEEE Trans Pattern Anal Mach Intell* 18(5):492–504
- Chen X, Kano H (2002) A new state observer for perspective systems. *IEEE Trans Autom Control* 47(4):658–663
- Dani A, Fischer N, Dixon WE (2012) Single camera structure and motion. *IEEE Trans Autom Control* 57(1):241–246
- Dani A, Fischer N, Kan Z, Dixon WE (2012) Globally exponentially stable observer for vision-based range estimation. *Mechatron* 22(4):381–389. Special issue on visual servoing
- Dixon WE, Fang Y, Dawson DM, Flynn TJ (2003) Range identification for perspective vision systems. *IEEE Trans Autom Control* 48:2232–2238
- Faugeras O (1993) *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge
- Goebel RG, Sanfelice R, Teel AR (2012) *Hybrid dynamical systems*. Princeton University Press, Princeton
- Hartley R, Zisserman A (2000) *Multiple view geometry in computer vision*. Cambridge University Press, New York
- Hu G, Aiken D, Gupta S, Dixon W (2008) Lyapunov-based range identification for a paracatadioptric system. *IEEE Trans Autom Control* 53(7):1775–1781
- Liberzon D (2003) *Switching in systems and control*. Birkhauser, Boston

- Mehta S, Hu G, Gans N, Dixon WE (2010) Robot localization and map building. InTech, ch. A Daisy-chaining visual servoing approach with applications in tracking, localization, and mapping, pp 383–408. [Online]. Available: <http://ncr.mae.ufl.edu/papers/sid-chap.pdf>
- Parikh A, Cheng T-H, Chen H-Y, Dixon WE (2017) A switched systems framework for guaranteed convergence of image-based observers with intermittent measurements. *IEEE Trans Robot* 33(2):266–280
- Parikh A, Cheng T-H, Licitra R, Dixon WE (2018) A switched systems approach to image-based localization of targets that temporarily leave the camera field of view. *IEEE Trans Control Syst Technol* 26(6):2149–2156

Inventory Theory

Suresh P. Sethi

Jindal School of Management, The University of Texas at Dallas, Richardson, TX, USA

Abstract

This entry is a brief survey of classical inventory models and their extensions in several directions such as world-driven demands, presence of forecast updates, multi-delivery modes and advanced demand information, incomplete inventory information, and decentralized inventory control in the context of supply chain management. Important references are provided. We conclude with suggestions for future research.

Keywords

EOQ model · Newsvendor model · Base stock policy · (s, S) policy · Incomplete information

Introduction

Optimal inventory theory deals with managing stock levels of goods to effectively meet the demand of those goods. Because of the huge amount of capital that is tied up in inventory,

its management is critical to the profitability of firms. A systematic analysis of inventory problems began with the development of the classical economic order quantity (EOQ) formula of Ford W. Harris in 1913. A substantial amount of research was reported in 1958 by Kenneth J. Arrow, Samuel Karlin, and Herbert Scarf, and much more has accumulated since then. Books on the topic include Zipkin (2000), Porteus (2002), Axsäter (2006), and Bensoussan (2011).

In this entry, we review single- and multi-period models with deterministic, stochastic, partially observed demand for a single product. In these models, our aim is to decide on the time of the orders and the order quantities. The time between issuing an order and its receipt is called the lead time. For most of this review, we will assume the lead time to be zero, and the reader can consult the referenced books for nonzero lead time extensions and other topics not covered here.

Deterministic Demand

We will describe two classical models: the EOQ model and the dynamic lot size model.

The EOQ Model

This basic and most important deterministic model is concerned with a product that has a constant demand rate D in continuous time over an infinite horizon. No shortages are allowed. The costs consist of a fixed setup/ordering cost K and a holding cost h per unit of average on-hand stock per unit time. The production/purchase cost per unit time is a sunk cost since there is no choice of a total amount to produce, and hence it can be ignored. Although dynamic, the model can be reduced to a static model by a simple argument of periodicity. Moreover, it is obvious that one should never produce or order except for when the inventory level is zero, and one should order the same lot size Q each time the inventory level reaches zero. Since the average inventory level over time is $Q/2$ and the number of setups is D/Q per unit time, the long-run average cost to be minimized is $KD/Q + hQ/2$. The optimal policy that minimizes this cost, obtained using

the first-order condition, is to order the lot size

$$Q = \sqrt{\frac{2KD}{h}} \quad (1)$$

every time the inventory level reaches zero. Harris (1913) introduced the model. Erlenkotter (1990) provides a historical account of the formula, and Beyer and Sethi (1998) provide a mathematically rigorous proof involving quasi-variational inequalities (QVI) that arise in the course of dealing with continuous-time optimization problems involving fixed costs.

The Dynamic Lot Size Model

This is an analogue of the EOQ model when the demand varies over time. Wagner and Whitin (1958) developed it in the discrete-time finite horizon framework. With $D(t)$ denoting the demand in period t and other costs similar to those in the EOQ model, they showed that there exists an optimal policy in which an order will be issued just as the inventory level reaches zero, except for the first order. This policy is called the zero-inventory policy. With this in hand, the problem reduces to selecting only the order times. This is accomplished by applying a shortest path algorithm. Moreover, there are forward (recursion) procedures for solving the problem.

An important feature of this model is that in most cases, one can detect a *forecast horizon* which essentially separates earlier periods from later ones. More specifically, T is a forecast horizon if the first order in a T horizon problem remains optimal in any finite horizon problem with horizon longer than T , regardless of the demands beyond the period T . For an extensive bibliography of this literature, see Chand et al. (2002).

Stochastic Demand

We shall discuss three classical models and some of their extensions.

The Single-Period Problem: The Newsvendor Model

The problem of a newsvendor is to decide on an order quantity of newspapers to meet a stochastic demand at a minimum cost. If the realized demand is larger than the ordered quantity, it is lost and there is an opportunity loss of c_u (selling price minus purchase cost) for each paper short. On the other hand, for each paper ordered but not sold, there is an opportunity loss of c_o (purchase cost plus holding cost). The newsvendor conceptualizes the decision by each additional paper as a separate marginal contribution. The first is almost certain to be sold. Each additional paper is less likely to be sold than the previous one. Thus, each additional paper will be worth somewhat less, and the marginal paper at the optimum should be worth exactly zero. Thus, c_u times the probability of selling the marginal paper minus c_o times the probability of not selling it should equal zero. Now, if F denotes the cumulative probability distribution function of the demand D , then clearly the optimal order quantity Q satisfies $c_o \cdot F(Q) - c_u \cdot (1 - F(Q)) = 0$, which gives us the famous newsvendor formula for the optimal order quantity

$$Q = F^{-1}\left(\frac{c_u}{c_u + c_o}\right), \quad (2)$$

where $c_u/(c_u + c_o)$ is known as the critical fractile.

If p denotes the unit sale price, c the unit cost, and h the holding cost per unit per unit time, then $c_u = p - c$ and $c_o = c + h$, and therefore, the critical fractile can be expressed as $(p - c)/(p + h)$. An extension of the newsvendor formula to allow for a unit cost g of lost goodwill and a unit salvage value s received at the end of the period for each unit not sold is immediate. If we let $\alpha > 0$ denote the periodic discount factor, then $c_u = p + g - c$ and $c_o = c + h - \alpha s$ and the critical fractile becomes $(p + g - c)/(p + g + h - \alpha s)$, and therefore,

$$Q = F^{-1}\left(\frac{p + g - c}{p + g + h - \alpha s}\right). \quad (3)$$

The newsvendor model has been used extensively in the context of supply chain management with multiple agents maximizing their individual objectives. In this case, inefficiencies arise due to double marginalization. Then, a question of appropriate contracts that can lead to the first-best solution, or coordinate the supply chain, becomes important. Cachon (2003) surveys this literature.

Multi-period Inventory Models: No Fixed Cost

The newsvendor model is a single-period model, and its multi-period generalization requires that the inventory not sold in a period is carried over to the next period. This results in the multi-period inventory model with lost sales. It is assumed that demand in each period is independent and identically distributed (i.i.d.) with F denoting its cumulative probability distribution function. A rigorous analysis requires the method of dynamic programming, and it shows that there is a stock level S_t called *base stock* in period t , that we would ideally like to have at the beginning of period t . Thus, the optimal policy in period t , called *the base stock policy*, is to order

$$Q_t(x) = \begin{cases} S_t - x & \text{if } x < S_t, \\ 0 & \text{if } x \geq S_t. \end{cases} \quad (4)$$

In the special case when the terminal salvage value of an item is exactly equal to its cost c , it is possible to come up with the optimal policy using intuition. Since we do not need to salvage unused items in the multi-period setting, one can argue that an item carried over to the next period is worth its purchase cost c . Therefore, its presence means that the next period will need to order one less and thus save an amount c . In the last period, when there is no next period, our terminal salvage value assumption also guarantees a leftover item's worth to be also c . Thus, we can modify (3) and obtain a stationary base stock level

$$\begin{aligned} S &= F^{-1} \left(\frac{p + g - c}{(p + g - c) + (c + h - \alpha c)} \right) \\ &= F^{-1} \left(\frac{p + g - c}{p + g + h - \alpha c} \right) \end{aligned} \quad (5)$$

for each period t .

Thus, the elimination of the endgame effect delivers us a *myopic policy*, a policy optimal in the single-period case to be also optimal in the dynamic multi-period setting. A more general concept than the optimality of a myopic policy is that of the forecast horizon mentioned earlier in the context of the dynamic lot size model.

Sometimes, when the demand exceeds the on-hand inventory in the period, the demand is not lost but backlogged. In this case, each unit of backlogged demand is satisfied in the next period, and unit revenue p is recovered, but a unit backlogging cost b is incurred, due to expediting, special handling, delayed receipt of revenue, and loss of goodwill. Thus, $c_u = b - (1 - \alpha)c$, where the second term represents the savings due to postponing the purchase of the backlogged demand unit by one period, and $c_o = c + h - \alpha c$ as in (4). This gives us the base stock level

$$S = F^{-1} \left(\frac{b - (1 - \alpha)c}{b + h} \right), \quad (6)$$

which can be used in (5) to give the optimal policy.

Sometimes it is possible to have multiple delivery modes such as fast, regular, and slow as well as demand forecast updates. Then, at the beginning of each period, on-hand inventory and demand information are updated. At the same time, decisions on how much to order using each of the modes are made. Fast, regular, and slow orders are delivered at the ends of the current, the next, and one beyond the next periods, respectively. In such models, a modified base stock policy is optimal only for the two fastest modes. For details and further generalization, see Sethi et al. (2005).

An important extension includes serial inventory systems where stage 1 receives supplies from an outside source and each downstream stage receives supplies from its immediate upstream stage. Clark and Scarf (1960) introduced the notion of the echelon inventory position at a stage to consist of the stock at that stage plus stock in transit to that stage plus all downstream stock minus the amount backlogged at the final stage. Then, the optimal ordering policy at each stage

is given by an echelon base stock policy with respect to the echelon inventory position at that stage. It is known that assembly systems can be reduced to a serial system. Details can be found in Zipkin (2000).

Multi-period Inventory Models: Fixed Cost

When there is a fixed cost of ordering, it is clear that it would not be reasonable to follow the base stock policy when the inventory level is not much below the base stock level. Indeed, Scarf (1960) proved that there are numbers s_t and S_t , $s_t < S_t$, for period t such that the optimal policy in period t is to order

$$Q_t(x) = \begin{cases} S_t - x & \text{if } x \leq s_t \\ 0 & \text{if } x > s_t. \end{cases} \quad (7)$$

Such a policy is famously known as an (s, S) policy.

When the demands are not i.i.d., the model has been extended to Markovian demands. In this case, there is an exogenous Markov process, and the distribution of the demand in each period depends on the state of the Markov process, called the demand state, in that period. It can be shown that the optimal policy in period t is (s_t^i, S_t^i) , where i denotes the demand state in the period. Such a policy is also called a state-dependent (s, S) policy. Further details are available in Beyer et al. (2010). Recent advances in information technology have allowed managers to obtain advance demand information in addition to forecast updates. In such cases, a state-dependent (s, S) policy can be shown to be optimal. For details, refer to Ozer (2011).

The Continuous-Time Model: Fixed Cost

The marriage of the two classical results (1) and (7) is accomplished by Presman and Sethi (2006) in a continuous-time stochastic inventory model involving a demand that is the sum of a constant demand rate and a compound Poisson process. The optimal policies that minimize a discounted cost or the long-run average cost are both of (s, S) type. The (s, S) policy minimizing the long-run average cost reduces to the EOQ formula when the intensity of the com-

pound Poisson process is set to zero. And when the constant demand component vanishes, the model reduces to the continuous-review stochastic inventory model with fixed cost and compound Poisson demand.

Incomplete Inventory Information Models (i3)

A critical assumption in the vast inventory theory literature has been that the level of inventory at any given time is fully observed. The celebrated results (1) and (7) have been obtained under the assumption of full observation. Yet the inventory level is often not fully observed in practice, for a variety of reasons such as replenishment errors, employee theft, customer shoplifting, improper handling and damaging of merchandise, misplaced inventories, uncertain yield, imperfect inventory audits, and incorrect recording of sales. In such an environment of incomplete information, inventories are known to be partially observed and most of the well-known inventory policies including (1) and (7) are not even admissible, let alone optimal. In such cases, Bensoussan et al. (2010) show that the dynamic programming equation can be written in terms of the unnormalized conditional probability of the current inventory level given past observations, referred to as signals, instead of just the inventory level in the full observation case. Furthermore, one can write the evolution of the conditional probability in terms of its current value, the current order, and the current observation. However, there are no longer simple optimal policies except in cases of information delay reported in Bensoussan et al. (2009) and demands are observed to learn about an unknown parameter of the demand distribution Bensoussan et al. (2019) where modified base stock and (s, S) policies are shown to be optimal.

Summary and Future Directions

We briefly describe some classical results in inventory theory. These are based on full observation. Some recent work on inventory models under incomplete information is reported.

This work leads to a number of new research directions, both theoretical and empirical as reported in Sethi (2010). It would be of much interest to know the industries where the i3 problem is serious enough to warrant the difficult mathematical analysis required. Furthermore, how are the observed signals related to the inventory level? It is also clear from the reviewed literature that there are no simple optimal policies for most i3 problems, so it would be important to develop efficient computational procedures to obtain optimal solutions or to specify a class of simple implementable policies and optimize within this class. An important benefit of solving i3 problems optimally is the provision of an economic justification for technologies such as RFID that may reduce inaccuracies in inventory observations.

Another area of research would be to study multi-period multi-agent supply chains with a stochastic inventory dynamics. While these can be formulated as dynamic games, there are a number of equilibrium concepts to deal with, depending on the information the agents have. Some of them are time consistent or subgame perfect and some are not. Regardless, there are inefficiencies that arise from these decentralized game settings, and developing contracts for coordinating dynamic supply chains remains a wide open topic of research.

Cross-References

- [Nonlinear Filters](#)
- [Stochastic Dynamic Programming](#)

Bibliography

- Arrow KJ, Karlin S, Scarf H (1958) Studies in the mathematical theory of inventory and production. Stanford University Press, Stanford
- Axsäter S (2006) Inventory control. Springer, New York
- Bensoussan A (2011) Dynamic programming and inventory control. IOS Press, Washington, DC
- Bensoussan A, Çakanyildirim M, Sethi SP, Wang M, Zhang H (2011) Average cost optimality in inventory models with dynamic information delays. *IEEE Transactions on Automatic Control* 56(12): 2869–2882.
- Bensoussan A, Sethi SP, Thiam A, Turi J (2020) Inventory control driven by demand data: Optimality and computation of base stocks. Working paper, University of Texas at Dallas. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3730630
- Bensoussan A, Çakanyildirim M, Feng Q, Sethi SP (2009) Optimal ordering policies for stochastic inventory problems with observed information delays. *Prod Oper Manag* 18(5):546–559
- Bensoussan A, Çakanyildirim M, Sethi SP (2010) Filtering for discrete-time Markov processes and applications to inventory control with incomplete information. In: Crisan D, Rozovsky B (eds) *Handbook on nonlinear filtering*. Oxford University Press, Oxford, UK, pp 500–525
- Beyer D, Cheng F, Sethi SP, Taksar MI (2010) Markovian demand inventory models. *International series in operations research and management science*. Springer, New York
- Beyer D, Sethi SP (1998) A proof of the EOQ formula using quasi-variational inequalities. *Int J Syst Sci* 29(11):1295–1299
- Cachon GP (2003) Supply chain coordination with contracts. In: de Kok AG, Graves S (eds) *Supply chain management-handbook in OR/MS*, chapter 6. North-Holland, Amsterdam, pp 229–340
- Chand S, Hsu VN, Sethi SP (2002) Forecast, solution and rolling horizons in operations management problems: a classified bibliography. *Manuf Serv Oper Manag* 4(1):25–43
- Clark AJ, Scarf H (1960) Optimal policies for a multi-echelon inventory problem. *Manag Sci* 6(4):475–490
- Erlenkotter D (1990) Ford Whitman Harris and the economic order quantity model. *Oper Res* 38(6):937–946
- Harris FW (1913) How many parts to make at once. *Fact Mag Manag* 10(2):135–136, 152. Reprinted (1990) *Oper Res* 38(6):947–950
- Ozer O (2011) Inventory management: information, coordination, and rationality. In: Kempf KG, Keskinocak P, Uzsoy R (eds) *Planning production and inventories in the extended enterprise*. Springer, New York
- Porteus EL (2002) *Stochastic inventory theory*. Stanford University Press, Stanford, CA
- Presman E, Sethi SP (2006) Inventory models with continuous and Poisson demands and discounted and average costs. *Prod Oper Manag* 15(2):279–293
- Scarf H (1960) The optimality of (s, S) policies in dynamic inventory problems. In: Arrow KJ, Karlin S, Suppes P (eds) *Mathematical methods in the social sciences*. Stanford University Press, Stanford, pp 196–202
- Sethi SP (2010) i3: incomplete information inventory models. *Decis Line* Oct:16–19
- Sethi SP, Yan H, Zhang H (2005) *Inventory and supply chain management with forecast updates*. Springer, New York
- Wagner HM, Whitin TM (1958) Dynamic version of the economic lot size model. *Manag Sci* 5:89–96
- Zipkin P (2000) *Foundations of inventory management*. McGraw Hill, New York

Investment-Consumption Modeling

L. C. G. Rogers

University of Cambridge, Cambridge, UK

Abstract

The simplest investment-consumption problem is the celebrated example of Robert Merton (J Econ Theory 3(4):373–413, 1971). This survey shows three different ways of solving the problem, each of which is a valuable solution method for more complicated versions of the question.

Keywords

Budget constraint ·
Hamilton-Jacobi-Bellman (HJB) equation ·
Merton problem · Value function

Introduction

Consider an investor in a market with a riskless bank account accruing continuously compounded interest at rate r_t , and with a single risky asset whose price S_t at time t evolves as

$$dS_t = S_t(\sigma_t dW_t + \mu_t dt), \quad (1)$$

where W is a standard Brownian motion, and σ and μ are processes previsible with respect to the filtration of W . The investor starts with initial wealth w_0 and chooses the rate c_t of consuming, and the wealth θ_t to invest in the risky asset, so that his overall wealth evolves as

$$\begin{aligned} dw_t &= \theta_t(\sigma_t dW_t + \mu_t dt) + r_t(w_t - \theta_t)dt - c_t dt \\ &= r_t w_t dt + \theta_t\{\sigma_t dW_t + (\mu_t - r_t)dt\} - c_t dt. \end{aligned} \quad (2)$$

(3)

For convenience, we assume that σ , σ^{-1} , and μ are bounded. See Rogers and Williams

(2000a, b) for background information on stochastic processes. The three terms in (2) have natural interpretations: The first expresses the evolution of the wealth invested in the stock, the second the interest accruing on the wealth $(w - \theta)$ invested in the bank account, and the third is the cash being withdrawn for consumption.

To avoid so-called doubling strategies, we insist that the wealth process so generated by the controls (c, θ) must remain bounded below in some suitable way, which here is just the condition $w_t \geq 0$ for all $t \geq 0$; any (c, θ) satisfying this condition will be called *admissible*. The set of admissible (c, θ) will be denoted $\mathcal{A}(w_0)$, a notation which makes explicit the dependence on the investor's initial wealth.

The investor's objective is taken to be to obtain

$$V(w_0) \equiv \sup_{(c, \theta) \in \mathcal{A}(w_0)} E \left[\int_0^\infty e^{-\rho t} U(c_t) dt \right] \quad (4)$$

for some constant $\rho > 0$. The problem cannot be solved explicitly at this level of generality, but if we take some special cases, we are able to illustrate the main methods used to attack it. Many other objectives with various different constraints can be handled by similar techniques: see Rogers (2013) for a wide range of examples.

The Main Techniques

We present here three important techniques for solving such problems: the value function approach; the use of dual variables; and the use of martingale representation. The first two methods only work if the problem is Markovian; the third only works if the market is complete. There is a further method, the Pontryagin-Lagrange approach; see Sect. 1.5 in Rogers (2013). While this is a quite general approach, we can only expect explicit solutions when further structure is available.

The Value Function Approach

To illustrate this, we focus on the original Merton problem (Merton 1971), where σ and μ are both constant, and the utility U is constant relative risk

aversion (CRRA):

$$U'(x) = x^{-R} \quad (x > 0) \quad (5)$$

for some $R > 0$ different from 1. The case $R = 1$ corresponds to logarithmic utility, and can be solved by similar methods. Perhaps the best starting point is the *Davis-Varaiya Martingale Principle of Optimal Control (MPOC)*: The process $Y_t = e^{-\rho t} V(w_t) + \int_0^t e^{-\rho s} U(c_s) ds$ is a supermartingale under any control, and a martingale under optimal control. If we use Itô's formula, we find that

$$\begin{aligned} e^{\rho t} dY_t &= -\rho V(w_t)dt + V'(w_t)dw_t \\ &\quad + \frac{1}{2}\sigma^2\theta_t^2 V''(w_t)dt + U(c_t)dt \\ &\doteq [-\rho V + \{\theta_t(\mu - r) - c_t + r\}V' \\ &\quad + \frac{1}{2}\sigma^2\theta_t^2 V'' + U(c_t)] dt, \end{aligned} \quad (6)$$

where the symbol $\doteq$ denotes that the two sides differ by a (local) martingale. If the MPOC is to hold, then we expect that the drift in dY should be non-positive under any control, and equal to zero under optimal control. We simply assume for now that local martingales are martingales; this is of course not true in general, and is a point that needs to be handled carefully in a rigorous proof. Directly from (6), we then deduce the *Hamilton-Jacobi-Bellman (HJB) equations* for this problem:

$$\begin{aligned} 0 = \sup_{c, \theta} [-\rho V + \{\theta(\mu - r) - c + r\}V' \\ + \frac{1}{2}\sigma^2\theta^2 V'' + U(c)]. \end{aligned} \quad (7)$$

Write $\tilde{U}(y) \equiv \sup \{U(x) - xy\}$ for the convex dual of U , which in this case has the explicit form

$$\tilde{U}(y) = -\frac{y^{1-R'}}{1-R'} \quad (8)$$

with $R' \equiv 1/R$. We are then able to perform the optimizations in (7) quite explicitly to obtain

$$0 = -\rho V + rV' + \tilde{U}(V') - \frac{1}{2} \kappa^2 \frac{(V')^2}{V''} \quad (9)$$

where

$$\kappa \equiv \frac{\mu - r}{\sigma}. \quad (10)$$

Nonlinear PDEs arising from stochastic optimal control problems are not in general easy to solve, but (9) is tractable in this special setting, because the assumed CRRA form of U allows us to deduce by a scaling argument that $V(w) \propto w^{1-R} \propto U(w)$, and we find that

$$V(w) = \gamma_M^{-R} U(w), \quad (11)$$

where

$$R\gamma_M = \rho + (R-1)(r + \frac{1}{2}\kappa^2/R). \quad (12)$$

The optimal investment and consumption behavior is easily deduced from the optimal choices which took us from (7) to (9). After some calculations, we discover that

$$\theta_t^* = \pi_M w_t \equiv \frac{\mu - r}{\sigma^2 R} w_t, \quad c_t^* = \gamma_M w_t \quad (13)$$

specifies the optimal investment/consumption behavior in this example. (The positivity of γ_M is necessary and sufficient for the problem to be well posed; see Sect. 1.6 in Rogers 2013). Unsurprisingly, the optimal solution scales linearly with wealth.

Dual Variables

We illustrate the use of dual variables in the constant-coefficient case of the previous section, except that we no longer suppose the special form (5) for U . The analysis runs as before all the way to (9), but now the convex dual $\tilde{U}$ is not simply given by (8). Although it is not now possible to guess and verify, there is a simple transformation which reduces the nonlinear ODE (9) to something we can easily handle. We introduce the new variable $z > 0$ related to w by $z = V'(w)$, and define a function J by

$$J(z) = V(w) - wz. \quad (14)$$

Simple calculus gives us $J' = -w$, $J'' = -1/V''$, so that the HJB equation (9) transforms

into

$$0 = \tilde{U}(z) - \rho J(z) + (\rho - r)zJ'(z) + \frac{1}{2}\kappa^2 z^2 J''(z), \quad (15)$$

which is now a second-order *linear* ODE, which can be solved by traditional methods; see Sect. 1.3 of Rogers (2013) for more details.

Use of Martingale Representation

This time, we shall suppose that the coefficients μ_t , r_t , and σ_t in the wealth evolution (3) are general previsible processes; to keep things simpler, we shall suppose that μ , r , and σ^{-1} are all bounded previsible processes. The Markovian nature of the problem which allowed us to find the HJB equation in the first two cases is now destroyed, and a completely different method is needed. The way in is to define a positive semimartingale ζ by

$$d\zeta_t = \zeta_t (-r_t dt - \kappa_t dW_t), \quad \zeta_0 = 1 \quad (16)$$

where $\kappa_t = (\mu_t - r_t)/\sigma_t$ is a previsible process, bounded by hypothesis. This process, called the *state-price density process*, or the *pricing kernel*, has the property that if w evolves as (3), then $M_t \equiv \zeta_t w_t + \int_0^t \zeta_s c_s ds$ is a positive local martingale.

Since positive local martingales are supermartingales, we deduce from this that

$$M_0 = w_0 \geq E \left[\int_0^\infty \zeta_s c_s ds \right]. \quad (17)$$

Thus, for any $(c, \theta) \in \mathcal{A}(w_0)$, the budget constraint (17) must hold. So the solution method here is to maximize the objective (4) subject to the constraint (17). Absorbing the constraint with a Lagrange multiplier λ , we find the unconstrained optimization problem

$$\sup E \left[\int_0^\infty \{e^{-\rho s} U(c_s) - \lambda \zeta_s c_s\} ds \right] + \lambda w_0 \quad (18)$$

whose optimal solution is given by

$$e^{-\rho s} U'(c_s) = \lambda \zeta_s, \quad (19)$$

and this determines the optimal c , up to knowledge of the Lagrange multiplier λ , whose value is fixed by matching the budget constraint (17) with equality.

Of course, the missing logical piece of this argument is that if we are given some $c \geq 0$ satisfying the budget constraint, is there necessarily some θ such that the pair (c, θ) is admissible for initial wealth w_0 ? In this setting, this can be shown to follow from the Brownian integral representation theorem, since we are in a complete market; however, in a multidimensional setting, this can fail, and then the problem is effectively insoluble.

Summary and Future Directions

This brief survey states some of the main ideas of consumption-investment optimization, and sketches some of the methods in common use. Explicit solutions are rare, and much of the interest of the subject focuses on efficient numerical schemes, particularly when the dimension of the problem is large. A further area of interest is in continuous-time principal-agent problems; Cvitanic and Zhang (2012) is a recent account of some of the methods of this subject, but it has to be said that the theory of such problems is much less complete than the simple single-agent optimization problems discussed here.

Cross-References

- [Risk-Sensitive Stochastic Control](#)
- [Stochastic Dynamic Programming](#)
- [Stochastic Linear-Quadratic Control](#)
- [Stochastic Maximum Principle](#)

Bibliography

- Cvitanic J, Zhang J (2012) Contract theory in continuous-time models. Springer, Berlin/Heidelberg
- Merton RC (1971) Optimum consumption and portfolio rules in a continuous-time model. J Econ Theory 3(4):373–413

- Rogers LCG (2013) Optimal investment. Springer, Berlin/New York
- Rogers LCG, Williams D (2000a) Diffusions, Markov processes and martingales, vol 1. Cambridge University Press, Cambridge/New York
- Rogers LCG, Williams D (2000b) Diffusions, Markov processes and martingales, vol 2. Cambridge University Press, Cambridge/New York

Iterative Learning Control

David H. Owens

Zhengzhou University, Zhengzhou, P.R. China
Automatic Control and Systems Engineering,
University of Sheffield, Sheffield, UK

Abstract

Iterative learning control addresses tracking control where the repetition of a task allows improved tracking accuracy from task to task. The area inherits the analysis and design issues of classical control but adds convergence conditions for task to task learning, the need for acceptable task-to-task performance and the implications of modelling errors for task-to-task robustness.

Keywords

Iterative learning control · Iteration · Adaptation · Repetition · Optimization · Robustness

Introduction

Iterative learning control (ILC) is relevant to trajectory tracking control problems on a finite interval $[0, T]$ (Bien and Xu 1998; Chen and Wen 1999; Ahn et al. 2007b; Owens 2016). It has close links to multipass process theory (Edwards and Owens 1982) and repetitive control (Rogers et al. 2007) plus conceptual links to adaptive control. It focusses on problems where the repetition of a specified task creates the possibility of improving tracking accuracy from task to task

and, in principle, reducing the tracking error to exactly zero. The iterative nature of the control schemes proposed the use of past executions of the control to update/improve control action, and the asymptotic learning of the required control signals puts the topic in the area of adaptive control although other areas of study are reflected in its methodologies.

Application areas are many and include robotic assembly (Arimoto et al. 1984), electromechanical test systems (Daley et al. 2007) and medical rehabilitation robotics (Rogers et al. 2010). For example, consider a manufacturing robot required to undertake an indefinite number of identical tasks (such as “pick and place” of components) specified by a spatial or temporal trajectory on a defined time interval. The problem is two-dimensional. More precisely, the controlled system evolves with *two* variables, namely, time $t \in [0, T]$ (elapsed in each iteration) and iteration index $k \geq 0$. Data consists of signals $f_k(t)$ giving the value of the signal f at time t on iteration k . The conceptual algorithm used is (Owens 2016):

Step one: (Preconditioning) Implement loop controllers to condition plant dynamics.

Step two: (Initialization) Given a demand signal $r(t)$, $t \in [0, T]$, choose an initial input $u_0(t)$, $t \in [0, T]$ and set $k = 0$.

Step three: (Response measurement) Return the plant to a defined initial state. Find the output response y_k to the input u_k . Construct the tracking error $e_k = r - y_k$. Store data.

Step four: (Input signal update) Use past records of inputs used and tracking errors generated to construct a new input $u_{k+1}(t)$, $t \in [0, T]$ to be used to improve tracking accuracy on the next trial.

Step five: (Termination/task repetition?) Either terminate the sequence or increase k by unity and return to step 3.

It is the updating of the input signal based on observation that provides the conceptual link to adaptive control. ILC *causality* defines *past data* at time t on iteration k as data on the interval $[0, t]$ on that iteration plus all data on $[0, T]$ on all previous iterations. Feedback plus

feedforward control is the name given to a control policy that uses real-time feedback of current iteration measurements and feedforward transfer of information from past iterations to the current iteration.

Modelling Issues

Design approaches have been model-based. Most non-linear problems assume non-linear-state space models relating the $\ell \times 1$ input vector $u(t)$ to the $m \times 1$ output vector $y(t)$ via an $n \times 1$ state vector $x(t)$ as follows:

$$\dot{x}(t) = f(x(t), u(t)), \quad y(t) = h(x(t), u(t)),$$

where $t \in [0, T]$, $x(0) = x_0$ and f and h are vector-valued functions. The discrete time (sample data) version replaces derivatives by a forward shift, where t is now a sample counter, $0 \leq t \leq N$. The continuous time linear model is:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

with an analogous model for discrete systems. In both case the matrices A, B, C, D are constant or time varying matrices of appropriate dimension.

Non-linear systems present the greatest technical challenge. Linear system's challenges are greater for the time varying, continuous time case. The simplest linear case of discrete time, time-invariant systems, can be described by a matrix relationship:

$$y = Gu + d \quad (1)$$

where y denotes the $m(N+1) \times 1$ "supervector" generated by the time series $y(0), y(1), \dots, y(N)$ and the construction $y = [y^T(0), y^T(1), \dots, y^T(N)]^T$, the supervector u is generated, similarly, by the time series $u(0), u(1), \dots, u(N)$, and d is generated by the times series $Cx_0, CAx_0, \dots, CA^N x_0$. The matrix G has the lower block triangular structure:

$$G = \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \vdots & & & & \\ CA^{N-1}B & CA^{N-2}B & \dots & D \end{bmatrix}$$

defined in terms of the Markov parameter matrices $D, CB, CAB, \dots$ of the plant. This structure has led to a focus on the discrete time, time invariant case and exploitation of matrix algebra techniques.

More generally, $G : \mathcal{U} \rightarrow \mathcal{Y}$ can be a bounded linear operator between suitable signal spaces $\mathcal{U}$ and $\mathcal{Y}$. Taking G as a convolution operator, the representation (1) also applies to time-varying continuous time and discrete time systems. The representation also applies to differential-delay systems, coupled algebraic and differential systems, multi-rate systems and other situations of interest.

Formal Design Objectives

Problem statement: Given a reference signal $r \in \mathcal{Y}$ and an initial input signal $u_0 \in \mathcal{U}$, construct a causal control update rule/algorithm:

$$u_{k+1} = \psi_k(e_{k+1}, e_k, \dots, e_0, u_k, u_{k-1}, \dots, u_0)$$

that ensures that $\lim_{k \rightarrow \infty} e_k = 0$ (convergence) in the norm topology of $\mathcal{Y}$.

The update rule $\psi_k(\cdot)$ represents the simple idea of expressing u_{k+1} in terms of past data. A general linear "high order" rule is:

$$u_{k+1} = \sum_{j=0}^k W_j u_{k-j} + \sum_{j=0}^{k+1} K_j e_{k+1-j} \quad (2)$$

with bounded linear operators $W_j : \mathcal{U} \rightarrow \mathcal{U}$ and $K_j : \mathcal{Y} \rightarrow \mathcal{U}$, regarded as compensation elements and/or filters to condition the signals. Typically $K_j = 0$ (resp. $W_j = 0$) for $j > M_e$ (resp. $j > M_u$). A simple structure is:

$$u_{k+1} = W_0 u_k + K_0 e_{k+1} + K_1 e_k \quad (3)$$

Assuming that G and W_0 commute (i.e. $GW_0 = W_0G$), the resultant error evolution takes the form:

$$e_{k+1} = (I + GK_0)^{-1}(W_0 - GK_1)e_k \\ + (I + GK_0)^{-1}(I - W_0)(r - d)$$

which relates errors on each iteration via the recursive structure:

$$e_{k+1} = Le_k + d_0. \quad (4)$$

Equations of this type (and their non-linear counterparts) play a central role throughout the subject area.

Robust ILC: An ILC algorithm is said to be robust if convergence is retained in the presence of a defined class of modelling errors.

Results from multipass systems theory (Edwards and Owens 1982) indicate robust convergence of the sequence $\{e_k\}_{k \geq 0}$ to a limit $e_\infty \in \mathcal{Y}$ (in the presence of small modelling errors) if the spectral radius condition:

$$r(L) = r[(I + GK_0)^{-1}(W_0 - GK_1)] < 1 \quad (5)$$

is satisfied where $r[\cdot]$ denotes the spectral radius of its argument. However, the desired condition $e_\infty = 0$ is true only if $W_0 = I$ (when $d_0 = 0$). For a given r , it may be possible to retain the benefits of choosing $d_0 \neq 0$ (i.e. $W_0 \neq I$) and still ensure that e_∞ is sufficiently small for the application in mind, e.g. by limiting limit errors to a high frequency band. This and other spectral radius conditions form the underlying convergence condition when choosing controller elements but are rarely computed. The simplest algorithm using eigenvalue computation for a linear discrete time system defines the *relative degree* to be $k^* = 0$ if $D \neq 0$ and the smallest integer k such that $CA^{k-1}B \neq 0$ otherwise. Replacing $\mathcal{Y}$ by the range of G , choosing $W_0 = I$, $K_0 = 0$ and $K_1 = I$ and supposing that $k^* \geq 1$, the Arimoto input update rule $u_{k+1}(t) = u_k(t) + e_k(t + k^*)$, $0 \leq t \leq N + 1 - k^*$ provides robust convergence if, and only if, $r[I - CA^{k^*-1}B] < 1$. It does not imply that the error signal necessarily improves

each iteration. Errors can reach very high values before finally converging to zero. However, if (5) is replaced by the operator norm condition:

$$\|(I + GK_0)^{-1}(W_0 - GK_1)\| < 1, \quad \text{then} \quad (6)$$

$\{\|e_k - e_\infty\|_Q\}_{k \geq 0}$ monotonically decreases to zero.

The spectral radius condition throws light on the nature of ILC robustness. Choosing, for simplicity, $K_0 = 0$ and $W_0 = I$, the requirement that $r[I - GK_1] < 1$ will be satisfied by a wide range of processes G , namely, those for which the eigenvalues of $I - GK_1$ lie in the open unit circle of the complex plane. Translating this requirement into useful robustness tests may not be easy in general. The discussion does however show that the behaviour of GK_1 must be “sign definite” to some extent, as if $r[I - GK_1] < 1$, then $r[I - (-G)K_1] > 1$, i.e., replacing the plant by $-G$ (no matter how small) will inevitably produce non-convergent behaviour. A more detailed characterization of this property is possible for inverse model ILC (Owens 2016).

Inverse Model-Based Iteration

If $m = \ell$ and a linear system G has a well-defined inverse model G^{-1} , then the required input signal to produce zero tracking error is $u_\infty = G^{-1}(r - d)$. The simple update rule:

$$u_{k+1} = u_k + \beta G^{-1}e_k, \quad (7)$$

where β is a *learning gain*, produces the dynamics:

$$e_{k+1} = (1 - \beta)e_k \quad \text{or} \quad e_{k+1} = (1 - \beta)^k e_0,$$

proving that zero error is attainable with added flexibility in convergence rate control by choosing $\beta \in (0, 2)$. The convergence then takes the form of a *monotonic* reduction in error magnitude. Errors in the system model used in (7) are an issue. Insight into this problem is easily obtained for single-input, single-output discrete time systems with multiplicative plant uncertainty U as

retention of monotonic convergence is ensured (Owens 2016; Owens and Chu 2012) by a frequency domain condition:

$$|\frac{1}{\beta} - U(e^{i\theta})| < \frac{1}{\beta}, \text{ for all } \theta \in [0, 2\pi] \quad (8)$$

that illustrates a number of general empirical rules for ILC robust design. The first is that a small learning gain (and hence small input update changes and slow convergence) will tend to increase robustness and, hence, that it is *necessary* that multiplicative uncertainties satisfy some form of strict positive real condition which, for (7), is:

$$\operatorname{Re} [U(e^{i\theta})] > 0, \text{ for all } \theta \in [0, 2\pi], \quad (9)$$

a condition that limits high-frequency roll-off error and constrains phase errors to the range $(-\frac{\pi}{2}, \frac{\pi}{2})$. The second observation is that, if G is non-minimum-phase, the inverse G^{-1} is unstable, a situation that cannot be tolerated in practice.

The case of multi-input, multi-output (MIMO) systems with a left and/or a right inverse is analysed in detail in Owens (2016) with monotonic robustness being described by an operator inequality that, for state space systems, becomes a frequency-dependent matrix inequality.

Optimization-Based Iteration

Design criteria can be strengthened by a monotonicity requirement. Measuring error magnitude by a norm $\|e\|_Q$ on $\mathcal{Y}$, such as the weighted mean square error (with Q symmetric, positive definite):

$$\|e\|_Q = \sqrt{\int_0^T e^T(t) Q e(t) dt},$$

then the monotonicity condition $\|e_{k+1}\|_Q < \|e_k\|_Q$ for all $k \geq 0$ provides a performance improvement from iteration to iteration. This idea

leads to a number of design approaches (Owens and Daley 2008) and, more extensively, in the texts (Ahn et al. 2007b; Owens 2016) (which also characterize aspects of robustness).

Function/Time Series Optimization

Control algorithms that reduce the error norm monotonically are, implicitly, searching for the minimum achievable error e_∞ satisfying:

$$\|e_\infty\| = \min\{\|e\| : e = r - y\}$$

subject to the dynamical constraint $y = Gu + d$. That is, iterative learning control has a natural link to optimization theory and algorithms.

The idea of gradient-based control (Owens 2016) takes the form, for linear models (1):

$$u_{k+1} = u_k + \beta G^* e_k$$

where $G^* : \mathcal{Y} \rightarrow \mathcal{U}$ is the adjoint operator of G and β is a scalar gain (sometimes called the step size). The error evolution takes the form:

$$e_{k+1} = (I - \beta G G^*) e_k.$$

As a consequence, monotonic reductions are achieved if β lies in the range:

$$0 < \beta < \frac{2}{\|G^*\|^2}.$$

For state space systems, the norm $\|G^*\|$ can be computed from frequency response data, and robust, monotonic convergence is achieved if a frequency-dependent matrix inequality is satisfied.

Norm optimal ILC (NOILC) (Owens 2016; Owens and Daley 2008) guarantees monotonicity and convergence to $e_\infty = 0$ by computing u_{k+1} to minimize an objective function:

$$J(u) = \|e\|_Q^2 + \|u - u_k\|_R^2,$$

subject to plant dynamics. The additional input-dependent term is there to influence the magnitude of the change in control of each iteration.

For linear models (1):

$$u_{k+1} = u_k + G^* e_{k+1}.$$

For continuous or discrete time linear-state space models, the problem is a classical optimal tracking problem with a solution consisting of online, current iteration-state feedback and a feedforward term generated offline by simulation of an “adjoint” model using the previous iteration data e_k and u_k . Reducing R in J leads to faster convergence rates but higher gains. The presence of non-minimum-phase zeros also has a negative effect on convergence (Owens and Chu 2010; Owens 2016). Monotonicity and convergence to zero are retained, but, after an initial fall, the error norm then reduces infinitesimally each iteration producing the practical effect of limited error reductions over finite iteration horizons. Rules exist, Owens and Chu (2010) and Owens (2016), to minimize the effect by choice of initial input u_0 and the reference signal r .

Related Linear NOILC Problems

If $\mathcal{Y}$ and $\mathcal{U}$ are real Hilbert spaces, geometrical arguments can be used (Owens 2016) to generate algorithms extending the NOILC algorithm to include *acceleration mechanisms, predictive control and the inclusion of input signal constraints*. They also allow more flexibility in the form of task specification.

In the *intermediate point NOILC problem* (denoted IPNOILC), the task requirement is that the output signal $y(t)$, $0 \leq t \leq T$ takes specified values $r(t_1), r(t_2), \dots, r(t_M)$ as it passes through the M intermediate points $0 < t_1 < t_2 < \dots < t_M$. The precise nature of the trajectory between points is of secondary importance. Again, the solution, for linear-state space systems, can be constructed from Riccati equation-based feedback rules combined with “jump” conditions and feedforward control signals computed offline. The IPNOILC solution is non-unique, and the remaining degrees of freedom can be used to satisfy other design objectives. Switching algorithms (Owens et al.

2013) converge to a solution of the problem whilst simultaneously minimizing an auxiliary criterion:

$$J_{\text{aux}}(u) = \|z - z_0\|_Q^2 + \|u - u_0\|_R^2.$$

Auxiliary optimization is a tool for shaping the solution of the IPNOILC problem. The auxiliary variable z could be internal states whose behaviour is important to plant operation or simply defined by the output, e.g. $z = \ddot{y}$, which, if small, might reduce input “forces” and hence actuator activity.

Each algorithm has the effect of producing a different form of the evolution operator L . This operator and its spectrum describe the properties of the algorithm. Other algorithms (Owens 2016) such as the “notch” algorithm annihilate parts of the spectrum in one iteration, opening up the possibility of sequentially, selectively and rapidly reducing errors in defined frequency ranges.

Parameter Optimization

NOILC can be simplified by reducing the degrees of freedom defining control action to a small number of control law parameters. For a discrete system (1), a general update rule is:

$$u_{k+1} = u_k + \Gamma(\beta_{k+1})e_k, \quad k \geq 0.$$

Here the matrix $\Gamma(\beta)$ is linear in the $p \times 1$ parameter vector β with $\Gamma(0) = 0$. Under these conditions $\Gamma(\beta)e = F(e)\beta$ where the matrix $F(e)$ is linear in e with $F(0) = 0$. Examples of useful parameterizations include inverse model control (Owens et al. 2012).

Monotonicity of the error norm is ensured by choosing the parameter vector β_{k+1} to minimize:

$$J(\beta) = \|e\|_Q^2 + \beta^T W_{k+1} \beta$$

subject to the dynamic constraint (1). Each $p \times p$ weighting matrix W_{k+1} is symmetric and positive definite and may be iteration dependent. The algorithm creates a non-linear ILC law providing a link between parameter evolution, past errors and the choice of weight W_{k+1} .

Summary and Future Directions

The basic structure of ILC is now well-understood with a number of algorithms available with known convergence properties and empirical links between parameter choice and convergence rates (Bristow et al. 2006; Ahn et al. 2007a; Wang et al. 2009; Xu 2011; Owens and Daley 2008; Owens 2016). Optimization-based algorithms provide a structured approach to convergence and have a familiar quadratic optimal control structure. Despite the practical benefits of monotonic error norms, this approach underlines the difficulties induced by non-minimum-phase (NMP) properties of the plant. Operator representations extend this theory to more general problems such as the intermediate point tracking problem and, where solutions are non-unique, can be converted into iterative algorithms that inherit the properties of NOILC but converge to a solution that also minimizes an auxiliary optimization criterion.

Many of the challenges addressed by NOILC are inherited by other algorithms, many of which mimic established control design paradigms. For example, the commonly used *PD update law*:

$$u_{k+1}(t) = u_k(t) + K_1 e_k(t) + K_2 \dot{e}_k(t)$$

can produce convergence by suitable choice of K_1 and K_2 . Proofs of convergence are typically based on spectral radius conditions similar to (5) for linear systems or on techniques such as contraction mapping (fixed point) theorems (Xu 2011) for non-linear systems. The non-linear case generally suggests *local* convergence conditions dependent on growth conditions on the non-linearity. They typically cannot be checked in practice but do link convergence to simple, empirical, gain selection rules.

ILC, as a topic, is a large area of study. Literature surveys indicate an increasing number of applications and also that progress has been made in a number of design areas including adaptive ILC, the use of intelligent control ideas, 2D systems theory and mathematical studies of fractional order control laws (Chen et al. 2013). The further development of ILC from its current

strong base will draw extensively from classical control knowledge but relies on the three aspects of *plant modelling*, *control design* and *coping with uncertainty*. Issues central to success include:

1. Extending current ILC knowledge to other classes of model needed for applications.
2. Integration of online data-based modelling into ILC schemes to enhance adaptive control options.
3. Ensuring the property of error monotonicity or characterizing any expected non-monotonicity.
4. The construction of robustness tests and using the ideas in new robust design methodologies.
5. Providing a better understanding of the effect of noise and disturbances on algorithm performance.
6. Extending the range of tasks to include, for example, different challenges for different outputs on different subintervals of $[0, T]$ (Owens 2016).
7. Creating design tools for non-linear plant that ensure convergence and a degree of robustness but, in particular, provide some control of internal plant states that may be subject to dynamical constraints.

Cross-References

- [Adaptive Control: Overview](#)
- [Robust Fault Diagnosis and Control](#)

Bibliography

- Ahn H-S, Chen YQ, Moore KL (2007a) Iterative learning control: brief survey and categorization. *IEEE Trans Syst Man Cybern Part C Appl Rev* 37(3):1099–1121
- Ahn H-S, Moore KL, Chen YQ (2007b) Iterative learning control: robustness and monotonic convergence for interval systems. *Series on communications and control engineering*. Springer, London
- Arimoto S, Miyazaki F, Kawamura S (1984) Bettering operation of robots by learning. *J Robot Syst* 1(2): 123–140
- Bien Z, Xu J-X (1998) Iterative learning control: analysis, design, integration and applications. Kluwer Academic Publishers, Boston/Dordrecht/London

- Bristow DA, Tharayil M, Alleyne AG (2006) A survey of iterative learning control a learning-based method for high-performance tracking control. *IEEE Control Syst Mag* 26(3):96–114
- Chen Y, Wen C (1999) Iterative learning control: convergence, robustness and applications, vol 48. Lecture notes in control and information sciences. Springer, London
- Chen YQ, Li Y, Ahn HS, Tian G (2013) A survey on fractional order iterative learning. *J Optim Theory Appl* 156:127–140
- Daley S, Owens DH, Hatonen J (2007) Application of optimal iterative learning control to the dynamic testing of mechanical structures. *Proc IMechE Part I J Syst Control Eng* 221:211–222
- Edwards JB, Owens DH (1982) Analysis and control of multipass processes. Research Studies Press, Wiley, Chichester
- Owens DH (2016) Iterative learning control: an optimization paradigm. *Advances in industrial control*. Springer, London
- Owens DH, Chu B (2010) Modelling of non-minimum phase effects in discrete-time norm optimal iterative learning control. *Int J Control* 83(10):2012–2027
- Owens DH, Chu B (2012) Combined inverse and gradient iterative learning control: performance, monotonicity, robustness and non-minimum-phase zeros. *Int J Robust Nonlinear Control*. (2):203–226. <https://doi.org/10.1002/rnc.2893>
- Owens DH, Daley S (2008) Iterative learning control – monotonicity and optimization. *Int J Appl Math Comput Sci* 18(3):279–293
- Owens DH, Chu B, Songjun M (2012) Parameter-optimal iterative learning control using polynomial representations of the inverse plant. *Int J Control* 85(5):533–544
- Owens DH, Freeman CT, Chu B (2013) Multivariable norm optimal iterative learning control with auxiliary optimization. *Int J Control* 1(1):1–2
- Rogers E, Galkowski K, Owens DH (2007) Control systems theory and applications for linear repetitive processes, vol 349. Lecture notes in control and information sciences. Springer, Berlin
- Rogers E, Owens DH, Werner H, Freeman CT, Lewin PL, Schmidt C, Kirchhoff S, Lichtenberg G (2010) Norm-optimal iterative learning control with application to problems in acceleration-based free electron lasers and rehabilitation robotics. *Eur J Control* 5:496–521
- Wang Y, Gao F, Doyle III FJ (2009) Survey on iterative learning control repetitive control and run-to run control. *J Process Control* 19:1589–1600
- Xu J-X (2011) A survey on iterative learning control for nonlinear systems. *Int J Control* 84(7):1275–1294

Kalman Filters

Frederick E. Daum
Raytheon Company, Woburn, MA, USA

Abstract

The Kalman filter is a very useful algorithm for linear Gaussian estimation problems. It is extremely popular and robust in practical applications. The algorithm is easy to code and test. There are many reasons for the popularity of the Kalman filter in the real world, including stability and generality and simplicity. Moreover, the real-time computational complexity is very reasonable for high-dimensional problems. In particular, the computational complexity scales as the cube of the dimension of the state vector.

Keywords

Controllability · Discrete time measurements · Estimation algorithm · Extended Kalman filter · Filtering · Gaussian errors · Linear dynamical system · Linear system · Observability · Recursive · Smoothing · Stability

Description of Kalman Filter

The Kalman filter is an algorithm that computes the best estimate of the state vector of a linear dynamical system given discrete time measurements of a linear function of the state vector corrupted by additive white Gaussian noise. The Kalman filter also quantifies the uncertainty in its estimate of the state vector, using the covariance matrix of estimation errors. The detailed equations of the Kalman filter algorithm and the problem that it solves are given in Gelb et al. (1974), which is the most accessible but thorough book on Kalman filters. The linear dynamical system can be time varying, but its parameters must be known exactly. The measurements can be made at arbitrary (nonuniform) discrete times, but these times must be known exactly. Likewise, the covariance matrices of the measurement errors and the process noise can be arbitrary and time varying, but the numerical values of these covariance matrices must be known exactly. Also, the initial uncertainty in the state vector must be Gaussian and the mean and covariance matrix can be arbitrary, but these must be known exactly. There is a very powerful theory of Kalman filter stability due to Kalman (1961), which guarantees that the Kalman filter is stable under very mild

technical assumptions which can always be satisfied in practice. In particular, the Kalman filter is stable for estimating the state vector of linear dynamical systems that are stable or unstable, for arbitrarily slow measurement rates, provided that the mild technical assumptions are fulfilled. These assumptions require that the dimension of the state vector is minimal and that the measurement error covariance matrix and the process noise covariance matrices are positive definite, although weaker conditions are also sufficient for stability in some cases; see Kailath et al. (2000) for such details on the stability of the Kalman filter. Kalman filter stability is connected with observability and controllability of the input-output model of the relevant dynamical system in Kalman (1961). The corresponding algorithm for continuous time linear measurements (with Gaussian additive white noise) and continuous time linear dynamical systems (with Gaussian additive white process noise) is called the Kalman-Bucy filter; see Kalman (1961).

Design Issues

In engineering practice, almost all real-world applications are nonlinear or non-Gaussian, and therefore, they do not fit the Kalman filter theory. Nevertheless, by approximating the nonlinear dynamics and measurements with linear equations, one can apply the Kalman filter theory; this is called the “extended Kalman filter” (EKF); see Gelb et al. (1974). The linearization of the nonlinear dynamics and measurements is made by computing the first-order Taylor series expansion and evaluating it at the estimated state vector; this is a very simple and fast approximation that is widely used in real-world applications, and it often gives good estimation accuracy, although there is no guarantee of that. Moreover, there is no guarantee that the EKF will be stable, even if the linearized system satisfies all the theoretical requirements for stability of the Kalman filter.

Even if the dynamics and measurements are exactly linear and if the measurement noise and process noise and initial uncertainty

are all exactly Gaussian with exactly known means and covariance matrices, there can still be significant practical problems with Kalman filter accuracy, owing to ill-conditioning. In particular, the Kalman filter can be extremely sensitive to quantization errors in the computer arithmetic and storage. On the other hand, there are many different methods to try to mitigate ill-conditioning including (1) double or quadruple or octuple precision arithmetic, (2) making the covariance matrices symmetric before and after every operation, (3) Tychonov regularization, (4) tuning the process noise covariance matrix, (5) coding the Kalman filter in principal coordinates or approximately principal coordinates (i.e., aligned with the eigenvectors of the state vector error covariance matrix), (6) sequential scalar measurement updates in a preferred order, and (7) various factorizations of the covariance matrices (e.g., square root, information matrix, information square root, upper triangular and lower triangular factorization, UDL, etc.). The classic book on error covariance matrix factorizations is by Bierman (2006). Unfortunately, there is no guarantee that the Kalman filter will work well even if all of these mitigation methods are used. Moreover, there is no useful theoretical analysis of this phenomenon, with the exception of a few not very tight upper bounds on the condition number. Plotting the numerical values of the condition number of the covariance matrix vs. time is often a helpful diagnostic. In certain real-world applications, the condition number of the Kalman filter error covariance matrix can be ten billion or larger.

Why Is the Kalman Filter So Useful and Popular?

The Kalman filter has been enormously successful in real-world applications, and it is interesting to reflect on why it has been so useful and so popular. In particular, Kalman himself believed that his filter was successful because it was based on probability rather than statistics; see Kalman (1978). For example, the error covariance matrix for the Kalman filter is computed

from the assumed dynamics and measurement model and the assumed values of the initial state uncertainty and process noise and measurement noise covariance matrices, rather than by computing sample covariance matrices. Likewise, the Kalman filter computes the estimated state vector from assumed Gaussian and linear probability models, rather than computing the sample mean, as would be done in statistics. There is substantial wisdom in Kalman's assertion, owing to the difficulty of estimating sample covariance matrices and sample vectors that are sufficiently accurate, given a limited number of samples and ill-conditioning and high-dimensional state vectors. A second reason that Kalman filters are so popular is that the real-time computational complexity is very reasonable for modern digital computers, even for problems with a high-dimensional state vector. In particular, the computational complexity of the Kalman filter scales as the cube of the dimension of the state vector; for example, the modern GPS system uses a Kalman filter with a state vector of dimension of about 1,000 to jointly estimate the orbits of the satellites in the GPS constellation. In 1960, when Kalman's paper was first published, digital computers were starting to become fast enough at reasonable cost to multiply large matrices in real time, which is the most challenging computation in a Kalman filter. Today computers are roughly ten orders of magnitude faster per unit cost than in 1960, and hence, we can run Kalman filters for high-dimensional problems on very inexpensive computers that fit into your wristwatch. A third reason that Kalman filters are popular is that the algorithms are easy to understand and code and test. A fourth reason is the guaranteed stability of the Kalman filter under very mild conditions which can always be satisfied in practical applications. A fifth reason is that the Kalman filter is optimal for time-varying unstable linear dynamics with time-varying measurement noise covariance and process noise covariance. A sixth reason is that one can use Kalman filters for nonlinear problems by approximating the nonlinear dynamics and measurement equations with a first-order Taylor series; this is called the extended Kalman filter (EKF), which is without

a doubt the most widely used algorithm in real-world estimation applications. A seventh reason is that the Kalman filter automatically provides a convenient quantification of uncertainty of the estimated state vector using the error covariance matrix. The final reason is that it is easy to test the accuracy of the Kalman filter by comparing the theoretical error covariance matrix to errors computed by Monte Carlo simulations of the filter; the two errors should agree approximately, and statistically significant discrepancies suggest bugs in the code or ill-conditioning of the error covariance matrices or nonlinearities or errors in modeling the dynamics or measurements.

Kalman's 1960 paper represented a big paradigm shift in two ways: (1) it exploited fast low-cost modern digital computers, whereas the literature up to that time did not, and (2) it used time domain methods rather than the ubiquitous Fourier transform methods, which limited the dynamics to steady state asymptotic in time, which in turn limited the theory to cover stable dynamics. Of course today we take both of these big points as normal engineering rather than revolutionary and surprising. The state of the art prior to Kalman's 1960 paper was the Wiener filter, which was based firmly on the Fourier transform. The Wiener filter required very lengthy and cumbersome algebraic spectral factorization with complex variables, resulting in erroneous formulas published in certain books, owing to algebraic errors which were not obvious and which could not be checked by computers, owing to the nonexistence of computer algebra software (e.g., MATHEMATICA) in 1960. Kalman explains many other problems with the Wiener filter in Kalman (2003).

Summary and Future Directions

To a large extent the Kalman filter theory is complete. There is a simple and useful theory of stability of the Kalman filter, which is completely lacking for nonlinear filters including the extended Kalman filter (EKF) and particle filters. Moreover, there are many robust versions of the Kalman filter that have been invented to

mitigate ill-conditioning of the covariance matrix as well as uncertainty in system models and system parameters. However, there remain many important design issues that need to be addressed for practical applications; see Daum (2005) for details. The most obvious issues include nonlinear measurements, nonlinear plant models, robustness to uncertainty in measurement models and plant models, non-Gaussian measurement noise and plant noise, non-Gaussian initial uncertainty in the state vector, and ill-conditioning of the error covariance matrix. That is, any deviation from the exact mathematical assumptions in the Kalman filter theory can cause problems in practice. It is the job of engineers to mitigate such problems and design filters that are robust to such perturbations. Kalman's recent papers on how the Kalman filter was invented and why it is so popular contain interesting and useful ideas; see Kalman (1978) and Kalman (2003).

Cross-References

- [Estimation, Survey on](#)
- [Extended Kalman Filters](#)
- [Nonlinear Filters](#)

Bibliography

- Bierman G (2006) Factorization methods for discrete sequential estimation. Dover, Mineola
- Daum F (2005) Nonlinear filters: beyond the Kalman filter. *IEEE AES Mag* 20:57–69
- Gelb A et al (1974) Applied optimal estimation. MIT, Cambridge
- Kailath T (1974) A view of three decades of linear filtering theory. *IEEE Trans Inf Theory* 20:146–181
- Kailath T, Sayed A, Hassibi B (2000) Linear estimation. Prentice Hall, Upper Saddle River
- Kalman R (1960) A new approach to linear filtering and prediction problems. *Trans ASME J* 82:35–45
- Kalman R (1961) New methods in Wiener filtering theory. RIAS technical report 61-1 (Feb 1961); also published in Bogdanoff J, Kozin F (eds) (1963) Proceedings of first symposium on engineering applications of random function theory and probability. Wiley, pp 270–388
- Kalman R (1978) A retrospective after twenty years: from the pure to the applied. In: Applications of the Kalman filter to hydrology, edited by Chao-lin Chiu, University of Pittsburgh, LC card number 78-069752, available on-line, pp 31–54
- Kalman R (2003) Discovery and invention: the Newtonian revolution in systems technology. *AIAA J Guid Control Dyn* 26(6):833–837
- Kalman R, Bucy R (1961) New results in linear filtering and prediction theory. *Trans ASME* 83: 95–107
- Sorenson H (1970) Least squares estimation from Gauss to Kalman. *IEEE Spectr* 7:63–68
- Stepanov OA (2011) Kalman filtering: past and present. *J Gyroscopy Navig* 2:99–110

KYP Lemma and Generalizations/Applications

Tetsuya Iwasaki

Department of Mechanical and Aerospace Engineering, University of California, Los Angeles, CA, USA

Abstract

Various properties of dynamical systems can be characterized in terms of inequality conditions on their frequency responses. The Kalman-Yakubovich-Popov (KYP) lemma shows equivalence of such frequency domain inequality (FDI) and a linear matrix inequality (LMI). The fundamental result has been a basis for robust and optimal control theories in the past several decades. The KYP lemma has recently been generalized to the case where an FDI on a possibly improper transfer function is required to hold in a (semi)finite frequency range. The generalized KYP lemma allows us to directly deal with practical situations where design parameters are sought to satisfy FDIs in multiple (semi)finite frequency ranges. Various design problems, including FIR filter and PID controller, reduce to LMI problems which can be solved via semidefinite programming.

Keywords

Bounded real · Frequency domain inequality · Linear matrix inequality ·

Multi-objective design · Optimal control ·
Positive real · Robust control

Introduction

In linear systems analysis and control design, dynamical properties are often characterized by frequency responses. The shape of a frequency response, as visualized by the Bode or Nyquist plot, is closely related to various performance measures including the steady state error, fast and smooth transient, and robustness against unmodeled dynamics. Hence, desired system properties can be formalized in terms of a set of frequency domain inequalities (FDIs) on selected transfer functions. The analysis and design problems then reduce to verification and satisfaction of the FDIs.

The Kalman-Yakubovich-Popov (KYP) lemma (Kalman 1963; Anderson 1967; Willems 1971; Rantzer 1996) establishes the equivalence between an FDI and a linear matrix inequality (LMI). The LMI is defined by state space matrices of the transfer function in the FDI so that the FDI holds true if and only if the LMI admits a solution. The LMI characterization of an FDI is useful since it replaces the process of checking the FDI at infinitely many frequency points by the search for a symmetric matrix satisfying a finite dimensional convex constraint defined by the LMI. In addition to exact and tractable computations, benefits of the LMI conditions include analytical understanding of robust and optimal controls through spectral factorizations and storage/Lyapunov functions. The KYP lemma is a fundamental result in the systems and control field that has provided, in the past half century, a theoretical basis for developments of various tools for system analysis and design.

A drawback of the KYP lemma is its inability to characterize an FDI in a finite frequency range. Feedback control designs typically involve a set of specifications given in terms of multiple FDIs in various frequency ranges. However, the KYP lemma is not capable of treating such FDIs directly since it has to consider the entire

frequency range. To address this deficiency, the KYP lemma has recently been generalized to characterize an FDI in a finite frequency range exactly (Iwasaki et al. 2000). Further generalizations (Iwasaki and Hara 2005) are available for FDIs within various frequency ranges for both continuous- and discrete-time, possibly improper, rational transfer functions. The generalized KYP lemma allows for direct multi-objective design of filters, controllers, and dynamical systems.

KYP Lemma

The KYP lemma may be motivated from various aspects, but let us explain it as an extension of a gain condition. Consider a stable linear system

$$\dot{x} = Ax + Bu, \quad G(s) := (sI - A)^{-1}B,$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the input, and $G(s)$ is the transfer function from u to x . If u is a disturbance to the system and x represents the error from a desired operating point, we may be interested in how large the state variables can become for a given magnitude of the disturbance. The gain $\|G(j\omega)\|$ captures this property for the case of a sinusoidal disturbance at frequency ω , where $\|\cdot\|$ denotes the spectral norm (= absolute value for a scalar). If $\|G(j\omega)\| < \gamma$ holds for all frequency ω with a small γ , then the system has a good disturbance attenuation property.

A version of the KYP lemma states that the FDI $\|G(j\omega)\| < \gamma$ with $\gamma = 1$ holds for all frequency ω if and only if there exists a symmetric matrix P satisfying the LMI:

$$\begin{bmatrix} PA + A^T P + I & PB \\ B^T P & -I \end{bmatrix} < 0.$$

Thus, existence of one particular P satisfying the LMI is enough to conclude that the gain is less than one for all, infinitely many, frequencies. This result is known as the bounded real lemma and has played a fundamental role in the robust and H_∞ control theories.

The KYP lemma can be introduced as a generalization of the bounded real lemma. First, note that the gain bound condition $\|G(j\omega)\| < 1$ and the LMI condition can equivalently be written as

$$\begin{bmatrix} G(j\omega) \\ I \end{bmatrix}^* \Theta \begin{bmatrix} G(j\omega) \\ I \end{bmatrix} < 0, \quad (1)$$

$$\begin{bmatrix} PA + A^T P & PB \\ B^T P & 0 \end{bmatrix} + \Theta < 0, \quad (2)$$

where

$$\Theta := \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}.$$

In these equations, the particular matrix Θ is chosen to describe the gain bound condition as a special case of the quadratic form (1), and we observe that Θ appears in the LMI as in (2). It turns out that the equivalence of (1) and (2) holds not only for this particular Θ but also for an arbitrary symmetric matrix Θ . This result is called the KYP lemma, which states that, given arbitrary matrices A , B , and $\Theta = \Theta^T$, the FDI (1) holds for all frequency ω if and only if there exists a matrix $P = P^T$ satisfying the LMI (2), provided A has no eigenvalues on the imaginary axis.

The FDI in (1) can be specialized to an FDI

$$\begin{bmatrix} L(j\omega) \\ I \end{bmatrix}^* \Pi \begin{bmatrix} L(j\omega) \\ I \end{bmatrix} < 0. \quad (3)$$

on transfer function

$$L(s) := C(sI - A)^{-1}B + D$$

by choosing

$$\Theta := \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^T \Pi \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}. \quad (4)$$

The choice of matrix Π allows for characterizations of important system properties involving gain and phase of $L(s)$. For instance, the FDI (3) with

$$\Pi := \begin{bmatrix} 0 & -I \\ -I & 0 \end{bmatrix}$$

gives $L(j\omega) + L(j\omega)^* > 0$. This is called the positive real property, with which the phase angle remains between $\pm 90^\circ$ when $L(j\omega)$ is a scalar.

Generalization

The standard KYP lemma deals with FDIs that are required to hold for all frequencies. To allow for more flexibility in practical system designs, the KYP lemma has been generalized to deal with FDIs in (semi)finite frequency ranges.

For instance, a version of the generalized KYP lemma states that the FDI (1) holds in the low frequency range $|\omega| \leq \varpi_\ell$ if and only if there exist matrices $P = P^T$ and $Q = Q^T > 0$ satisfying

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix}^T \begin{bmatrix} -Q & P \\ P & \varpi_\ell^2 Q \end{bmatrix} \begin{bmatrix} A & B \\ I & 0 \end{bmatrix} + \Theta < 0, \quad (5)$$

provided A has no imaginary eigenvalues in the frequency range. In the limiting case where ϖ_ℓ approaches infinity and the FDI is required to hold for the entire frequency range, the solution Q to (5) approaches zero, and we recover (2).

The role of the additional parameter Q is to enforce the FDI only in the low frequency range. To see this, consider the case where the system is stable and a sinusoidal input $u = \Re[\hat{u}e^{j\omega t}]$, with (complex) phasor vector $\hat{u}$, is applied. The state converges to the sinusoid $x = \Re[\hat{x}e^{j\omega t}]$ in the steady state where $\hat{x} := G(j\omega)\hat{u}$. Multiplying (5) by the column vector obtained by stacking $\hat{x}$ and $\hat{u}$ in a column from the right, and by its complex conjugate transpose from the left, we obtain

$$(\varpi_\ell^2 - \omega^2)\hat{x}^* Q \hat{x} + \begin{bmatrix} \hat{x} \\ \hat{u} \end{bmatrix}^* \Theta \begin{bmatrix} \hat{x} \\ \hat{u} \end{bmatrix} < 0.$$

In the low frequency range $|\omega| \leq \varpi_\ell$, the first term is nonnegative, enforcing the second term to be negative, which is exactly the FDI in (1). If ω is outside of the range, however, the first term is negative, and the FDI is not required to hold.

Similar results hold for various frequency ranges. The term involving Q in (5) can be expressed as the Kronecker product $\Psi \otimes Q$ with Ψ being a diagonal matrix with entries $(-1, \varpi_\ell^2)$. The matrix Ψ arises from characterization of the low frequency range:

$$\begin{bmatrix} j\omega \\ 1 \end{bmatrix}^* \Psi \begin{bmatrix} j\omega \\ 1 \end{bmatrix} = \varpi_\ell^2 - \omega^2 \geq 0.$$

By different choices of Ψ , middle and high frequency ranges can also be characterized:

	Low	Middle	High
Ω	$ \omega \leq \varpi_\ell$	$\varpi_1 \leq \omega \leq \varpi_2$	$ \omega \geq \varpi_h$
Ψ	$\begin{bmatrix} -1 & 0 \\ 0 & \varpi_\ell^2 \end{bmatrix}$	$\begin{bmatrix} -1 & j\varpi_c \\ -j\varpi_c & -\varpi_1\varpi_2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -\varpi_h^2 \end{bmatrix}$

where $\varpi_c := (\varpi_1 + \varpi_2)/2$ and Ω is the frequency range. For each pair (Ω, Ψ) , the FDI (1) holds in the frequency range $\omega \in \Omega$ if and only if there exist real symmetric matrices P and $Q > 0$ satisfying

$$F^T(\Phi \otimes P + \Psi \otimes Q)F + \Theta < 0, \quad (6)$$

provided A has no eigenvalues in Ω , where

$$\Phi := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad F := \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}.$$

Further generalizations are available Iwasaki and Hara (2005). The discrete-time case (frequency variable on the unit circle) can be similarly treated by a different choice of Φ . FDIs for descriptor systems and polynomial (rather than rational) functions can also be characterized in a form similar to (6) by modifying the matrix F . More specifically, the choices

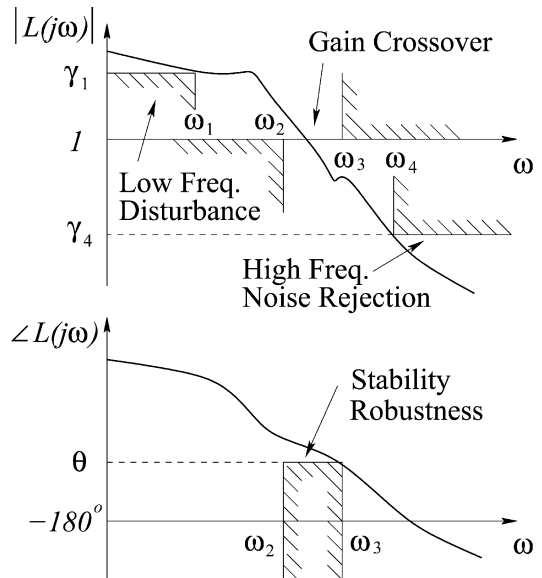
$$\Phi := \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad F := \begin{bmatrix} A & B \\ E & O \end{bmatrix}$$

give the result for the discrete-time transfer function $L(z) = (zE - A)^{-1}(B - zO)$.

Applications

The generalized KYP lemma is useful for a variety of dynamical system designs. As an example, let us consider a classical feedback control design via shaping of a scalar open-loop transfer function in the frequency domain. The objective is to design a controller $K(s)$ for a given plant $P(s)$ such that the closed-loop system is stable and possesses a good performance dictated by reference tracking, disturbance attenuation, noise sensitivity, and robustness against uncertainties.

Typical design specifications are given in terms of bounds on the gain and phase of the open-loop transfer function $L(s) := P(s)K(s)$ in various frequency ranges as shown in Fig. 1. The controller $K(s)$ should be designed so that the frequency response $L(j\omega)$ avoids the shaded regions. For instance, the gain should satisfy $|L(j\omega)| \geq 1$ for $|\omega| < \omega_2$ and $|L(j\omega)| \leq 1$ for $|\omega| > \omega_3$ to ensure the gain crossover occurs in the range $\omega_2 \leq \omega \leq \omega_3$, and the phase bound $\angle L(j\omega) \geq \theta$ in this range ensures robust stability by the phase margin.



KYP Lemma and Generalizations/Applications, Fig. 1
Loop shaping design specifications

The design specifications can be expressed as FDIs of the form (3), where a particular gain or phase condition can be specified by setting Π as

$$\pm \begin{bmatrix} 1 & 0 \\ 0 & -\gamma^2 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 & j - \tan \theta \\ -j - \tan \theta & 0 \end{bmatrix}$$

with $\gamma = \gamma_1, \gamma_4$, or 1, and the $+/-$ signs for upper/lower gain bounds. These FDIs in the corresponding frequency ranges can be converted to inequalities of the form (6) with Θ given by (4).

The control problem is now reduced to the search for design parameters satisfying the set of inequality conditions (6). In general, both coefficient matrices F and Θ may depend on the design parameters, but if the poles of the controller are fixed (as in the PID control), then the design parameters will appear only in Θ . If in addition an FDI specifies a convex region for $L(j\omega)$ on the complex plane, then the corresponding inequality (6) gives a convex constraint on P, Q , and the design parameters. This is the case for specifications of gain upper bound (disk: $|L| < \gamma$) and phase bound (half plane: $\theta \leq \angle L \leq \theta + \pi$). A gain lower bound $|L| > \gamma$ is not convex but can often be approximated by a half plane. The design parameters satisfying the specifications can then be computed via convex programming.

Various design problems other than the open-loop shaping can also be solved in a similar manner, including finite impulse response (FIR) digital filter design with gain and phase constraints in a passband and stop-band and sensor or actuator placement for mechanical control systems (Hara et al. 2006; Iwasaki et al. 2003). Control design with the Youla parametrization also falls within the framework if a basis expansion is used for the Youla parameter and the coefficients are sought to satisfy convex constraints on closed-loop transfer functions.

are expressed in terms of state space matrices without involving the frequency variable. The resulting LMI condition has been found useful for developing robust and optimal control theories.

A recent generalization of the KYP lemma characterizes an FDI for a possibly improper rational function in a (semi)finite frequency range. The result allows for direct solutions of practical design problems to satisfy multiple specifications in various frequency ranges. A design problem is essentially solvable when transfer functions are affine in the design parameters and are required to satisfy convex FDI constraints. An important problem, which falls outside of this framework and remains open, is the design of feedback controllers to satisfy multiple FDIs on closed-loop transfer functions in various frequency ranges. There have been some attempts to address this problem, but none of them has so far succeeded to give an exact solution.

The KYP lemma has been extended in other directions as well, including FDIs with frequency-dependent weights (Graham and de Oliveira 2010), internally positive systems (Tanaka and Langbort 2011), full rank polynomials (Ebihara et al. 2008), real multipliers (Pipeleers and Vandenberghe 2011), a more general class of FDIs (Gusev 2009), multi-dimensional systems (Bachelier et al. 2008), negative imaginary systems (Xiong et al. 2012), symmetric formulations for robust stability analysis (Tanaka and Langbort 2013), and multiple frequency intervals (Pipeleers et al. 2013). Extensions of the KYP lemma and related S-procedures are thoroughly reviewed in Gusev and Likhtarnikov (2006). A comprehensive tutorial of robust LMI relaxations is provided in Scherer (2006) where variations of the KYP lemma, including the generalized KYP lemma as a special case, are discussed in detail.

Summary and Further Directions

The KYP lemma has played a fundamental role in systems and control theories, equivalently converting an FDI to an LMI. Dynamical systems properties characterized in the frequency domain

Cross-References

- [Classical Frequency-Domain Design Methods](#)
- [H-Infinity Control](#)
- [LMI Approach to Robust Control](#)

Bibliography

- Anderson B (1967) A system theory criterion for positive real matrices. *SIAM J Control* 5(2): 171–182
- Bachelier O, Paszke W, Mehdi D (2008) On the Kalman-Yakubovich-Popov lemma and the multidimensional models. *Multidimens Syst Signal Proc* 19(3–4): 425–447
- Ebihara Y, Maeda K, Hagiwara T (2008) Generalized S-procedure for inequality conditions on one-vector-lossless sets and linear system analysis. *SIAM J Control Optim* 47(3):1547–1555
- Graham M, de Oliveira M (2010) Linear matrix inequality tests for frequency domain inequalities with affine multipliers. *Automatica* 46:897–901
- Gusev S (2009) Kalman-Yakubovich-Popov lemma for matrix frequency domain inequality. *Syst Control Lett* 58(7):469–473
- Gusev S, Likhtarnikov A (2006) Kalman-Yakubovich-Popov lemma and the S-procedure: a historical essay. *Autom Remote Control* 67(11):1768–1810
- Hara S, Iwasaki T, Shiokata D (2006) Robust PID control using generalized KYP synthesis. *IEEE Control Syst Mag* 26(1):80–91
- Iwasaki T, Hara S (2005) Generalized KYP lemma: unified frequency domain inequalities with design applications. *IEEE Trans Autom Control* 50(1):41–59
- Iwasaki T, Meinsma G, Fu M (2000) Generalized S-procedure and finite frequency KYP lemma. *Math Probl Eng* 6:305–320
- Iwasaki T, Hara S, Yamauchi H (2003) Dynamical system design from a control perspective: finite frequency positive-realness approach. *IEEE Trans Autom Control* 48(8):1337–1354
- Kalman R (1963) Lyapunov functions for the problem of Lur'e in automatic control. *Proc Natl Acad Sci* 49(2):201–205
- Pipeleers G, Vandenberghe L (2011) Generalized KYP lemma with real data. *IEEE Trans Autom Control* 56(12):2940–2944
- Pipeleers G, Iwasaki T, Hara S (2013) Generalizing the KYP lemma to the union of intervals. In: *Proceedings of European control conference, Zurich*, pp 3913–3918
- Rantzer A (1996) On the Kalman-Yakubovich-Popov lemma. *Syst Control Lett* 28(1):7–10
- Scherer C (2006) LMI relaxations in robust control. *Eur J Control* 12(1):3–29
- Tanaka T, Langbort C (2011) The bounded real lemma for internally positive systems and H-infinity structured static state feedback. *IEEE Trans Autom Control* 56(9):2218–2223
- Tanaka T, Langbort C (2013) Symmetric formulation of the S-procedure, Kalman-Yakubovich-Popov lemma and their exact losslessness conditions. *IEEE Trans Autom Control* 58(6): 1486–1496
- Willems J (1971) Least squares stationary optimal control and the algebraic Riccati equation. *IEEE Trans Autom Control* 16:621–634
- Xiong J, Petersen I, Lanzon A (2012) Finite frequency negative imaginary systems. *IEEE Trans Autom Control* 57(11):2917–2922

L1 Optimal Control

Murti V. Salapaka
Department of Electrical and Computer
Engineering, University of Minnesota
Twin-Cities, Minneapolis, MN, USA

Abstract

An approach to feedback controller synthesis involves minimizing a uniform across time bound for the size of a controlled output, given a bound of the same kind for the disturbance input. In the case of linear time-invariant dynamics, this corresponds to minimizing the ℓ_1 norm of the closed-loop impulse response. This optimal control problem can be recast exactly as a finite-dimensional linear program for single-input-single-output systems. On the other hand, for multiple-input-multiple-output systems, the optimal control problem can be approximated to arbitrary accuracy by a linear program of size that grows as the accuracy is improved. An overview of these results is provided in the entry below.

Keywords

Feedback control · Bounded signals · Linear time-invariant systems · Youla parametrization · Linear programming

Glossary

- ℓ_1 Space of right-sided absolutely summable real sequences with the norm given by $\|x\|_{\ell_1} = \sum_{k=0}^{\infty} |x(k)|$
- $\hat{x}(\lambda)$ λ -Transform of a right-sided real sequence $x = (x(k))_{k=0}^{\infty}$ defined as $\hat{x}(\lambda) = \sum_{k=0}^{\infty} x(k)\lambda^k$.
- P_N Truncation operator for sequences defined as $P_N \begin{pmatrix} x(0) & x(1) & \dots \end{pmatrix} = \begin{pmatrix} x(0) & x(1) & \dots & x(N-1) & 0 & \dots \end{pmatrix}$

Introduction: The ℓ_1 Norm

Feedback control effectively reduces the impact of uncertainty; indeed, apart from stabilizing unstable systems, minimizing the adverse effects of uncertainty is its primary goal. In any control system design where mathematical descriptions of signals and system is assumed, the models can only approximate the reality where the description is incomplete or its complexity makes its incorporation in controller synthesis ineffective.

In modern control design, uncertainty is classified into two classes: (i) signal uncertainty and (ii) model uncertainty. Signal uncertainty is caused by an incomplete knowledge of the inputs affecting the system; for example, for an aircraft, there is scant or incomplete information

on the direction and strength of the wind that affects its motion. However, it is often possible to provide bounds that encapsulate the knowledge of the external inputs, whereby these exogenous influences can be restricted to a set. Various measures can be employed in characterizing the inputs. A widely used measure in a probabilistic setting is the variance of the input modeled as white noise. In a deterministic description, the energy, $\|\cdot\|_{\ell_2}$, and the maximum magnitude, $\|\cdot\|_{\ell_\infty}$ of the signal are often employed. The ℓ_∞ norm and the ℓ_2 norms are defined as

$$\|w\|_{\ell_\infty} := \sup_{0 \leq t < \infty} |w(t)| \text{ and}$$

$$\|w\|_{\ell_2} := \sqrt{\sum_{t=0}^{\infty} |w(t)|^2},$$

respectively, for a discrete time scalar signal w . For a vector valued signal w with $w(t) \in R^d$ the norms are defined as

$$\|w\|_{\ell_\infty} := \max_{1 \leq i \leq d} \sup_{0 \leq t < \infty} |w_i(t)| \text{ and } \|w\|_{\ell_2}$$

$$:= \sqrt{\sum_{t=0}^{\infty} \sum_{i=1}^d |w_i(t)|^2}.$$

For the purposes here, the exogenous input to the system is restricted to lie in a set $\mathcal{W} = \{w : \|w\|_{\ell_\infty} \leq M\}$.

To quantify the performance of a system, apart from using a norm for inputs, it is essential to employ measures on outputs. In many applications, outputs are constrained from exceeding a magnitude bound. For example, tracking errors need to remain within limits in many application where reference trajectories are to be followed over time and in many applications actuators and safety concerns necessitate signals do not cross imposed limits (e.g., in power applications voltage signals need to be within acceptable bounds). Here, the ℓ_∞ norm is an apt measure to quantify the output signals.

ℓ_1 optimal control is pertinent to the case where the input and output signals are quantified using the ℓ_∞ norm of the signals, where the

input is considered to lie in a set $\mathcal{W} = \{w : \|w\|_{\ell_\infty} \leq 1\}$. Here we restrict our study to linear, time-invariant (LTI) causal discrete time systems; systems will be denoted using boldface symbols. Thus if a LTI system, $\mathbf{T}$, maps a input signal, $w(\cdot)$, to yield an output signal, $e(\cdot) = (\mathbf{T}w)(\cdot)$, where the impulse response of the system $\mathbf{T}$ is T , then the output, $e = \mathbf{T}w$, satisfies $(\mathbf{T}w)(t) = (T * w)(t)$ (or $\hat{e} = \hat{T}\hat{w}$) where $*$ denotes convolution operator between signals and $\hat{T}$, the transfer function map, is the λ -Transform of T . A possible quantification of the system performance is the worst possible output (evaluated using the ℓ_∞ norm) over any input from the admissible class $\mathcal{W}$. Thus,

$$\sup_{w \in \mathcal{W}} \frac{\|(\mathbf{T}w)(\cdot)\|_{\ell_\infty}}{\|w\|_{\ell_\infty}},$$

which considers the worst-case amplification of a input signal, provides a quantification of the performance of the system $\mathbf{T}$. Note that the above measure on the system $\mathbf{T}$ is the ℓ_∞ induced norm of the system $\mathbf{T}$.

The ℓ_1 norm of a scalar signal, h , is given by

$$\|h\|_{\ell_1} = \sum_{t=0}^{\infty} |h(t)|.$$

Suppose for a linear system, the exogenous input w is n_w -dimensional and the regulated output is n_z -dimensional. Let the response of the i th output of the system, with the j th input being an impulse, be $T_{ij}(\cdot)$. Then the ℓ_1 norm of the multiple-input-multiple-output system $\mathbf{T}$ is defined as

$$\|\mathbf{T}\|_{\ell_1} = \max_{1 \leq i \leq n_z} \sum_{j=1}^{n_w} \|T_{ij}\|_{\ell_1}.$$

The following result shows how the ℓ_∞ induced norm of the system $\mathbf{T}$ is related to the ℓ_1 norm of the impulse response, T , of the system.

Theorem 1

$$\sup_{w \in \mathcal{W}} \frac{\|(\mathbf{T}w)(\cdot)\|_{\ell_\infty}}{\|w(\cdot)\|_{\ell_\infty}} = \|\mathbf{T}\|_{\ell_1}.$$

Note that for a single-input-single-output system the ℓ_1 norm of the system is the absolute sum $\sum_{t=0}^{\infty} |h(t)|$ of the impulse response, h , of the system. The above result motivates the ℓ_1 norm for controller synthesis.

Control Synthesis

Consider a system, $\mathbf{G}$, with external input, w , that affect the outputs, the control input u with measured outputs y and output, z , to be regulated. The measurements are used by the controller to produce the control signal u . Figure 1a describes the generalized plant G . Here

$$\begin{pmatrix} z \\ y \end{pmatrix} = \underbrace{\begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix}}_G * \begin{pmatrix} w \\ u \end{pmatrix}, \text{ with } u = K * y$$

where G and K are the impulse responses of $\mathbf{G}$ and the controller $\mathbf{K}$. Here, $z = G_{11} * w + G_{12} * u$ and $y = G_{21} * w + G_{22} * u$. Substituting $u = K * y$, the closed-loop system, $F_\ell(\mathbf{G}, \mathbf{K})$, is obtained as

$$\hat{z} = \underbrace{[\hat{G}_{11} + \hat{G}_{12} \hat{K} (I - \hat{G}_{22} \hat{K})^{-1} \hat{G}_{21}]}_{\hat{F}_\ell(\hat{G}, \hat{K})} \hat{w}.$$

It is important that only closed-loop maps that result via controllers that stabilize the interconnection in Fig. 1a be considered.

Definition 2 (Stabilizing controllers, Achievable closed-loop maps) A controller $\mathbf{K}$ which results in a stable interconnection of $\mathbf{G}$ and $\mathbf{K}$ systems represented in Fig. 1a is a stabilizing controller. A transfer function, $\hat{\Phi}$, that maps the exogenous inputs $\hat{w}$ to regulated variables

$\hat{z}$ is said to be an achievable closed-loop map if there exists a stabilizing controller $\mathbf{K}$ such that $\hat{\Phi} = \hat{G}_{11} + \hat{G}_{12} \hat{K} (I - \hat{G}_{22} \hat{K})^{-1} \hat{G}_{21}$.

The ℓ_1 optimal controller synthesis problem is to determine a stabilizing controller $\mathbf{K}$ where the worst-case response of the closed-loop system over all signals in $\mathcal{W}$ is minimized. Thus the ℓ_1 controller synthesis problem is given by

$$\begin{aligned} v &= \inf_{\mathbf{K} \text{ stabilizing}} \sup_{w \in \mathcal{W}} \frac{\|F_\ell(\mathbf{G}, \mathbf{K})w\|_{\ell_\infty}}{\|w\|_{\ell_\infty}} \\ &= \inf_{\mathbf{K} \text{ stabilizing}} \|F_\ell(G, K)\|_{\ell_1} \end{aligned} \quad (1)$$

where $F_\ell(G, K)$ is the impulse response of the closed-loop map from w to z . The following example is illustrative of the discussion thus far.

Example

Consider the problem described in Fig. 2a where a plant $\mathbf{P}$ is subjected to a input disturbance modeled as filtered signal $\mathbf{W}w$, where $\mathbf{W}$ models the frequency content of the disturbance and $w \in \mathcal{W} = \{v : \|v\|_\infty \leq 1\}$. The generalized plant is described as

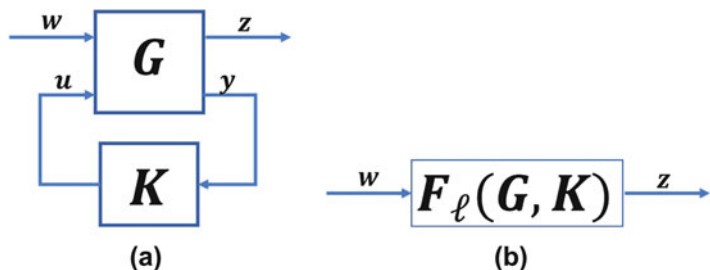
$$\begin{pmatrix} \hat{z} \\ \hat{y} \end{pmatrix} = \underbrace{\begin{pmatrix} \hat{P}\hat{W} & -\hat{P} \\ \hat{P}\hat{W} & -\hat{P} \end{pmatrix}}_{\hat{G}} \begin{pmatrix} \hat{w} \\ \hat{u} \end{pmatrix},$$

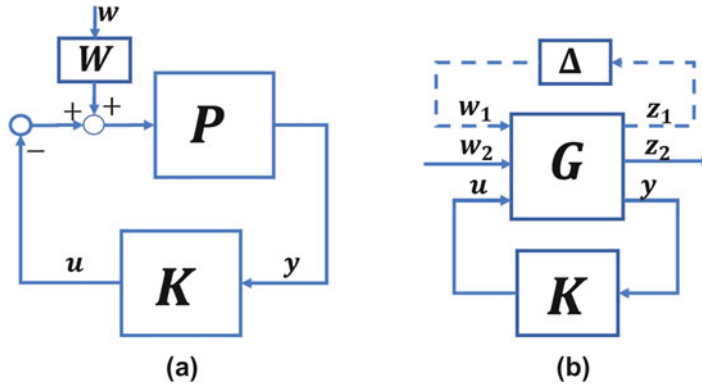
with $\hat{G}_{11} = \hat{G}_{21} = \hat{P}\hat{W}$ and $\hat{G}_{12} = \hat{G}_{22} = -\hat{P}$. The closed-loop map from w to z has a transfer function

$$\begin{aligned} \hat{F}_\ell(\hat{G}, \hat{K}) &= \hat{G}_{11} + \hat{G}_{12} \hat{K} (I - \hat{G}_{22} \hat{K})^{-1} \hat{G}_{21} \\ &= \hat{P}\hat{W} - \hat{P}\hat{K} (1 + \hat{P}\hat{K})^{-1} \hat{P}\hat{W}. \end{aligned}$$

L1 Optimal Control,

Fig. 1 (a) shows a generalized plant $\mathbf{G}$ in a feedback interconnection with the controller $\mathbf{K}$. (b) shows the closed-loop system, $F_\ell(\mathbf{G}, \mathbf{K})$ which maps exogenous inputs to regulated output z





L1 Optimal Control, Fig. 2 (a) shows a plant P subject to input disturbance modeled as a magnitude constrained signal w filtered by W . (b) shows the interconnection between G , K , and Δ . The unstructured uncertainty Δ is

characterized by $\|\Delta\|_{\ell_1} < \frac{1}{\gamma}$. To maintain stability, a hard constraint that the closed-loop map from w_1 to z_1 should be less than γ is imposed

Thus the ℓ_1 optimal controller synthesis problem translates to solving,

$$\nu = \inf_{\mathbf{K} \text{ stabilizing}} \|\hat{P}\hat{W} - \hat{P}\hat{K}(1 + \hat{P}\hat{K})^{-1}\hat{P}\hat{W}\|_{\ell_1}.$$

It is evident that the ℓ_1 optimal controller synthesis problem is non-convex in the optimization variable $\mathbf{K}$ as posed in (1).

Solution Methodologies

The solution approaches to the ℓ_1 optimal controller synthesis problem have to address the following two challenges: (i) the non-convexity of the optimization problem (1) in the argument $\mathbf{K}$ that needs to be stabilizing and (ii) the possible infinite dimensional nature of the optimization problem. To address the non-convexity, Youla parametrization is employed.

Youla parameterization (see Youla et al. (1976) and related articles Optimal Control via Factorization and Model Matching and Polynomial/algebraic design methods in this encyclopedia) translates the search over stabilizing controllers to search over Q parameters where the closed-loop map is convex in Q .

Theorem 3 (Youla Parametrization) Let G be stabilizable from u and detectable from y . Let $\hat{G}$

be the transfer-function map with

$$\begin{pmatrix} \hat{z} \\ \hat{y} \end{pmatrix} = \underbrace{\begin{pmatrix} \hat{G}_{11} & \hat{G}_{12} \\ \hat{G}_{21} & \hat{G}_{22} \end{pmatrix}}_{\hat{G}} \begin{pmatrix} \hat{w} \\ \hat{u} \end{pmatrix}.$$

Suppose $\hat{G}_{22}$ admits a double co-prime factorization where

$$G_{22} = \hat{N}\hat{M}^{-1} = \hat{\tilde{M}}^{-1}\hat{\tilde{N}},$$

with

$$\begin{pmatrix} \hat{\tilde{X}} & -\hat{\tilde{Y}} \\ -\hat{\tilde{N}} & \hat{\tilde{M}} \end{pmatrix} \begin{pmatrix} \hat{M} & \hat{Y} \\ \hat{N} & \hat{X} \end{pmatrix} = I,$$

and $\tilde{X}, \tilde{Y}, \tilde{M}, \tilde{N}, X, Y, M, N$ have finite ℓ_1 norms.

Then, $\hat{\Phi}$ is a closed-loop map with $\hat{\Phi} = \hat{G}_{11} + \hat{G}_{12}\hat{K}(I - \hat{G}_{22}\hat{K})^{-1}\hat{G}_{21}$ for a stabilizing controller $\mathbf{K}$ if and only if

$$\hat{\Phi} \in \{\hat{H} - \hat{U}\hat{Q}\hat{V} : Q \in \ell_1\}$$

where $\hat{H} = \hat{G}_{11} + \hat{G}_{12}\hat{Y}\hat{M}\hat{G}_{21}$, $\hat{U} = \hat{G}_{12}\hat{M}$ and $\hat{V} = \hat{\tilde{M}}$.

Using Youla Parameterization, the ℓ_1 optimal control problem can be recast in terms of the Youla parameter Q .

Theorem 4 *The following holds:*

$$v = \inf_{\hat{K}, \Phi} \{ \|\Phi\|_{\ell_1} : \hat{K} \text{ stabilizing}, \hat{\Phi} = \hat{G}_{11} + \hat{G}_{12} \hat{K} (I - \hat{G}_{22} \hat{K})^{-1} \hat{G}_{21} \} \quad (2)$$

$$= \inf_{Q \in \ell_1, \Phi \in \ell_1} \{ \|\Phi\|_{\ell_1} : \Phi = H - U * Q * V \}. \quad (3)$$

Note that (3) is convex in the optimization variables Φ and Q ; however, possibly infinite-dimensional in the impulse response parameters of Q and Φ . One of the earliest approaches taken is to eliminate the variable Q from the optimization in (3) and to recast the problem retaining only the closed-loop variable Φ . This approach is elucidated for the SISO setting in what follows.

SISO Case

Theorem 5 *Consider the SISO case with $n_w = n_z = 1$. Suppose Youla Parameterization yields that Φ is achievable closed-loop map via a stabilizing controller if and only if $\Phi = H - U * Q$ (as the system is SISO, V of the Youla parameterization can be subsumed into U), for some $Q \in \ell_1$. Furthermore assume that the transfer function $\hat{U}$ has no zeros on the unit circle with all its unstable zeros (which are zeros inside the unit disc) $z_1, z_2, \dots, z_n$ distinct. Let*

$$A = \begin{pmatrix} 1 & z_1 & z_1^2 & \dots \\ 1 & z_2 & z_2^2 & \dots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & z_n & z_n^2 & \dots \end{pmatrix}, \quad b = \begin{pmatrix} \hat{H}(z_1) \\ \hat{H}(z_2) \\ \vdots \\ \hat{H}(z_n) \end{pmatrix}.$$

Then

$$\begin{aligned} & \{ \Phi : \Phi = H - U * Q, Q \in \ell_1 \} \\ &= \{ \Phi : A\Phi = b, \Phi \in \ell_1 \}, \end{aligned}$$

where $A\Phi = b$ denotes the n constraints $\sum_{k=0}^{\infty} \Phi(k)z_i^k = \hat{H}(z_i)$ for $i = 1, \dots, n$.

Thus

$$\begin{aligned} v &= \inf_{\Phi} \{ \|\Phi\|_{\ell_1} : \Phi = H - U * Q, Q \in \ell_1 \} \\ &= \inf_{\Phi \in \ell_1} \{ \|\Phi\|_{\ell_1} : A\Phi = b \}. \end{aligned}$$

Proof. A sketch of the proof is provided. Note that $\hat{\Phi} = \hat{H} - \hat{U} \hat{Q}$ implies that $\hat{Q} = \frac{\hat{H} - \hat{\Phi}}{\hat{U}}$. Evidently, $Q \in \ell_1$ implies that the unstable zeros of $\hat{U}$ are cancelled by the zeros of $\hat{H} - \hat{\Phi}$ which in turn implies $\hat{\Phi}(z_i) = \hat{H}(z_i)$ for all unstable zeros, z_i , of $\hat{U}$. Thus for $\Phi \in \ell_1$ with $A\Phi = b$, a $Q \in \ell_1$ can be obtained such that $\hat{\Phi} = \hat{H} - \hat{U} \hat{Q}$. This completes the sketch of the proof. $\square$

The assumption that all unstable zeros of $\hat{U}$ are distinct can be removed and is included for ease of presentation. Note that the problem $\inf_{\Phi \in \ell_1} \{ \|\Phi\|_{\ell_1} : A\Phi = b \}$ appears to be infinite dimensional in $\Phi(t)$, $t = 0, 1, \dots$

The condition $A\Phi = b$ with $\Phi \in \ell_1$ replaces the constraint that $\Phi = H - U * Q$ with $Q \in \ell_1$. Here the “unstable” zeros of $\hat{U}$ are interpolated to obtain the condition $A\Phi = b$. Using the zero-interpolation, the seminal article (Dahleh and Pearson 1987) established that the optimal solution is a finite impulse response sequence where the dimension of the impulse response can be determined a priori. Suppose the dimension of the optimal impulse response Φ^o is N . Then the ℓ_1 optimal control problem in the SISO setting (with the assumption that $\hat{U}$ has no zeros on the unit circle) results in a finite dimensional problem that can be solved using linear programming. Indeed with A_N denoting a matrix which retains the first $N - 1$ columns of A the result below can be established.

Theorem 6 *The following holds:*

$$\begin{aligned} v = \inf_{\Phi \in R^N} \{ \|\Phi\|_{\ell_1} : A_N \Phi = b \} = & \min \sum_{k=0}^{N-1} (\Phi_+(k) + \Phi_-(k)) \\ & \text{subject to} \\ & A_N(\Phi_+(k) - \Phi_-(k)) = b \\ & \Phi_+(k) \geq 0, \Phi_-(k) \geq 0. \end{aligned}$$

The above strategy also proves effective in addressing other SISO problems where multiple objectives involving ℓ_1 and H_2 norm are present (see Salapaka et al. 1997 and Voulgaris 1995).

Though the design in the SISO case for ℓ_1 optimal controller can be obtained via a finite dimensional linear program, for the MIMO case, the strategy of casting the optimization problem in terms of the closed-loop map Φ alone by imposing zero-interpolation conditions does not suffice.

MIMO Case

For the MIMO case, consider translating the condition $\Phi = H - U * Q * V$, with $Q \in \ell_1$ to constraints only on Φ . Here, $\hat{U} \hat{Q} \hat{V} = \hat{H} - \hat{\Phi}$ where $\hat{U}$, $\hat{Q}$, $\hat{V}$ are $n_z \times n_u$, $n_u \times n_y$, $n_y \times n_w$ matrices of transfer functions where $\hat{H}$ and $\hat{\Phi}$ are $n_z \times n_w$ dimensional. There is no loss of generality in assuming that $\hat{U}$ and $\hat{V}$ have normal rank n_u and n_y , respectively. In the MIMO case, $\hat{U} \hat{Q} \hat{V} = \hat{H} - \hat{\Phi}$, with $Q \in \ell_1$, imposes conditions where the unstable zeros (now with directions) of $\hat{U}$ and $\hat{V}$ have to be interpolated by the zeros of $\hat{H} - \hat{\Phi}$. However, in the MIMO case, in addition to the zero-interpolation conditions, rank-interpolation conditions need to be imposed on $\hat{H} - \hat{\Phi}$ when $n_u = n_z$ and $n_y = n_w$ is not met. The rank-interpolation conditions impose infinite number of constraints on the impulse response Φ . Here one approach is to limit the number of rank interpolation constraints to a finite number and include only finite impulse response sequences for Φ that result in finite dimensional approximations to provide converging upper and lower bounds (see Dahleh and Diaz-Bobillo 1994). The limiting controller is obtained

by increasing the number of constraints and variables included, and the closed-loop map needs to guarantee that the corresponding $Q \in \ell_1$. In such an approach, there is little advantage in not including Q in the optimization.

Moreover, in a MIMO setting, it is more likely that a hard constraint is needed on the performance between an input-output pair while optimizing the ℓ_1 performance between another input-output pair. Thus consider the case where exogenous input, $w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$ and the regulated output $z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$.

For example suppose the model is uncertain with the uncertainty modeled by Δ , where it is known that $\|\Delta\|_{\ell_1} < \frac{1}{\gamma}$ and enters the system description as shown in Fig. 1b. Suppose the closed-loop map $\mathbf{F}_\ell(\mathbf{G}, \mathbf{K})$ is such that the closed-loop map from w_1 to z_1 has ℓ_1 norm less than γ . Then using small-gain theorem, it follows that the closed-loop system with Δ in Fig. 1b will remain stable. As stability is a primary concern, it is appropriate that the controller ensures that the ℓ_1 norm of the map from w_1 to z_1 be bounded by γ . Suppose in addition it is desired that ℓ_1 norm of the closed-loop transfer map Φ_{22} , from w_2 to z_2 , is small (a soft objective); see Fig. 2b. The resulting optimization problem takes the form:

$$\inf_{\mathbf{K} \text{ stabilizing}} \{ \|\Phi_{22}(\mathbf{K})\|_{\ell_1} : \|\Phi_{11}(\mathbf{K})\|_{\ell_1} \leq \gamma \}.$$

Using the Youla-Parametrization approach, closed-loop maps achieved via stabilizing controllers are given by $H - U * Q * V$ and the problem becomes

$$\nu = \inf_{Q \in \ell_1} \left\{ \|H^{22} - U^2 * Q * V^2\|_{\ell_1} : \right. \\ \left. \|H^{11} - U^1 * Q * V^1\|_{\ell_1} \leq \gamma \right\}.$$

The above problem might be ill-posed where there is no Q in ℓ_1 that achieves the infimizing value ν . This motivates including a “regularizing” constraint, on Q where the problem to be solved becomes

$$\nu = \inf_{Q \in \ell_1} \left\{ \|H^{22} - U^2 * Q * V^1\|_{\ell_1} : \|H^{11} - U^1 * Q * V^2\|_{\ell_1} \leq \gamma, \|Q\|_{\ell_1} \leq \beta \right\}. \quad (4)$$

The problem (4) is infinite dimensional in the impulse response parameters $Q(k)$. The following problems, where the closed-loop maps are truncated, provide a lower bound on ν for each N :

$$\nu_N = \inf_{Q \in \ell_1} \{ \|P_N(H^{22} - U^2 * Q * V^2)\|_{\ell_1} : \\ \|P_N(H^{11} - U^1 * Q * V^1)\|_{\ell_1} \leq \gamma, \\ \|Q\|_{\ell_1} \leq \beta \}. \quad (5)$$

Note that the above problem is finite-dimensional with $Q \in R^N$. The following problems provide upper bounds on ν by truncating the Youla parameter Q , for each N :

$$\nu^N = \inf_{Q \in \ell_1} \{ \|H^{22} - U^2 * P_N(Q) * V^2\|_{\ell_1} : \\ \|H^{11} - U^1 * P_N(Q) * V^1\|_{\ell_1} \leq \gamma, \\ \|P_N(Q)\|_{\ell_1} \leq \beta \}. \quad (6)$$

If it is assumed that $\hat{H}$, $\hat{U}$, and $\hat{V}$ have numerator polynomials with finite number of terms then it can be shown that (6) is a finite dimensional problem. The assumption that $\hat{H}$, $\hat{U}$, and $\hat{V}$ are polynomial matrices of λ is not restrictive; indeed, the denominator terms of $\hat{H}$, $\hat{U}$, and $\hat{V}$ which are stable can be absorbed into the parameter $\hat{Q}$.

Problems (5) and (6) associated with ν_N and ν^N can be solved via finite-dimensional linear programming following similar approach to the one presented for the SISO case in Theorem 6. Moreover, the following can be established.

Theorem 7 $\nu_N \nearrow \nu$ and $\nu^N \searrow \nu$.

The above methodology is elucidated in Qi et al. (2004) with accompanying software in Qi et al. (2001); here, it is also established that in the framework described above, it is relatively straightforward to incorporate structural constraints induced on the Q parameter that arise due to structural constraints on the controller $\mathbf{K}$. Another methodology is the scaled Q approach (Khammash 2000) where a scaled ℓ_1 norm of Q is introduced into the optimization objective which plays the role of the ℓ_1 norm constraint on the Q parameter in the regularized setup of (4).

Summary

The ℓ_1 optimal controller synthesis can incorporate time-domain constraints directly. For the SISO case, an optimal controller can be determined by solving a linear programming problem. For the MIMO case, under realistic assumptions, the optimal performance can be approximated within any desired accuracy by controllers determined via finite dimensional linear programming problems. Many of the mathematical tools employed for solving the ℓ_1 optimal controller problem have proven to be instrumental in solving larger class of problems including multi-objective controller synthesis problem (Elia and Dahleh 1997; Salapaka and Dahleh 2000). A key aspect of future efforts needs to be in adoption of the ℓ_1 control framework in applications.

Cross-References

- [Optimal Control via Factorization and Model Matching](#)
- [Polynomial/algebraic design methods](#)

Bibliography

- Dahleh MA, Diaz-Bobillo IJ (1994) Control of uncertain systems: a linear programming approach. Prentice-Hall, Inc.
- Dahleh M, Pearson B (1987) ℓ_1 -optimal feedback controllers for mimo discrete-time systems. *IEEE Trans Autom Control* 32(4):314–322
- Elia N, Dahleh MA (1997) Controller design with multiple objectives. *IEEE Trans Autom control* 42(5):596–613
- Khammash M (2000) A new approach to the solution of the ℓ_1 control problem: the scaled-Q method. *IEEE Trans Autom Control* 45(2):180–187
- Qi X, Khammash M, Salapaka M (2001) A matlab package for multiobjective control synthesis. In: *Proceedings of the 40th IEEE Conference on Decision and Control*, IEEE, vol 4, pp 3991–3996
- Qi X, Salapaka MV, Voulgaris PG, Khammash M (2004) Structured optimal and robust control with multiple criteria: a convex solution. *IEEE Trans Autom Control* 49(10):1623–1640
- Salapaka MV, Dahleh M (2000) Multiple objective control synthesis. Springer, London
- Salapaka MV, Dahleh M, Voulgaris P (1997) Mixed objective control synthesis: optimal ℓ_1/h_2 control. *SIAM J Control Optim* 35(5):1672–1689
- Voulgaris PG (1995) Optimal h_2/ℓ_1 control via duality theory. *IEEE Trans Autom Control* 40(11):1881–1888
- Youla D, Jabr H, Bongiorno J (1976) Modern Wiener-Hopf design of optimal controllers—part ii: the multi-variable case. *IEEE Trans Autom Control* 21(3):319–338

Lane Keeping Systems

Paolo Falcone
 Department of Electrical Engineering,
 Mechatronics Group, Chalmers University of
 Technology, Göteborg, Sweden
 Engineering Department “Enzo Ferrari”,
 University of Modena and Reggio Emilia,
 Modena, Italy

Abstract

This chapter provides an overview of lane keeping systems (LKS). First, a general architecture is introduced common to both existing and future LKS, along with a description of the used sensors and actuators. Then, the threat

assessment and the lane position control problems are discussed highlighting challenges, existing solutions, and future directions.

Keywords

ADAS · Vehicle control · Intelligent transportation systems · Lane keeping

Introduction

Lane keeping systems aim at preventing unintended lane departures, which may lead to accidents involving surrounding obstacles and vehicles. This goal can be achieved by resorting to different combinations of sensor sets, threat assessment criteria, and interventions. For example, lane departure warning (LDW) systems monitor the lane markers through cameras installed behind the windshield and issue a (visual, acoustic, haptic) warning to the driver if the vehicle is too close to or has crossed the lane boundaries. The haptic warning typically consists of a slight counter-steering force or a vibration applied to the steering wheel. In Lane Departure Assist or Aid (LDA) systems, a steering torque is applied in order to keep the vehicle within the intended lane, once a crossing of the lane markings is detected or predicted within a certain time interval. Interventions and warnings can be differently combined, depending on the specific system implementation.

While basic LDW and LDA systems are triggered by crossing (or the prediction of crossing) the lane markers, more advanced (emergency) lane keeping systems (Cars 2019; Mercedes 2013) intervene if a threat of collision with the surrounding vehicles and objects is detected. Such advanced functionalities can be implemented by using radars to track the surrounding objects. Clearly, depending on the traffic scenario, assessing the threat of collision with surrounding vehicles may be complicated. Nevertheless, with the development (deployment) of autonomous driving (AD) technologies, enhanced perception and increased awareness of the surrounding traffic situation may allow interventions that also account for

the road users around the vehicle, thus allowing LDW/LDA systems to act only if necessary.

In this chapter, we overview the most important aspects in the design of a lane keeping system. The chapter is structured as follows. Section “Lane Keeping System Architecture” illustrates the architecture of a generic lane keeping system. Section “Sensing and Actuation” reviews the most used sensors suitable for lane keeping applications. Section “Decision-Making and Control” introduces the threat assessment and the lane position control problems highlighting the most relevant challenges.

Lane Keeping System Architecture

The main components of a LKS and their interconnections are shown in Fig. 1, where a general architecture is sketched that can describe existing LDW and LDA systems, as well as future LKS described in section “Introduction”.

Relative positions and velocities of the host vehicle are measured by one radar, typically installed on the front of the vehicle, and possibly by the camera, typically installed behind the windshield. Position of the host vehicle within the lane and further information like road geometry are measured by the camera. These measurements are then fused by the *sensor fusion* module to provide accurate measurements of the position and velocity of the vehicle w.r.t. the surrounding environment and the lane in

the widest range of operating conditions and scenarios.

The task of the *decision-making and control module* is to assess the risk that the vehicle crosses the lane in a dangerous way and, possibly, to take an action that can range from warning the driver or issuing an assisting intervention like braking and/or steering. Such steering and braking commands are actually implemented by *low-level controllers*.

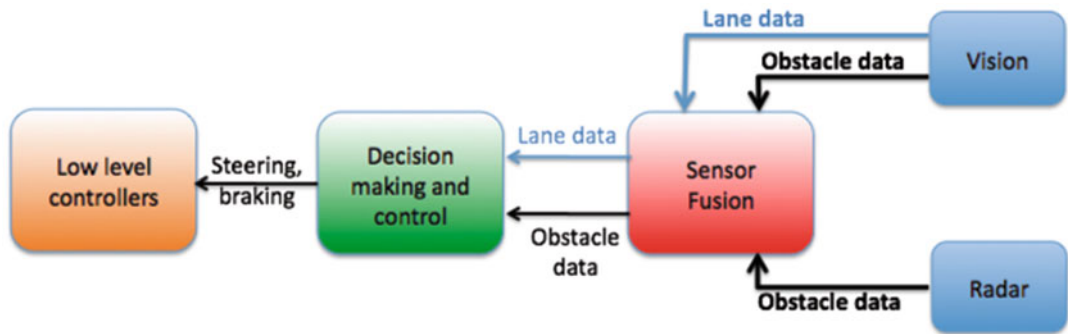
The different modules will be overviewed in the following sections.

Sensing and Actuation

Current LKS systems rely on a sensor package consisting of a forward-looking radar and camera, which are described next in sections “Radar” and “Vision Systems”. Additional sensors, in some system already available (Cars 2017), may include rear-looking radars and cameras and blind spot cameras, which are installed in the side mirrors.

Radar

Radars for automotive applications are placed in the front of the car, typically behind the grille. The radar emits radio millimeter waves, and distance from the vehicle ahead is calculated by measuring the arrival time and direction of the reflected radio waves. The relative velocity is determined by relying on the Doppler effect, i.e., by measuring the frequency change of the



Lane Keeping Systems, Fig. 1 The lane keeping architecture

reflected waves. Relative distance and velocity measurements are typically updated with a frequency of 10–15 Hz.

Radars for automotive applications emit waves within a frequency range of 24–77 GHz and detect objects within an approximate range of 150 m and a view angle of about $\pm 10^\circ$, with a deviation of 20–30 cm from the correct value for 95% of the measurements (Eidehall 2004). New radar systems support the *multimode* operation, increasing the range up to about 200 m with a view angle of about $\pm 10^\circ$ (News Releases Denso Corporation 2013a). In particular, *multimode* radars can switch between two modes, namely, one with shorter range and wider view angle and the other with longer range and narrower view angle.

Typically, radar units are equipped with computer systems running signal processing algorithms that detect and track objects and, for each of them, calculate relative position and speed and azimuth angle and also provide additional information like the time an object has been tracked and a flag indicating that a target has been locked. Such additional information are typically used in implementing the decision-making algorithms of, among others, the lane keeping system.

There are several issues arising from the use of a radar in automotive applications like wave reflections due to road bumps or barriers that may induce the signal processing algorithms to false object detections (Eidehall 2004). Moreover, interference and the vehicle dynamics (News Releases Denso Corporation 2013a) like pitching due to braking may limit the capability of the signal processing algorithms of correctly detecting and tracking the surrounding objects. The latter may be solved by, e.g., using electric motors that adjust the radar antenna axes in order to compensate for the vehicle dynamics (News Releases Denso Corporation 2013a).

Vision Systems

Vision systems in lane keeping applications are typically based on a single, CCD camera mounted next to the rear view mirror placed at the center of the windshield. The image

is typically captured by 640 times 480 pixels and then processed by an image processing unit. The sampling time of the vision system is about 0.1 s, but it can change depending on, e.g., the complexity of the scene, for example, in city traffic (Eidehall 2004). For instance, sampling times lower than 0.1 s are used in LKA applications.

Lane markings are detected by using differences in the image contrast (Technology Daimler and Safety Innovation 2013). The camera can be either monochrome or full color. The latter is used to enhance the detection of lane markings, which have different colors around the world (News Releases Denso Corporation 2013b). Distances to the lane markings and road geometry parameters like heading angle and curvature are determined by the image processing algorithms, which must be robust to bad weather conditions or worn lane markings. Estimation of road geometry parameters like curvature measurement can be a challenging problem (Lundquist and Schön 2011), especially with rain or fog (Eidehall 2004).

Depending on the image processing algorithms the cameras are equipped with, surrounding objects can also be detected and tracked. In particular, pattern recognition algorithm can be used to find objects in the images and classify them into cars, trucks, motorcycles, and pedestrians. Vehicles (or other objects) can be typically detected in a range of about 60–70 m, with lower accuracy than a radar (Eidehall 2004).

Actuators

In order to keep the vehicle within its lane, the most convenient actuator is the steering. Hence, a lane keeping system can be quite easily built in those vehicles equipped with Electric Power Assist Steering (EPAS) systems. In particular, an additional steering torque can be added by the EPAS to the driver's steering torque, in order to generate the desired yaw moment calculated by the *decision-making and control module*.

Clearly, the steering command is not the only one available that affects the vehicle yaw motion, thus changing its orientation and lateral position within the lane. Individual wheel braking

may also be used (Technology Daimler and Safety Innovation 2013). In particular, in vehicles equipped with yaw motion control system via individual braking, a braking torque request for each wheel can be sent to the yaw motion control system in order to generate the desired yaw moment.

Decision-Making and Control

The decision-making and control in a lane keeping problem can be conceptually divided into two tasks: *the threat assessment* and *the lane position control*. The threat assessment problem can be stated as the problem of detecting the risk of accident due to an unintended lane departure for a given situation of the surrounding environment (i.e., surrounding vehicles and obstacles). The lane position control problem is the problem of controlling the vehicle yaw and lateral motion in order to stay within the lane. The lane position control is activated once the threat assessment detects the risk of accident.

We point out that the border between the corresponding modules executing these two tasks may be blur for different existing commercial lane keeping systems, i.e., the two problems cannot be solved by two separate modules, but rather can be seen and solved as a single problem. Moreover, the following presentation of the threat assessment and the lane position control problems and approaches abstracts from the implementation of a particular lane keeping system available on the market, rather than focusing on fundamental concepts.

As a last remark, it should be noticed that *override functionalities* are always present in LKSs, to immediately return the control of the vehicle to the driver. Typically, the driver's steering commands resulting in large steering torques are not counteracted by the system, which rather lets the driver gain the control of the vehicle.

Threat Assessment

The core information in a threat assessment algorithm for lane keeping applications is given by a measure called Time to Lane Crossing (TLC).

This is the predicted time when a front tire intersects a lane boundary. As explained in van Winsum et al. (2000), the TLC can be calculated in different ways. Next, its simplest expression is reported (Eidehall 2004):

$$\text{TLC} = \frac{W/2 - W_{\text{veh}}/2 - y_{\text{off}}}{\dot{y}_{\text{off}}} \quad (1)$$

where W is the lane width, y_{off} is the vehicle lateral position within the lane, and W_{veh} is the vehicle width. Equation (1) can be easily modified to calculate the TLC w.r.t. any lane boundary relative to the adjacent lanes.

The simplest way of using the TLC is just monitoring it and triggering an action as the TLC passes a threshold. Nevertheless, depending on the vehicle manufacturer, more sophisticated algorithms can be developed in order to correctly interpret the driver's intention and minimize unnecessary assisting interventions. Next, a few scenarios follow that must be taken into account while developing such algorithms in order not to interfere with the driver. In particular, the threat assessment module should stop or not trigger any assisting intervention while the vehicle is approaching or crossing a lane boundary if

- the indicators are active;
- a risk of collision with the preceding vehicle is detected, such that the vehicle is crossing the lane markings as results of an evasive maneuver;
- the radar detects a slower vehicle ahead and the driver accelerates, since this may be an overtaking (Technology Daimler and Safety Innovation 2013);
- the driver's steering wheel torque indicates that the driver is acting against the system;
- the driver manually initiates a maneuver driving the vehicle back to its lane (i.e., the driver executes "the correct" maneuver);
- the vehicle enters a motorway or a bend (Technology Daimler and Safety Innovation 2013).

Part of the threat assessment task is predicting the trajectories of the surrounding vehicles. For instance, if a *threat* vehicle is traveling in the

adjacent lane (in the same or opposite direction), its position has to be predicted at the TLC in order to decide whether to trigger an intervention, if a collision is predicted, or not (Eidehall 2004). This step is repeated for all the detected threat vehicles, provided that the onboard radar and the camera support multiple target tracking.

In order to minimize the interference of the lane keeping system with the driver and/or to not let the system perform dangerous maneuvers, assisting interventions should not be triggered if the quality of the measurements is such that the information about the surrounding environment is poor. For instance, in case of low visibility that limits the detection of the lane markings and the estimation of the road geometry, the system should be temporarily deactivated or downgraded.

In summary, the threat assessment module has to be designed with the objective of detecting the risk of accident due to lane departure while not interfering with the driver with unnecessary interventions (i.e., nuisance minimization).

Lane Position Control

As observed in section “Actuators”, the vehicle motion within the lane can be affected in two ways, i.e., through steering and individual wheel braking. Clearly, a steering command can be issued by both the driver and the lane keeping system.

Before issuing a steering command, in order to minimize the system nuisance, the lane keeping system can issue other types of low-intrusiveness interventions. For instance, if a “low”-level threat is detected by the threat assessment module (i.e., a threat where the risk of accidents is not imminent), warnings or other stimuli to the driver can be issued in order to induce the driver to execute the right maneuver. For instance, based on, e.g., spectrum analysis of the driver’s steering command, driver’s inattention or drowsiness may be detected and a warning issued. As observed in Technology Daimler and Safety Innovation (2013), different types of warning can be used for different vehicle types ranging from a vibration motor in the steering wheel to a sound or a vibrating seat for passenger vehicles. In trucks,

audible, directional warning signals can be used to let the driver know that the vehicle trajectory needs to be adjusted. In buses, in order to avoid bothering the passengers, driver warning is issued through vibration motors placed in the driver’s seat.

Other types of “soft intervention” aim at increasing the steering impedance in the direction leading to lane crossing that might cause a collision with surrounding vehicles. Generating the desired steering impedance can be easily formulated as a steering torque control problem. Nevertheless, tuning the control algorithm to obtain the desired steering feeling can be an involving and time-consuming procedure based on extensive in-vehicle testing.

Besides warnings and “soft interventions” aiming at inducing the driver to perform correct maneuvers, as part of the lane position control task in a lane keeping system, a lateral control algorithm w.r.t. the lane boundaries is needed. Consider the vehicle sketched in Fig. 2. The equations describing the vehicle motion within the lane can be compactly written in a state-space form as follows:

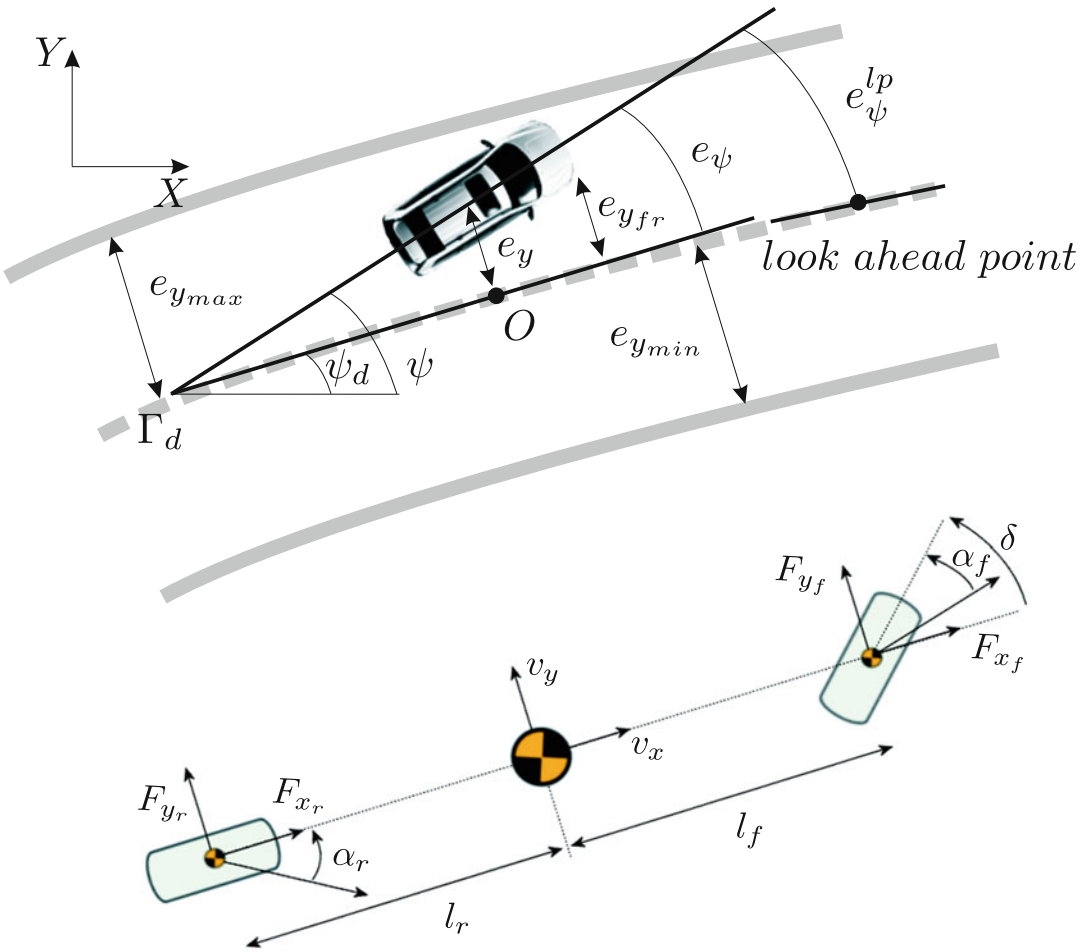
$$\dot{x} = Ax + B\delta + D\dot{\psi}_{\text{des}} \quad (2)$$

where $x = [e_y \ \dot{e}_y \ e_\psi \ \dot{e}_\psi]$ is the vector of lateral position and yaw errors and error rates, respectively; $\dot{\psi}_{\text{des}}$ is the desired yaw rate, e.g., calculated based on the road curvature; and A , B , and D are speed-dependent matrices that can be found in Rajamani (2006). The (unstable) system can be stabilized by a state feedback control law:

$$\delta = -Kx + \delta_{\text{ff}} \quad (3)$$

where K is a stabilizing static gain and δ_{ff} is a feedforward term that can be used to compensate for the road curvature. In Rajamani (2006), it is shown that, while $e_y(t) \rightarrow 0$ as $t \rightarrow \infty$, e_ψ approaches a nonzero steady-state value, no matter how δ_{ff} is chosen, for non-straight road.

Despite a simple problem formulation and solution, controlling the vehicle position within the lane is not a trivial task. Indeed, having the



Lane Keeping Systems, Fig. 2 Vehicle modeling notation

control law (3) active all time may increase the nuisance leading to unacceptable driving experience. For this reason, the steering command calculated through Eq.(3) may be active only when the vehicle significantly deviates from the road centerline, i.e., approaches the lane markings. Clearly, adding such conditions complicates the analysis of the closed-loop behavior, thus making necessary extensive in-vehicle tuning and verification.

Cross-References

- [Image-Based Robot Control](#)
- [Robot Visual Control](#)

Acknowledgments The author would like to thank Dr. Erik Coelingh from Volvo Car Corporation for helpful discussions.

Bibliography

- Cars V (2017) Blind spot information blis with steer assist
- Cars V (2019) Collision avoidance mitigation
- Eidehall A (2004) An automotive lane guidance system. Lic Thesis 1122, Linköping University
- Lundquist C, Schön TB (2011) Joint ego-motion and road geometry estimation. Inf Fusion 4(12):253–263
- Mercedes (2013) Active lane keeping assist
- News Releases Denso Corporation (2013a) Denso develops higher performance millimeter-wave radar, June 2013
- News Releases Denso Corporation (2013b) Denso develops new vision sensor for active safety systems, June 2013

- Rajamani R (2006) *Vehicle dynamics and control*, chap 2. Springer, New York
- Technology Daimler and Safety Innovation (2013) Lane keeping assist: always on the right track, June 2013
- van Winsum W, Brookhuis KA, de Waard D (2000) A comparison of different ways to approximate time-to-line crossing (TLC) during car driving. *Accid Anal Prev* 32(1):47–56

Learning Control of Quantum Systems

Daoyi Dong
School of Engineering and Information
Technology, University of New South Wales,
Canberra, ACT, Australia

Abstract

This chapter presents a brief introduction to learning control of quantum systems. Gradient-based learning methods, evolutionary computation algorithms, and reinforcement learning approaches are outlined for searching optimal or robust fields in quantum control problems. The state-of-the-art methods and future directions in learning control of quantum systems are discussed.

Keywords

Quantum control · Quantum learning
control · Quantum optimal control ·
Quantum robust control · Quantum systems

Introduction

Controlling quantum systems has become a central task in the development of quantum technologies, and quantum control has witnessed rapid progress in the last two decades; for an overview, see, e.g., the survey papers (Dong and Petersen 2010; Rabitz et al. 2000; Glaser et al. 2015; Brif et al. 2010; Altafini and Ticozzi 2012) or the monographs (Wiseman and Milburn 2010; D'Alessandro 2007). The general goal of

quantum control is to actively manipulate and control the dynamics of quantum systems for achieving given objectives (Acín et al. 2018; Guo et al. 2019) (e.g., rapid state transfer, high-fidelity gate operation). Two of fundamental issues in quantum control include investigating controllability of quantum systems and designing control laws to achieve expected control systems performance. Controllability is concerned with what control targets can be achieved and the controllability of finite-dimensional closed systems has been well addressed (D'Alessandro 2007). A few results on the controllability of open quantum systems have also been presented. For control law design, optimal control theory (Dong and Petersen 2010), Lyapunov control approaches (Kuang et al. 2017), learning control algorithms (Rabitz et al. 2000), and robust control methods (Dong et al. 2020) have been developed in manipulating quantum systems for achieving various control objectives.

Among various control design approaches, learning control is recognized as a powerful method for many complex quantum control tasks and has achieved great success in laser control of molecules and other applications since the approach was presented in the seminal paper (Judson and Rabitz 1992). Many quantum control tasks may be formulated as an optimization problem, and a learning algorithm can be employed to search for an optimal control field satisfying a desired performance condition. Gradient algorithms have been demonstrated to be an excellent candidate for numerically finding an optimal field and have achieved successful applications in nuclear magnetic resonance (NMR) systems due to their high efficiency (Khaneja et al. 2005). In many other optimal control problems, the gradient information may not be easy to obtain, and some complex problems may have local optima. For these situations, stochastic search algorithms usually have improved performance to find a good control field. The genetic algorithm (GA) and differential evolution (DE) (Dong et al. 2020) have been widely used in the area of quantum control of molecular systems and achieved great success (Rabitz et al. 2000). Another task

in quantum control is to achieve robustness performance in quantum systems. Gradient-based learning algorithms and stochastic search algorithms may be smartly modified to search for robust control fields. Also, some other machine learning algorithms such as reinforcement learning have found successful applications in various tasks (e.g., quantum error correction Fösel et al. 2018).

Gradient-Based Learning for Optimal Control of Quantum Systems

Consider a finite-dimensional quantum control system where its state $|\psi(t)\rangle$ (using the Dirac notation) is described by the following Schrödinger equation (setting $\hbar = 1$):

$$\begin{aligned} \frac{d}{dt}|\psi(t)\rangle &= -i[H_0 + \sum_{m=1}^M u_m(t)H_m]|\psi(t)\rangle, \quad t \in [0, T], \end{aligned} \quad (1)$$

where H_0 is the free Hamiltonian of the system and $H_c(t) = \sum_{m=1}^M u_m(t)H_m$ is the control Hamiltonian at time t that represents the interaction of the system with the external fields $u_m(t)$. The H_m are Hermitian operators through which the controls couple to the system. The objective of quantum optimal control is to find control fields $u_m(t)$ for maximizing a performance functional Φ . Φ may be a given functional of the state $|\psi\rangle$ and control defined according to practical requirement. For example, the fidelity $\Phi = |\langle\psi(T)|\psi_f\rangle|^2$ between the final state $|\psi(T)\rangle$ and target state $|\psi_f\rangle$ or the expectation $\Phi = |\langle\psi(T)|\hat{O}|\psi(T)\rangle|^2$ of an operator $\hat{O}$ may be defined as a performance index for state transfer task. In order to maximize the performance Φ , we may employ the GRAPE (gradient ascent pulse engineering) algorithm (Khaneja et al. 2005) or the Krotov method (Jäger et al. 2014) to search for the control field. For simplicity, we may discretize time T in N equal steps and during each step let the control fields u_m be

constant. The basic idea in the GRAPE algorithm is that the control fields are iteratively updated following the gradient direction of $\frac{\delta\Phi}{\delta u_m(k)}$ with a learning rate η , i.e.,

$$u_m(k+1) = u_m(k) + \eta \frac{\delta\Phi}{\delta u_m(k)}. \quad (2)$$

The gradient-based learning method can also be extended to the optimal control problem of unitary transformations (e.g., quantum gates) and open quantum systems. For example, for a unitary transformation U , its evolution is described by the following equation:

$$\dot{U}(t) = -i[H_0 + \sum_{m=1}^M u_m(t)H_m]U(t), \quad U(0) = I. \quad (3)$$

Now the objective is to design the controls $u_m(t)$ to steer the unitary $U(t)$ from $U(0) = I$ to a desired target U_F with high fidelity. We may define the performance function as $\Phi = |\langle U_F | e^{i\varphi} U(T) \rangle|^2$ for an arbitrary phase factor φ . Then we can calculate the gradient $\delta\Phi/\delta u_m(k)$, and the optimal control field can be searched for by following the gradient (Dong et al. 2016). When we consider the optimal control problem of an open quantum system, its state should be represented by a density matrix ρ and its dynamics should be described by a master equation. The dynamics of a Markovian open quantum system can be described using the following master equation in the Lindblad form as (Wiseman and Milburn 2010)

$$\begin{aligned} \dot{\rho}(t) &= -i[H_0 + \sum_{m=1}^M u_m(t)H_m, \rho(t)] \\ &+ \sum_k \gamma_k \mathcal{D}[L_k]\rho(t), \end{aligned} \quad (4)$$

where the non-negative coefficients γ_k specify the relevant relaxation rates, L_k are appropriate Lindblad operators, and $\mathcal{D}[L_k]\rho = (L_k\rho L_k^\dagger - \frac{1}{2}L_k^\dagger L_k\rho - \frac{1}{2}\rho L_k^\dagger L_k)$. The open GRAPE algorithm has also been developed to calculate the gradient based on the master equation (see Schulte-Herbrüggen et al. 2011).

Using the basic idea of gradient-based learning control, some variants have been developed for various requirements in quantum optimal control. For example, a data-driven gradient optimization algorithm (d-GRAPE) has been proposed to correct deterministic gate errors in high-precision quantum control by jointly learning from a design model and the experimental data from quantum tomography (Wu et al. 2018). A gradient-based frequency-domain optimization algorithm has been developed to solve the optimal control problem with constraints in the frequency domain (Shu et al. 2016). Existing results show that gradient-based learning methods can usually achieve excellent performance for solving optimal control problems when the system model is known and the dynamics can be equivalently (or approximately) described using a closed quantum system. This is also analyzed using quantum control landscape theory (Chakrabarti and Rabitz 2007).

Evolutionary Computation for Learning Control of Quantum Systems

Gradient algorithms have shown powerful capability for numerically finding optimal controls due to their excellent performance (Khaneja et al. 2005). In many practical applications, it may be difficult to obtain the gradient information or there exist local optima in complex quantum control problems. For these situations, a natural idea is to employ stochastic search algorithms to seek good controls. Evolutionary computation including GA and DE has been widely used in the area of quantum control. In these evolutionary computation methods, crossover, mutation, and selection operations are iteratively implemented to search for good solutions (optimal controls) in a parameter space. For example, a subspace-selective self-adaptive differential evolution (SUSSADE) algorithm has been proposed to achieve a high-fidelity single-shot Toffoli gate and single-shot three-qubit gates (Zahedinejad et al. 2015, 2016). Existing results showed that

DE with equally mixed strategies can achieve improved performance for quantum control problems (Ma et al. 2017). Several promising evolution algorithms have been investigated comparatively in Zahedinejad et al. (2014), and it was found that DE usually outperformed GA and particle swarm optimization for hard quantum control problems.

The above introduction of gradient-based learning and evolutionary computation mainly involves open-loop control strategies. Evolutionary computation has demonstrated extremely powerful capability when it is integrated into closed-loop control design. Closed-loop learning control, where each cycle of the closed loop is executed on a new sample, has achieved great successes in the laser control of laboratory chemical reactions (Rabitz et al. 2000; Judson and Rabitz 1992). A closed-loop learning control procedure generally involves three components (Dong and Petersen 2010; Rabitz et al. 2000): (i) a trial laser control input, (ii) the laboratory generation of the control that is applied to the sample and subsequently observed for its impact, and (iii) a learning algorithm to suggest the form of the next control input by considering the prior experiments. The initial trial control input may be a random input field or a well-designed laser pulse. A feature of a good closed-loop learning control design is its insensitivity to the initial trials. A key task is to develop a good learning algorithm for ensuring that the learning process converges to achieve a predetermined objective. GA, DE, and several rapid convergence algorithms have been developed for this task (Brif et al. 2010; Judson and Rabitz 1992). The optimal control problem is usually formulated as solving an optimization problem by maximizing a functional which is related to some variables such as the control inputs, quantum states, and control time but may have no analytical form. In the learning process, the optimization problem is solved iteratively. First, a trial input is applied to a sample to be controlled and the result is observed. Second, a learning algorithm suggests a better control input based on the prior experiments. Third, the “better” control input is applied to a new sample. This process continues until the

control objective is achieved or the maximum permitted iteration number is reached. It is often feasible to produce many identical-state samples for laboratory chemical molecules. If the control objective is well selected, there is a capability to apply specified control inputs to the samples, and the learning algorithm is sufficiently smart for searching for good control inputs, this process will converge and an optimal control pulse can be found (Rabitz et al. 2000).

Learning-Based Quantum Robust Control

The robust control of quantum systems has been recognized as a key task in developing practical quantum technology since the existence of noise and uncertainties is unavoidable. Learning control is an effective candidate for achieving robust performance in some quantum control problems (Dong et al. 2020). We first consider the control problem of inhomogeneous quantum ensembles. An inhomogeneous quantum ensemble consists of many individual quantum systems (e.g., atoms, molecules, or spin systems), and the parameters describing the system dynamics of these individual systems could have variations (Li and Khaneja 2006; Chen et al. 2014a). An example is that a spin ensemble in NMR may encounter large dispersion in the strength of the applied radio-frequency field and there also exist variations in the natural frequencies of these spins (Li and Khaneja 2006). Inhomogeneous quantum ensembles have wide applications in many fields ranging from quantum memory to magnetic resonance imaging. Hence, it is highly desirable to design control laws for an inhomogeneous ensemble to employ the same control inputs to steer individual systems with different dynamics from a given initial state to a target state.

A sampling-based learning control (SLC) method has been developed to achieve high-fidelity control of inhomogeneous quantum ensembles (Chen et al. 2014a). Consider an inhomogeneous ensemble in which the Hamiltonian of each individual system has the following form:

$$H_{\omega,\theta}(t) = \omega H_0 + \sum_{m=1}^M \theta u_m(t) H_m. \quad (5)$$

We assume that the parameters $\omega \in [1 - \Omega, 1 + \Omega]$ and $\theta \in [1 - \Theta, 1 + \Theta]$ and the constants $\Omega \in [0, 1]$ and $\Theta \in [0, 1]$ represent the bounds of the parameter dispersion. The objective is to design the controls $\{u_m(t)\}$ to simultaneously stabilize the individual systems (with different ω and θ) of the quantum ensemble from an initial state $|\psi_0\rangle$ to the same target state $|\psi_f\rangle$ with high fidelity. This task can be achieved using the SLC method including two steps of “training” and “testing and evaluation” (Chen et al. 2014a). In the training step, we select N samples from the quantum ensemble regarding the distribution (e.g., uniform distribution) of the inhomogeneity parameters and then construct a generalized system as follows:

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} |\psi_{\omega_1, \theta_1}(t)\rangle \\ |\psi_{\omega_2, \theta_2}(t)\rangle \\ \vdots \\ |\psi_{\omega_N, \theta_N}(t)\rangle \end{pmatrix} \\ = -i \begin{pmatrix} H_{\omega_1, \theta_1}(t) |\psi_{\omega_1, \theta_1}(t)\rangle \\ H_{\omega_2, \theta_2}(t) |\psi_{\omega_2, \theta_2}(t)\rangle \\ \vdots \\ H_{\omega_N, \theta_N}(t) |\psi_{\omega_N, \theta_N}(t)\rangle \end{pmatrix} \end{aligned} \quad (6)$$

where $H_{\omega_n, \theta_n} = \omega_n H_0 + \sum_m \theta_n u_m(t) H_m$ with $n = 1, 2, \dots, N$. The cost function for the generalized system is defined by

$$\Phi_N(u) := \frac{1}{N} \sum_{n=1}^N \Phi_{\omega_n, \theta_n}(u). \quad (7)$$

The task of the training step is to find a control strategy u^* to maximize the cost functional $\Phi_N(u)$. A gradient-based learning algorithm (s-GRAPE) can be developed to complete this task. In the process of testing and evaluation, a number of sampling individual systems are randomly selected to evaluate the control performance. Results show that the SLC method is potential for control design of various inhomogeneous

quantum ensembles (including inhomogeneous open quantum ensembles).

Besides inhomogeneous quantum ensembles, the SLC method is useful for robust control of single quantum systems with various uncertainties. For example, Eq. (5) can also correspond to the Hamiltonian of a quantum system with inaccurate model parameter ω and uncertain multiplicative noise θ . In order to achieve robust control for such a quantum system, we may employ the SLC method to search for robust control pulses (Dong et al. 2015a, b; Wu et al. 2017). The performance of SLC approach can be further improved by exploring the richness and diversity of samples. Inspired by deep learning, a batch-based gradient algorithm (b-GRAPE) has been presented for efficiently seeking robust quantum controls, and numerical results showed that b-GRAPE can achieve improved performance over the SLC method for remarkably enhancing the control robustness while maintaining high fidelity (Wu et al. 2019). In other applications where we need to enhance the robustness in closed-loop learning control, we may either use the Hessian matrix information (Xing et al. 2014) or integrate the idea of SLC into the learning algorithm in searching for robust control fields. For example, an improved DE algorithm (called as *msMS_DE*) has been proposed to search for robust femtosecond laser pulses to control fragmentation of the molecule CH_2BrI (Dong et al. 2020). In *msMS_DE*, multiple samples are used for fitness evaluation and a mixed strategy is employed for the mutation operation.

Reinforcement Learning for Quantum Control

Reinforcement learning (RL) (Sutton and Barto 1998) is another important machine learning approach, and it addresses the problem of how an active agent can learn to approximate an optimal strategy while interacting with its environment. It is a model-free feedback-based approach and works well even when the system model is unknown or with uncertainties. RL has been used for learning control of quantum systems.

For example, a fidelity-based probabilistic Q-learning approach has been presented to naturally solve the balance problem between exploration and exploitation and was applied to learning control of quantum systems (Chen et al. 2014b). The authors in Bukov et al. (2018) showed that the performance of RL is comparable to optimal control approaches in the task of finding a short and high-fidelity protocol, controlling from an initial to a given target state in nonintegrable many-body quantum systems of interacting qubits. RL can also help identify variational protocols with nearly optimal fidelity even in the glassy phase. In Niu et al. (2019), deep reinforcement learning is employed to simultaneously optimize the speed and fidelity of quantum computation against both leakage and stochastic control errors. A universal quantum control framework was presented to improve the control robustness by adding control noise into training environments for RL agents trained with trusted region policy optimization. By utilizing two-stage learning with teacher and student networks and a reward quantifying the capability to recover the quantum information stored in a quantum system, the authors in Fösel et al. (2018) showed how a network-based “agent” in RL can discover good quantum error correction strategies to protect qubits against noise.

Conclusions

Machine learning has shown powerful capability in discovering high-quality controls to achieve optimal control and enhance robust performance for quantum systems. This chapter briefly introduces typical applications of machine learning to quantum control and illustrates how learning algorithms can assist in solving control problems of quantum systems with improved performance.

Summary and Future Directions

Learning control has been one of the main control design methods for quantum systems. Gradient-based methods have shown their efficiency advantages in numerically solving

quantum optimal and robust control problems. Evolutionary learning approaches including genetic algorithms and differential evolution algorithms have potential for control of open quantum systems and laboratory quantum control design. Reinforcement learning may provide an effective method for solving quantum control problems with feedback.

Future directions of quantum learning control include the following: (1) to further develop or improve existing machine learning algorithms to effectively solve complex (especially high-dimensional) quantum control problems emerged from new quantum technologies; (2) to explore various cutting-edge machine learning techniques (e.g., deep learning) for applications in the area of quantum control; (3) to integrate machine learning algorithms into quantum control experimental design for saving expensive quantum resources (i.e., quantum entanglement and quantum coherence) and discovering quantum mechanism; and (4) to develop novel quantum machine learning algorithms for simulating complex quantum control systems.

Cross-References

- [Application of Systems and Control Theory to Quantum Engineering](#)
- [Control of Quantum Systems](#)
- [Learning in Games](#)
- [Learning Theory: The Probably Approximately Correct Framework](#)

Bibliography

- Acín A, Bloch I, Buhrman H, Calarco T, Eichler C, Eisert J et al (2018) The quantum technologies roadmap: a European community view. *New J Phys* 20:080201
- Altafini C, Ticozzi F (2012) Modeling and control of quantum systems: an introduction. *IEEE Trans Autom Control* 57(8):1898–1917
- Brif C, Chakrabarti R, Rabitz H (2010) Control of quantum phenomena: past, present and future. *New J Phys* 12:075008
- Bukov M, Day AGR, Sels D, Weinberg P, Polkovnikov A, Mehta P (2018) Reinforcement learning in different phases of quantum control. *Phys Rev X* 8:031086

- Chakrabarti R, Rabitz H (2007) Quantum control landscapes. *Int Rev Phys Chem* 26(4):671–735
- Chen C, Dong D, Long R, Petersen IR, Rabitz HA (2014a) Sampling-based learning control of inhomogeneous quantum ensembles. *Phys Rev A* 89(2):023402
- Chen C, Dong D, Li HX, Chu J, Tam TJ (2014b) Fidelity-based probabilistic Q-learning for control of quantum systems. *IEEE Trans Neural Netw Learn Syst* 25:920–933
- D'Alessandro D (2007) Introduction to quantum control and dynamics. Chapman & Hall/CRC, Boca Raton
- Dong D, Petersen IR (2010) Quantum control theory and applications: a survey. *IET Control Theory Appl* 4(12):2651–2671
- Dong D, Mabrok MA, Petersen IR, Qi B, Chen C, Rabitz H (2015a) Sampling-based learning control for quantum systems with uncertainties. *IEEE Trans Control Syst Technol* 23:2155–2166
- Dong D, Chen C, Qi B, Petersen IR, Nori F (2015b) Robust manipulation of superconducting qubits in the presence of fluctuations. *Sci Rep* 5:7873
- Dong D, Wu C, Chen C, Qi B, Petersen IR, Nori F (2016) Learning robust pulses for generating universal quantum gates. *Sci Rep* 6:36090
- Dong D, Xing X, Ma H, Chen C, Liu Z, Rabitz H (2020) Learning-based quantum robust control: algorithm, applications and experiments. *IEEE Trans Cybern.* <https://doi.org/10.1109/TCYB.2019.2921424>, online: <https://ieeexplore.ieee.org/abstract/document/8759071>
- Fösel T, Tighineanu P, Weiss T, Marquardt F (2018) Reinforcement learning with neural networks for quantum feedback. *Phys Rev X* 8:031084
- Glaser SJ, Boscain U, Calarco T, Koch CP, Köckenberger W, Kosloff R, Kuprov I, Luy B, Schirmer S, Schulte-Herbrüggen T, Sugny D, Wilhelm FK (2015) Training Schrödinger's cat: quantum optimal control. *Eur Phys J D* 69:279
- Guo Y, Shu CC, Dong D, Nori F (2019) Vanishing and revival of resonance Raman scattering. *Phys Rev Lett* 123:223202
- Jäger G, Reich DM, Goerz MH, Koch CP, Hohenester U (2014) Optimal quantum control of Bose-Einstein condensates in magnetic microtraps: comparison of gradient-ascent-pulse-engineering and Krotov optimization schemes. *Phys Rev A* 90:033628
- Judson RS, Rabitz H (1992) Teaching lasers to control molecules. *Phys Rev Lett* 68:1500–1503
- Khaneja N, Reiss T, Kehlet C, Schulte-Herbrüggen T, Glaser SJ (2005) Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. *J Magn Reson* 172(2):296–305
- Kuang S, Dong D, Petersen IR (2017) Rapid Lyapunov control of finite-dimensional quantum systems. *Automatica* 81:164–175
- Li JS, Khaneja N (2006) Control of inhomogeneous quantum ensembles. *Phys Rev A* 73(3):030302
- Ma H, Dong D, Shu C-C, Zhu Z, Chen C (2017) Differential evolution with equally-mixed strategies for robust control of open quantum systems. *Control Theory Technol* 15:226–241

- Niu MY, Boixo S, Smelyanskiy VN, Neven H (2019) Universal quantum control through deep reinforcement learning. *npj Quantum Inf* 5:33
- Rabitz H, De Vivie-Riedle R, Motzkus M, Kompa K (2000) Whither the future of controlling quantum phenomena? *Science* 288(5467):824–828
- Schulte-Herbrüggen T, Spörl A, Khaneja N, Glaser SJ (2011) Optimal control for generating quantum gates in open dissipative systems. *J Phys B Atomic Mol Opt Phys* 44:154013
- Shu CC, Ho TS, Xing X, Rabitz H (2016) Frequency domain quantum optimal control under multiple constraints. *Phys Rev A* 93:033417
- Sutton R, Barto AG (1998) Reinforcement learning: an introduction. MIT Press, Cambridge
- Wiseman HM, Milburn GJ (2010) Quantum measurement and control. Cambridge University Press, Cambridge
- Wu C, Qi B, Chen C, Dong D (2017) Robust learning control design for quantum unitary transformations. *IEEE Trans Cybern* 47:4405–4417
- Wu RB, Chu B, Owens DH, Rabitz H (2018) Data-driven gradient algorithm for high-precision quantum control. *Phys Rev A* 97:042122
- Wu R, Ding H, Dong D, Wang X (2019) Learning robust and high-precision quantum controls. *Phys Rev A* 99:042327
- Xing X, Rey-de-Castro R, Rabitz H (2014) Assessment of optimal control mechanism complexity by experimental landscape Hessian analysis: fragmentation of CH₂BrI. *New J Phys* 16:125004
- Zahedinejad E, Schirmer S, Sanders BC (2014) Evolutionary algorithms for hard quantum control. *Phys Rev A* 90(3):032310
- Zahedinejad E, Ghosh J, Sanders BC (2015) High-fidelity single-shot Toffoli gate via quantum control. *Phys Rev Lett* 114(20):200502
- Zahedinejad E, Ghosh J, Sanders BC (2016) Designing high-fidelity single-shot three-qubit gates: a machine-learning approach. *Phys Rev Appl* 6:054005

Learning in Games

Jeff S. Shamma

School of Electrical and Computer Engineering,
Georgia Institute of Technology, Atlanta,
GA, USA

Abstract

In a Nash equilibrium, each player selects a strategy that is optimal with respect to the strategies of other players. This definition does not mention the process by which players

reach a Nash equilibrium. The topic of learning in games seeks to address this issue in that it explores how simplistic learning/adaptation rules can lead to Nash equilibrium. This entry presents a selective sampling of learning rules and their long-run convergence properties, i.e., conditions under which player strategies converge or not to Nash equilibrium.

Keywords

Cournot best response · Fictitious play · Log-linear learning · Mixed strategies · Nash equilibrium

Introduction

In a *Nash equilibrium*, each player's strategy is optimal with respect to the strategies of other players. Accordingly, Nash equilibrium offers a predictive model of the outcome of a game. That is, given the basic elements of a game – (i) a set of players; (ii) for each player, a set of strategies; and (iii) for each player, a utility function that captures preferences over strategies – one can model/assert that the strategies selected by the players constitute a Nash equilibrium.

In making this assertion, there is no suggestion of *how* players may come to reach a Nash equilibrium. Two motivating quotations in this regard are:

The attainment of equilibrium requires a disequilibrium process (Arrow 1986).

and

The explanatory significance of the equilibrium concept depends on the underlying dynamics (Skyrms 1992).

These quotations reflect that a foundation for Nash equilibrium as a predictive model is dynamics that lead to equilibrium. Motivated by these considerations, the topic of “learning in games” shifts the attention away from equilibrium and towards underlying dynamic processes and their long-run behavior. The intent is to understand how players may reach an equilibrium as well

as understand possible barriers to reaching Nash equilibrium.

In the setup of learning in games, players repetitively play a game over a sequence of stages. At each stage, players use past experiences/observations to select a strategy for the current stage. Once player strategies are selected, the game is played, information is updated, and the process is repeated. The question is then to understand the long-run behavior, e.g., whether or not player strategies converge to Nash equilibrium.

Traditionally the dynamic processes considered under learning in games have players selecting strategies based on a myopic desire to optimize for the current stage. That is, players do not consider long-run effects in updating their strategies. Accordingly, while players are engaged in repetitive play, the dynamic processes generally are not optimal in the long run (as in the setting of “repeated games”). Indeed, the survey article of Hart (2005) refers to the dynamic processes of learning in games as “adaptive heuristics.” This distinction is important in that an implicit concern in learning in games is to understand how “low rationality” (i.e., suboptimal and heuristic) processes can lead to the “high rationality” (i.e., mutually optimal) notion of Nash equilibrium.

This entry presents a sampling of results from the learning in games literature through a selection of illustrative dynamic processes, a review of their long-run behaviors relevant to Nash equilibrium, and pointers to further work.

Illustration: Commuting Game

We begin with a description of learning in games in the specific setting of the commuting game, which is a special case of so-called congestion games (cf., Roughgarden 2005). The setup is as follows. Each player seeks to plan a path from an origin to a destination. The origins and destinations can differ from player to player. Players seek to minimize their own travel times. These travel times depend both on the chosen path (distance traveled) and the paths of other players (road congestion). Every day, a player uses past

information and observations to select that day’s path according to some selection rule, and this process is repeated day after day.

In game-theoretic terms, player “strategies” are paths linking their origins to destinations, and player “utility functions” reflect travel times. At a Nash equilibrium, players have selected paths such that no individual player can find a shorter travel time given the chosen paths of others. The learning in games question is then whether player paths indeed converge to Nash equilibrium in the long run. Not surprisingly, the answer depends on the specific process that players use to select paths and possible additional structure of the commuting game.

Suppose that one of the players, say “Alice,” is choosing among a collection of paths. For the sake of illustration, let us give Alice the following capabilities: (i) Alice can observe the paths chosen by all other players and (ii) Alice can compute off-line her travel time as a function of her path and the paths of others.

With these capabilities, Alice can compute running averages of the travel times along all available paths. Note that the assumed capabilities allow Alice to compute the travel time of a path and hence its running average, *whether or not* she took the path on that day. With average travel time values in hand, two possible learning rules are:

- *Exploitation*: Choose the path with the lowest average travel time.
- *Exploitation with Exploration*: With high probability, choose the path with the lowest average travel time, and with low probability, choose a path at random.

Assuming that all players implement the same learning rule, each case induces a dynamic process that governs the daily selection of paths and determines the resulting long-run behavior. We will revisit these processes in a more formal setting in the next section.

A noteworthy feature of these learning rules is that they do not explicitly depend on the utility functions of other players. For example, suppose one of the other players is willing to trade off

travel time for more scenic routes. Similarly, suppose one of the other players prefers to travel on high congestion paths, e.g., a rolling billboard seeking to maximize exposure. The aforementioned learning rules for Alice remain unchanged. Of course, Alice's actions implicitly depend on the utility functions of other players, but only indirectly through their selected paths. This characteristic of no explicit dependence on the utility functions of others is known as “uncoupled” learning, and it can have major implications on the achievable long-run behavior (Hart and Mas-Colell 2003a).

In assuming the ability to observe the paths of other players and to compute off-line travel times as a function of these paths, these learning rules impose severe requirements on the information available to each player. Less restrictive are learning rules that are “payoff based” (Young 2005). A simple modification that leads to payoff-based learning is as follows. Alice maintains an empirical average of the travel times of a path using only the days that she took that path. Note the distinction – on any given day, Alice remains unaware of travel times for the routes not selected. Using these empirical average travel times, Alice can then mimic any of the aforementioned learning rules. As intended, she does not directly observe the paths of others, nor does she have a closed-form expression for travel times as a function of player paths. Rather, she only can select a path and measure the consequences. As before, all players implementing such a learning rule induce a dynamic process, but the ensuing analysis in payoff-based learning can be more subtle.

Learning Dynamics

We now give a more formal presentation of selected learning rules and results concerning their long-run behavior.

Preliminaries

We begin with the basic setup of games with a finite set of players, $\{1, 2, \dots, N\}$, and for each player i , a finite set of strategies, $\mathcal{A}_i$. Let

$$\mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_N$$

denote the set of strategy profiles. Each player, i , is endowed with a utility function

$$u_i : \mathcal{A} \rightarrow \mathbb{R}.$$

Utility functions capture player preferences over strategy profiles. Accordingly, for any $a, a' \in \mathcal{A}$, the condition

$$u_i(a) > u_i(a')$$

indicates that player i prefers the strategy profile a over a' .

The notation $-i$ indicates the set of players *other than* player i . Accordingly, we sometimes write $a \in \mathcal{A}$ as (a_i, a_{-i}) to isolate a_i , the strategy of player i , versus a_{-i} , the strategies of other players. The notation $-i$ is used in other settings as well.

Utility functions induce best-response sets. For $a_{-i} \in \mathcal{A}_{-i}$, define

$$\mathcal{B}_i(a_{-i}) = \{a_i : u_i(a_i, a_{-i}) \geq u_i(a'_i, a_{-i}) \text{ for all } a'_i \in \mathcal{A}_i\}.$$

In words, $\mathcal{B}_i(a_{-i})$ denotes the set of strategies that are optimal for player i in response to the strategies of other players, a_{-i} .

A strategy profile $a^* \in \mathcal{A}$ is a *Nash equilibrium* if for any player i and any $a'_i \in \mathcal{A}_i$,

$$u_i(a_i^*, a_{-i}^*) \geq u_i(a'_i, a_{-i}^*).$$

In words, at a Nash equilibrium, no player can achieve greater utility by unilaterally changing strategies. Stated in terms of best-response sets, a strategy profile, a^* , is a Nash equilibrium if for every player i ,

$$a_i^* \in \mathcal{B}_i(a_{-i}^*).$$

We also will need the notions of *mixed strategies* and mixed strategy Nash equilibrium. Let $\Delta(\mathcal{A}_i)$ denote probability distributions (i.e., non-negative vectors that sum to one) over the set

$\mathcal{A}_i$. A mixed strategy profile is a collection of probability distributions, $\alpha = (\alpha_1, \dots, \alpha_N)$, with $\alpha_i \in \Delta(\mathcal{A}_i)$ for each i . Let us assume that players choose a strategy randomly and independently according to these mixed strategies. Accordingly, define $\mathbf{Pr}[a; \alpha]$ to be the probability of strategy a under the mixed strategy profile α , and define the expected utility of player i as

$$U_i(\alpha) = \sum_{a \in \mathcal{A}} u_i(a) \cdot \mathbf{Pr}[a; \alpha].$$

A *mixed strategy Nash equilibrium* is a mixed strategy profile, α^* , such that for any player i and any $\alpha'_i \in \Delta(\mathcal{A}_i)$,

$$U_i(\alpha_i^*, \alpha_{-i}^*) \geq U_i(\alpha'_i, \alpha_{-i}^*).$$

Special Classes of Games

We will reference three special classes of games: (i) zero-sum games, (ii) potential games, and (iii) weakly acyclic games.

Zero-sum games: There are only two players (i.e., $N = 2$), and $u_1(a) = -u_2(a)$.

Potential games: There exists a (potential) function,

$$\phi : \mathcal{A} \rightarrow \mathbb{R}$$

such that for any pair of strategies, $a = (a_i, a_{-i})$ and $a' = (a'_i, a_{-i})$, that differ only in the strategy of player i ,

$$u_i(a_i, a_{-i}) - u_i(a'_i, a_{-i}) = \phi(a_i, a_{-i}) - \phi(a'_i, a_{-i}).$$

Weakly acyclic games: There exists a function

$$\phi : \mathcal{A} \rightarrow \mathbb{R}$$

with the following property: if $a \in \mathcal{A}$ is *not* a Nash equilibrium, then at least one player, say player i , has an alternative strategy, say $a'_i \in \mathcal{A}_i$, such that

$$u_i(a'_i, a_{-i}) > u_i(a_i, a_{-i})$$

and

$$\phi(a'_i, a_{-i}) > \phi(a_i, a_{-i}).$$

Potential games are a special class of games for which various learning dynamics converge to a Nash equilibrium. The aforementioned commuting game constitutes a potential game under certain special assumptions. These are as follows: (i) the delay on a road only depends on the number of users (and not their identities) and (ii) all players measure delay in the same manner (Monderer and Shapley 1996).

Weakly acyclic games are a generalization of potential games. In potential games, there exists a potential function that captures differences in utility under unilateral (i.e., single player) changes in strategy. In weakly acyclic games (see Young 1998), if a strategy profile is not a Nash equilibrium, then there *exists* a player who can simultaneously achieve an increase in utility while increasing the potential function. The characterization of weakly acyclic games through a potential function herein is not traditional and is borrowed from Marden et al. (2009a).

Forecasted Best-Response Dynamics

One family of learning dynamics involves players formulating a forecast of the strategies of other players based on past observations and then playing a best response to this forecast.

Cournot Best-Response Dynamics

The simplest illustration is Cournot best-response dynamics. Players repetitively play the same game over stages $t = 0, 1, 2, \dots$. At stage t , a player forecasts that the strategies of other players are the strategies played at the previous stage $t - 1$. The following rules specify Cournot best response with inertia. For each stage t and for each player i :

- With probability $p \in (0, 1)$, $a_i(t) = a_i(t - 1)$ (inertia).
- With probability $1 - p$, $a_i(t) \in \mathcal{B}_i(a_{-i}(t - 1))$ (best response).
- If $a_i(t - 1) \in \mathcal{B}_i(a_{-i}(t - 1))$, then $a_i(t) = a_i(t - 1)$ (continuation).

Proposition 1 For weakly acyclic (and hence potential) games, player strategies under

Cournot best-response dynamics with inertia converge to a Nash equilibrium.

Cournot best-response dynamics need not always converge in games with a Nash equilibrium, hence the restriction to weakly acyclic games.

Fictitious Play

In fictitious play, introduced in Brown (1951), players also use past observations to construct a forecast of the strategies of other players. Unlike Cournot best-response dynamics, this forecast is *probabilistic*.

As a simple example, consider the commuting game with two players, Alice and Bob, who both must choose between two paths, *A* and *B*. Now suppose that on stage $t = 10$, Alice has observed Bob used path *A* for 6 out of the previous 10 days and path *B* for the remaining days. Then Alice's forecast of Bob is that he will chose path *A* with 60 % probability and path *B* with 40 % probability. Alice then chooses between path *A* and *B* in order to optimize her *expected* utility. Likewise, Bob uses Alice's empirical averages to form a probabilistic forecast of her next choice and selects a path to optimize his expected utility.

More generally, let $\pi_j(t) \in \Delta(\mathcal{A}_j)$ denote the *empirical frequency* for player j at stage t . This vector is a probability distribution that indicates the relative frequency of times player j played each strategy in $\mathcal{A}_j$ over stages $0, 1, \dots, t-1$. In fictitious play, player i assumes (incorrectly) that at stage t , other players will select their strategies independently and randomly according to their empirical frequency vectors. Let $\Pi_{-i}(t)$ denote the induced probability distribution over $\mathcal{A}_{-i}$ at stage t . Under fictitious play, player i selects an action according to

$$a_i(t) \in \arg \max_{a_i \in \mathcal{A}_i} \sum_{a_{-i} \in \mathcal{A}_{-i}} u_i(a_i, a_{-i}) \cdot \Pr[a_{-i}; \Pi_{-i}(t)].$$

In words, player i selects the action that maximizes expected utility assuming that other players select their strategies randomly and

independently according to their empirical frequencies.

Proposition 2 *For (i) zero-sum games, (ii) potential games, and (iii) two-player games in which one player has only two actions, player empirical frequencies under fictitious play converge to a mixed strategy Nash equilibrium.*

These results are reported in Fudenberg and Levine (1998), Hofbauer and Sandholm (2002), and Berger (2005). Fictitious play need not converge to Nash equilibria in all games. An early counterexample is reported in Shapley (1964), which constructs a two-player game with a unique mixed strategy Nash equilibrium. A weakly acyclic game with multiple pure (i.e., non-mixed) Nash equilibria under which fictitious play does not converge is reported in Foster and Young (1998).

A variant of fictitious play is “joint strategy” fictitious play (Marden et al. 2009b). In this framework, players construct as forecasts empirical frequencies of the *joint* play of other players. This formulation is in contrast to constructing and combining empirical frequencies for each player. In the commuting game, it turns out that joint strategy fictitious play is equivalent to the aforementioned “exploitation” rule of selecting the path with lowest average travel time. Marden et al. (2009b) show that action profiles under joint strategy fictitious play (with inertia) converge to a Nash equilibrium in potential games.

Log-Linear Learning

Under forecasted best-response dynamics, players chose a best response to the forecasted strategies of other players. Log-linear learning, introduced in Blume (1993), allows the possibility of “exploration,” in which players can select nonoptimal strategies but with relatively low probabilities.

Log-linear learning proceeds as follows. First, introduce a “temperature” parameter, $T > 0$.

- At stage t , a single player, say player i , is selected at random.
- For player i ,

$$\Pr[a_i(t) = a'_i] = \frac{1}{Z} e^{u_i(a'_i, a_{-i}(t-1))/T}.$$

- For all other players, $j \neq i$,

$$a_j(t) = a_j(t-1).$$

In words, under log-linear learning, only a single player performs a strategy update at each stage. The probability of selecting a strategy is exponentially proportional to the utility garnered from that strategy (with other players repeating their previous strategies). In the above description, the dummy parameter Z is a normalizing variable used to define a probability distribution. In fact, the specific probability distribution for strategy selection is a Gibbs distribution with temperature parameter, T . For very large T , strategies are chosen approximately uniformly at random. However, for small T , the selected strategy is a best response (i.e., $a_i(t) \in \mathcal{B}_i(a_{-i}(t-1))$) with high probability, and an alternative strategy is selected with low probability.

Because of the inherent randomness, strategy profiles under log-linear learning never converge. Nonetheless, the long-run behavior can be characterized probabilistically as follows.

Proposition 3 *For potential games with potential function $\phi(\cdot)$ under log-linear learning, for any $a \in \mathcal{A}$,*

$$\lim_{t \rightarrow \infty} \Pr[a(t) = a] = \frac{1}{Z} e^{\phi(a)/T}.$$

In words, the long-run probabilities of strategy profiles conform to a Gibbs distribution constructed from the underlying potential function. This characterization has the important implication of (probabilistic) equilibrium *selection*. Prior convergence results stated convergence to Nash equilibria, but did not specify which Nash equilibrium in the case of multiple equilibria. Under log-linear learning, there is a probabilistic preference for the Nash equilibrium that maximizes the underlying potential function.

Extensions and Variations

Payoff-based learning. The discussion herein presumed that players can observe the actions of other players and can compute utility functions off-line. Payoff-based algorithms, i.e., algorithms in which players only measure the utility garnered in each stage, impose less restrictive informational requirements. See Young (2005) for a general discussion, as well as Marden et al. (2009c), Marden and Shamma (2012), and Arslan and Shamma (2004) for various payoff-based extensions.

No-regret learning. The broad class of so-called “no-regret” learning rules has the desirable property of converging to broader solution concepts (namely, Hannan consistency sets and correlated equilibria) in general games. See Hart and Mas-Colell (2000, 2001, 2003b) for an extensive discussion.

Calibrated forecasts. Calibrated forecasts are more sophisticated than empirical frequencies in that they satisfy certain long-run consistency properties. Accordingly, forecasted best-response learning using calibrated forecasts has stronger guaranteed convergence properties, such as convergence to correlated equilibria. See Foster and Vohra (1997), Kakade and Foster (2008), and Mannor et al. (2007).

Impossibility results. This entry focused on convergence results in various special cases. There are broad impossibility results that imply the impossibility of families of learning rules to converge to Nash equilibria in all games. The focus is on *uncoupled* learning, i.e., the learning dynamics for player i does not depend explicitly on the utility functions of other players (which is satisfied by all of the learning dynamics presented herein). See Hart and Mas-Colell (2003a, 2006), Hart and Mansour (2007), and Shamma and Arslan (2005). Another type of impossibility result concerns lower bounds on the required rate of convergence to equilibrium (e.g., Hart and Mansour 2010).

Welfare maximization. Of special interest is learning dynamics that select welfare (i.e., sum of utilities) maximizing strategy profiles, whether or not they are Nash equilibria. Recent contributions include Pradelski and Young (2012), Marden et al. (2011), and Arieli and Babichenko (2012).

Summary and Future Directions

We have presented a selection of learning dynamics and their long-run characteristics, specifically in terms of convergence to Nash equilibria. As stated early on, the original motivation of learning in games research has been to add credence to solution concepts such as Nash equilibrium as a model of the outcome of a game. An emerging line of research stems from engineering considerations, in which the objective is to use the framework of learning in games as a design tool for distributed decision architecture settings such as autonomous vehicle teams, communication networks, or smart grid energy systems. A related emerging direction is social influence, in which the objective is to steer the collective behaviors of human decision makers towards a socially desirable situation through the dispersment of incentives. Accordingly, learning in games can offer baseline models on how individuals update their behaviors to guide and inform social influence policies.

Cross-References

- [Evolutionary Games](#)
- [Stochastic Games and Learning](#)

Recommended Reading

Monographs on learning in games:

- Fudenberg D, Levine DK (1998) The theory of learning in games. MIT, Cambridge
- Hart S, Mas Colell A (2013) Simple adaptive strategies: from regret-matching to uncoupled dynamics. World Scientific Publishing Company

- Young HP (1998) Individual strategy and social structure. Princeton University Press, Princeton
- Young HP (2005) Strategic learning and its limits. Princeton University Press, Princeton

Overview articles on learning in games:

- Fudenberg D, Levine DK (2009) Learning and equilibrium. *Annu Rev Econ* 1:385–420
- Hart S (2005) Adaptive heuristics. *Econometrica* 73(5):1401–1430
- Young HP (2007) The possible and the impossible in multi-agent learning. *Artif Intell* 171:429–433

Articles relating learning in games to distributed control:

- Li N, Marden JR (2013) Designing games for distributed optimization. *IEEE J Sel Top Signal Process* 7:230–242
- Mannor S, Shamma JS (2007) Multi-agent learning for engineers. *Artif Intell* 171:417–422
- Marden JR, Arslan G, Shamma JS (2009) Cooperative control and potential games. *IEEE Trans Syst Man Cybern B Cybern* 6:1393–1407

Bibliography

- Arieli I, Babichenko Y (2012) Average-testing and Pareto efficiency. *J Econ Theory* 147: 2376–2398
- Arrow KJ (1986) Rationality of self and others in an economic system. *J Bus* 59(4):S385–S399
- Arslan G, Shamma JS (2004) Distributed convergence to Nash equilibria with local utility measurements. In: *Proceedings of the 43rd IEEE conference on decision and control*. Atlantis, Paradise Island, Bahamas
- Berger U (2005) Fictitious play in $2 \times n$ games. *J Econ Theory* 120(2):139–154
- Blume L (1993) The statistical mechanics of strategic interaction. *Games Econ Behav* 5:387–424
- Brown GW (1951) Iterative solutions of games by fictitious play. In: Koopmans TC (ed) *Activity analysis of production and allocation*. Wiley, New York, pp 374–376
- Foster DP, Vohra R (1997) Calibrated learning and correlated equilibrium. *Games Econ Behav* 21:40–55

- Foster DP, Young HP (1998) On the nonconvergence of fictitious play in coordination games. *Games Econ Behav* 25:79–96
- Fudenberg D, Levine DK (1998) *The theory of learning in games*. MIT, Cambridge
- Hart S (2005) Adaptive heuristics. *Econometrica* 73(5):1401–1430
- Hart S, Mansour Y (2007) The communication complexity of uncoupled Nash equilibrium procedures. In: STOC'07 proceedings of the 39th annual ACM symposium on the theory of computing, San Diego, CA, USA, pp 345–353
- Hart S, Mansour Y (2010) How long to equilibrium? The communication complexity of uncoupled equilibrium procedures. *Games Econ Behav* 69(1):107–126
- Hart S, Mas-Colell A (2000) A simple adaptive procedure leading to correlated equilibrium. *Econometrica* 68(5):1127–1150
- Hart S, Mas-Colell A (2001) A general class of adaptive strategies. *J Econ Theory* 98:26–54
- Hart S, Mas-Colell A (2003a) Uncoupled dynamics do not lead to Nash equilibrium. *Am Econ Rev* 93(5):1830–1836
- Hart S, Mas-Colell A (2003b) Regret based continuous-time dynamics. *Games Econ Behav* 45:375–394
- Hart S, Mas-Colell A (2006) Stochastic uncoupled dynamics and Nash equilibrium. *Games Econ Behav* 57(2):286–303
- Hofbauer J, Sandholm W (2002) On the global convergence of stochastic fictitious play. *Econometrica* 70:2265–2294
- Kakade SM, Foster DP (2008) Deterministic calibration and Nash equilibrium. *J Comput Syst Sci* 74: 115–130
- Mannor S, Shamma JS, Arslan G (2007) Online calibrated forecasts: memory efficiency vs universality for learning in games. *Mach Learn* 67(1):77–115
- Marden JR, Shamma JS (2012) Revisiting log-linear learning: asynchrony, completeness and a payoff-based implementation. *Games Econ Behav* 75(2):788–808
- Marden JR, Arslan G, Shamma JS (2009a) Cooperative control and potential games. *IEEE Trans Syst Man Cybern Part B Cybern* 39:1393–1407
- Marden JR, Arslan G, Shamma JS (2009b) Joint strategy fictitious play with inertia for potential games. *IEEE Trans Autom Control* 54:208–220
- Marden JR, Young HP, Arslan G, Shamma JS (2009c) Payoff based dynamics for multi-player weakly acyclic games. *SIAM J Control Optim* 48:373–396
- Marden JR, Peyton Young H, Pao LY (2011) Achieving Pareto optimality through distributed learning. *Oxford economics discussion paper #557*
- Monderer D, Shapley LS (1996) Potential games. *Games Econ Behav* 14:124–143
- Pradelski BR, Young HP (2012) Learning efficient Nash equilibria in distributed systems. *Games Econ Behav* 75:882–897
- Roughgarden T (2005) *Selfish routing and the price of anarchy*. MIT, Cambridge
- Shamma JS, Arslan G (2005) Dynamic fictitious play, dynamic gradient play, and distributed convergence

to Nash equilibria. *IEEE Trans Autom Control* 50(3):312–327

Shapley LS (1964) Some topics in two-person games. In: Dresher M, Shapley LS, Tucker AW (eds) *Advances in game theory*. University Press, Princeton, pp 1–29

Skyrms B (1992) Chaos and the explanatory significance of equilibrium: strange attractors in evolutionary game dynamics. In: PSA: proceedings of the biennial meeting of the Philosophy of Science Association. Volume Two: symposia and invited papers, East Lansing, MI, USA, pp 274–394

Young HP (1998) *Individual strategy and social structure*. Princeton University Press, Princeton

Young HP (2005) *Strategic learning and its limits*. Oxford University Press, Oxford

Learning Theory: The Probably Approximately Correct Framework

Mathukumalli Vidyasagar

University of Texas at Dallas, Richardson, TX, USA

Abstract

In this article, a brief overview is given of one particular approach to machine learning, known as PAC (probably approximately correct) learning theory. A central concept in PAC learning theory is the Vapnik-Chervonenkis (VC) dimension. Finiteness of the VC-dimension is sufficient for PAC learnability, and in some cases, is also necessary. Some directions for future research are also indicated.

Keywords

Machine learning · Probably approximately correct (PAC) learning · Support vector machine · Vapnik-Chervonenkis (V-C) dimension

Introduction

How does a machine learn an abstract concept from examples? How can a machine generalize to previously unseen situations? Learning theory

is the study of (formalized versions of) such questions. There are many possible ways to formulate such questions. Therefore, the focus of this entry is on one particular formalism, known as PAC (probably approximately correct) learning. It turns out that PAC learning theory is rich enough to capture intuitive notions of what learning should mean in the context of applications and, at the same time, is amenable to formal mathematical analysis. There are several precise and complete studies of PAC learning theory, many of which are cited in the bibliography. Therefore, this article is devoted to sketching some high-level ideas.

Problem Formulation

In the PAC formalism, the starting point is the premise that there is an unknown set, say an unknown convex polygon, or an unknown half-plane. The unknown set cannot be *completely* unknown; rather, something should be specified about its nature, in order for the problem to be both meaningful and tractable. For instance, in the first example above, the learner knows that the unknown set is a convex polygon, though it is not known *which* polygon it might be. Similarly, in the second example, the learner knows that the unknown set is a half-plane, though it is not known *which* half-plane. The collection of all possible unknown sets is known as the **concept class**, and the particular unknown set is referred to as the “target concept.” In the first example, this would be the set of all convex polygons and in the second case it would be the set of half-planes. The unknown set cannot be directly observed of course; otherwise, there would be nothing to learn. Rather, one is given clues about the target concept by an “oracle,” which informs the learner whether or not a particular element belongs to the target concept. Therefore, the information available to the learner is a collection of “labelled samples,” in the form $\{(x_i, I_T(x_i)), i = 1, \dots, m\}$, where m is the total number of labelled samples and $I_T(\cdot)$ is the indicator function of the target concept T . Based on this information, the learner is expected

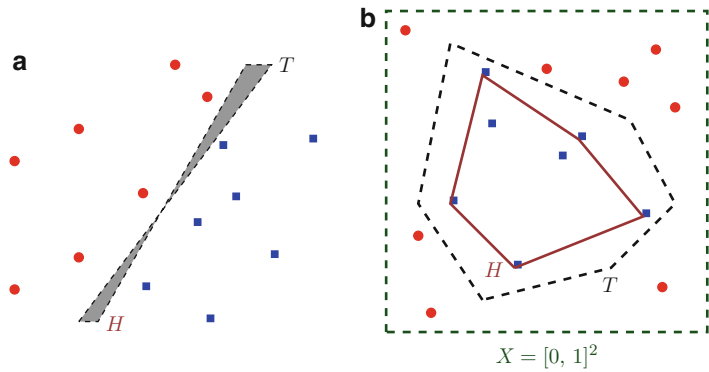
to generate a “hypothesis” H_m that is a good approximation to the unknown target concept T .

One of the main features of PAC learning theory that distinguishes it from its forerunners is the observation that, no matter how many training samples are available to the learner, the hypothesis H_m can never *exactly equal* the unknown target concept T . Rather, all that one can expect is that H_m converges to T in some appropriate metric. Since the purpose of machine learning is to generate a hypothesis H_m that can be used to approximate the unknown target concept T for prediction purposes, a natural candidate for the metric that measures the disparity between H_m and T is the so-called generalization error, defined as follows: Suppose that, after m training samples that have led to the hypothesis H_m , a testing sample x is generated at random. One can now ask: what is the probability that the hypothesis H_m misclassifies x ? In other words, what is the value of $\Pr\{I_{H_m}(x) \neq I_T(x)\}$? This quantity is known as the generalization error, and the objective is to ensure that it approaches zero as $m \rightarrow \infty$.

The manner in which the samples are generated leads to different models of learning. For instance, if the learner is able to choose the next sample x_{m+1} on the basis of the previous m labelled samples, which is then passed on to the oracle for labeling, this is known as “active learning.” More common is “passive learning,” in which the sequence of training samples $\{x_i\}_{i \geq 1}$ is generated at random, in an independent and identically distributed (i.i.d.) fashion, according to some probability distribution P . In this case, even the hypothesis H_m and the generalization error are random, because they depend on the randomly generated training samples. This is the rationale behind the nomenclature “probably approximately correct.” The hypothesis H_m is not expected to equal to unknown target concept T exactly, only approximately. Even that is only probably true, because in principle it is possible that the randomly generated training samples could be totally unrepresentative and thus lead to a poor hypothesis. If we toss a coin many times, there is a small but always positive probability that it could turn up heads every time. As the coin

Learning Theory: The Probably Approximately Correct Framework,

Fig. 1 Examples of learning problems. (a) Learning half-planes. (b) Learning convex polygons



is tossed more and more times, this probability becomes smaller, but will never equal zero.

Examples

Example 1 Consider the situation where the concept class consists of all half-planes in $\mathbb{R}^2$, as indicated in the left side of Fig. 1. Here the unknown target concept T is some fixed but unknown half-plane. The symbol T is next to the boundary of the half-plane, and all points to the right of the line constitute the target half-plane. The training samples, generated at random according some unknown probability distribution P , are also shown in the figure. The samples that belong to T are shown as blue rectangles, while those that do not belong to T are shown as red dots. Knowing only these labelled samples, the learner is expected to guess what T might be.

A reasonable approach is to choose some half-plane that agrees with the data and correctly classifies the labelled data. For instance, the well-known support vector machine (SVM) algorithm chooses the unique half-plane such that the closest sample to the dividing line is as far as possible from it; see the paper by Cortes and Vapnik (1997).

The symbol H denotes the boundary of a hypothesis, which is another half-plane. The shaded region is the **symmetric difference** between the two half-planes. The set $T \Delta H$ is the set of points that are misclassified by the hypothesis H . Of course, we do not know what this set is, because we do not know T .

It can be shown that, whenever the hypothesis H is chosen to be **consistent** in the sense of correctly classifying all labelled samples, the generalization error goes to zero as the number of samples approaches infinity.

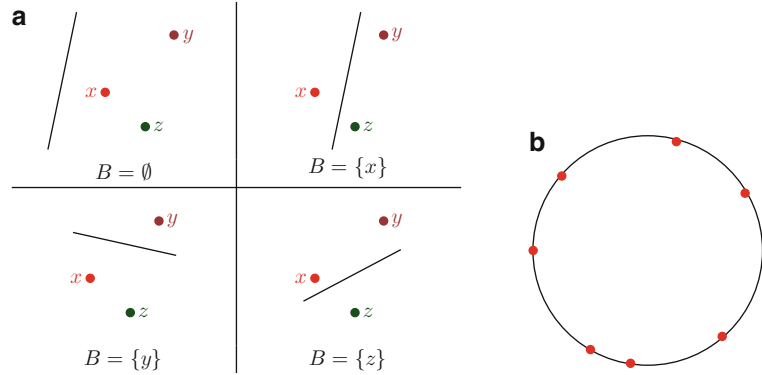
Example 2 Now suppose the concept class consists of all convex polygons in the unit square, and let T denote the (unknown) target convex polygon. This situation is depicted in the right side of Fig. 1. This time let us assume that the probability distribution that generates the samples is the uniform distribution on X . Given a set of positively and negatively labelled samples (the same convention as in Example 1), let us choose the hypothesis H to be the convex hull of all positively labelled samples, as shown in the figure. Since every positively labelled sample belongs to T , and T is a convex set, it follows that H is a subset of T . Moreover, $P(T \setminus H)$ is the generalization error. It can be shown that this algorithm also “works” in the sense that the generalization error goes to zero as the number of samples approaches infinity.

Vapnik-Chervonenkis Dimension

Given any concept class C , there is a single integer that offers a measure of the richness of the class, known as the Vapnik-Chervonenkis (or VC) dimension, after its originators.

Definition 1 A set $S \subseteq X$ is said to be **shattered** by a concept class C if, for every subset $B \subseteq S$, there is a set $A \in C$ such that $S \cap A = B$. The

Learning Theory: The Probably Approximately Correct Framework, Fig. 2 VC dimension illustrations. (a) Shattering a set of three elements. (b) Infinite VC dimension



VC dimension of C is the largest integer d such that there is a finite set of cardinality d that is shattered by C .

Example 3 It can be shown that the set of half-planes in $\mathbb{R}^2$ has VC dimension two. Choose a set $S = \{x, y, z\}$ consisting of three points that are not collinear, as in Fig. 2. Then there are $2^3 = 8$ subsets of S . The point is to show that for each of these eight subsets, there is a half-plane that contains precisely that subset, nothing more and nothing less. That this is possible is shown in Fig. 2. Four out of the eight situations are depicted in this figure, and the remaining four situations can be covered by taking the complement of the half-plane shown. It is also necessary to show that *no set with four or more elements can be shattered*, but that step is omitted; instead the reader is referred to any standard text such as Vidyasagar (1997). More generally, it can be shown that the set of half-planes in $\mathbb{R}^k$ has VC dimension $k + 1$.

Example 4 The set of convex polygons has infinite VC dimension. To see this, let S be a strictly convex set, as shown in Fig. 2b. (Recall that a set is “strictly convex” if none of its boundary points is a convex combination of other points in the set.) Choose any finite collection of boundary points, call it $S = \{x_1, \dots, x_n\}$. If B is a subset of S , then the convex hull of B does not contain any other point of S , due to the strict convexity property. Since this argument holds for every integer n , the class of convex polygons has infinite VC dimension.

Two Important Theorems

Out of the many important results in learning theory, two are noteworthy.

Theorem 1 (Blumer et al. (1989)) *A concept class is distribution-free PAC learnable if and only if it has finite VC dimension.*

Theorem 2 (Benedek and Itai (1991)) *Suppose P is a fixed probability distribution. Then the concept class C is PAC learnable if and only if, for every positive number ϵ , it is possible to cover C by a finite number of balls of radius ϵ , with respect to the pseudometric d_P .*

Now let us return to the two examples studied previously. Since the set of half-planes has finite VC dimension, it is distribution-free PAC learnable. The set of convex polygons can be shown to satisfy the conditions of Theorem 2 if P is the uniform distribution and is therefore PAC learnable. However, since it has infinite VC dimension, it follows from Theorem 1 that it is *not* distribution-free PAC learnable.

Summary and Future Directions

This brief entry presents only the most basic aspects of PAC learning theory. Many more results are known about PAC learning theory, and of course many interesting problems remain unsolved. Some of the known extensions are:

- Learning under an “intermediate” family of probability distributions $\mathcal{P}$ that is not neces-

sarily equal to $\mathcal{P}^*$, the set of *all* distributions (Kulkarni and Vidyasagar 1997)

- Relaxing the requirement that the algorithm should work uniformly well for all target concepts and requiring instead only that it should work with high probability (Campi and Vidyasagar 2001)
- Relaxing the requirement that the training samples are independent of each other and permitting them to have Markovian dependence (Gamarnik 2003; Meir 2000) or β -mixing dependence (Vidyasagar 2003)

There is considerable research in finding alternate sets of necessary and sufficient conditions for learnability. Unfortunately, many of these conditions are unverifiable and amount to tautological restatements of the problem under study.

Cross-References

- [Iterative Learning Control](#)
- [Learning in Games](#)
- [Optimal Control and the Dynamic Programming Principle](#)
- [Stochastic Games and Learning](#)

Bibliography

- Anthony M, Bartlett PL (1999) Neural network learning: theoretical foundations. Cambridge University Press, Cambridge
- Anthony M, Biggs N (1992) Computational learning theory. Cambridge University Press, Cambridge
- Benedek G, Itai A (1991) Learnability by fixed distributions. Theor Comput Sci 86:377–389
- Blumer A, Ehrenfeucht A, Haussler D, Warmuth M (1989) Learnability and the Vapnik-Chervonenkis dimension. J ACM 36(4):929–965
- Campi M, Vidyasagar M (2001) Learning with prior information. IEEE Trans Autom Control 46(11):1682–1695
- Devroye L, Györfi L, Lugosi G (1996) A probabilistic theory of pattern recognition. Springer, New York
- Gamarnik D (2003) Extension of the PAC framework to finite and countable Markov chains. IEEE Trans Inf Theory 49(1):338–345
- Kearns M, Vazirani U (1994) Introduction to computational learning theory. MIT, Cambridge
- Kulkarni SR, Vidyasagar M (1997) Learning decision rules under a family of probability measures. IEEE Trans Inf Theory 43(1):154–166
- Meir R (2000) Nonparametric time series prediction through adaptive model selection. Mach Learn 39(1):5–34
- Natarajan BK (1991) Machine learning: a theoretical approach. Morgan-Kaufmann, San Mateo
- van der Vaart AW, Wallner JA (1996) Weak convergence and empirical processes. Springer, New York
- Vapnik VN (1995) The nature of statistical learning theory. Springer, New York
- Vapnik VN (1998) Statistical learning theory. Wiley, New York
- Vidyasagar M (1997) A theory of learning and generalization. Springer, London
- Vidyasagar M (2003) Learning and generalization: with applications to neural networks. Springer, London

Lie Algebraic Methods in Nonlinear Control

Matthias Kowski

School of Mathematical and Statistical Sciences,
Arizona State University, Tempe, AZ, USA

Abstract

Lie algebraic methods generalize matrix methods and algebraic rank conditions to smooth nonlinear systems. They capture the essence of noncommuting flows and give rise to noncommutative analogues of Taylor expansions. Lie algebraic rank conditions determine controllability, observability, and optimality. Lie algebraic methods are also employed for state-space realization, control design, and path planning.

Keywords

Baker-Campbell-Hausdorff formula ·
Chen-Fliess series · Lie bracket

Definition

This article considers generally nonlinear control systems (affine in the control) of the form

This work was partially supported by the National Science Foundation through the grant DMS 09-08204.

$$\begin{aligned}\dot{x} &= f_0(x) + u_1 f_1(x) + \dots + u_m f_m(x) \\ y &= \varphi(x)\end{aligned}\quad (1)$$

where the state x takes values in $\mathbb{R}^n$, or more generally in an n -dimensional manifold M^n , the f_i are smooth vector fields, $\varphi: \mathbb{R}^n \mapsto \mathbb{R}^p$ is a smooth output function, and the controls $u = (u_1, \dots, u_m): [0, T] \mapsto U$ are piecewise continuous, or, more generally, measurable functions taking values in a closed convex subset $U \subseteq \mathbb{R}^m$ that contains 0 in its interior.

Lie algebraic techniques refers to analyzing the system (1) and designing controls and stabilizing feedback laws by employing relations satisfied by iterated Lie brackets of the system vector fields f_i .

Introduction

Systems of the form (1) contain as a special case time-invariant linear systems $\dot{x} = Ax + Bu$, $y = Cx$ (with constant matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and $C \in \mathbb{R}^{p \times n}$) that are well-studied and are a mainstay of classical control engineering. Properties such as controllability, stabilizability, observability, and optimal control and various others are determined by relationships satisfied by higher-order matrix products of A , B , and C .

Since the early 1970s, it has been well understood that the appropriate generalization of this matrix algebra, and, e.g., invariant linear subspaces, to nonlinear systems is in terms of the Lie algebra generated by the vector fields f_i , integral submanifolds of this Lie algebra, and the algebra of iterated Lie derivatives of the output function.

The Lie bracket of two smooth vector fields $f, g: M \mapsto TM$ is defined as the vector field $[f, g]: M \mapsto TM$ that maps any smooth function $\varphi \in C^\infty(M)$ to the function $[f, g]\varphi = fg\varphi - gf\varphi$.

In local coordinates, if

$$f(x) = \sum_{i=1}^n f^i(x) \frac{\partial}{\partial x^i} \quad \text{and}$$

$$g(x) = \sum_{i=1}^n g^i(x) \frac{\partial}{\partial x^i},$$

then

$$\begin{aligned}[f, g](x) &= \sum_{i,j=1}^n \left(f^j(x) \frac{\partial g^i}{\partial x^j}(x) \right. \\ &\quad \left. - g^j(x) \frac{\partial f^i}{\partial x^j}(x) \right) \frac{\partial}{\partial x^i}.\end{aligned}$$

With some abuse of notation, one may abbreviate this to $[f, g] = (Dg)f - (Df)g$, where f and g are considered as column vector fields and Df and Dg denote their Jacobian matrices of partial derivatives.

Note that with this convention the Lie bracket corresponds to the negative of the commutator of matrices: If $P, Q \in \mathbb{R}^{n \times n}$ define, in matrix notation, the linear vector fields $f(x) = Px$ and $g(x) = Qx$, then $[f, g](x) = (QP - PQ)x = -[P, Q]x$.

Noncommuting Flows

Geometrically the Lie bracket of two smooth vector fields f_1 and f_2 is an infinitesimal measure of the lack of commutativity of their flows. For a smooth vector field f and an initial point $x(0) = p \in M$, denote by $e^{tf}p$ the solution of the differential equation $\dot{x} = f(x)$ at time t . Then

$$\begin{aligned}[f_1, f_2]\varphi(p) &= \lim_{t \rightarrow 0} \frac{1}{2t^2} \left(\varphi \left(e^{-tf_2} e^{-tf_1} e^{tf_2} e^{tf_1} p \right) \right. \\ &\quad \left. - \varphi(p) \right).\end{aligned}$$

As a most simple example, consider parallel parking a unicycle, moving it sideways without slipping. Introduce coordinates (x, y, θ) for the location in the plane and the steering angle. The dynamics are governed by $\dot{x} = u_1 \cos \theta$, $\dot{y} = u_1 \sin \theta$, and $\dot{\theta} = u_2$ where the control u_1 is interpreted as the signed rolling speed and u_2 as the angular velocity of the steering angle. Written in the form (1), one has $f_1 = (\cos \theta, \sin \theta, 0)^T$ and $f_2 = (0, 0, 1)^T$. (In this

case the drift vector field $f_0 \equiv 0$ vanishes.) If the system starts at $(0, 0, 0)^T$, then via the sequence of control actions of the form *turn left*, *roll forward*, *turn back*, and *roll backwards*, one may steer the system to a point $(0, \Delta y, 0)^T$ with $\Delta y > 0$. This sideways motion corresponds to the value $(0, 1, 0)^T$ of the Lie bracket $[f_1, f_2] = (-\sin \theta, \cos \theta, 0)^T$ at the origin. It encapsulates that steering and rolling do not commute. This example is easily expanded to model, e.g., the sideways motion of a car, or a truck with multiple trailers; see, e.g., Bloch (2003), Bressan and Piccoli (2007), and Bullo and Lewis (2005). In such cases longer iterated Lie brackets correspond to the required more intricate control actions needed to obtain, e.g., a pure sideways motion.

In the case of linear systems, if the Kalman rank condition $\text{rank}[B, AB, A^2B, \dots, A^{n-1}B] = n$ is not satisfied, then all solutions curves of the system starting from the same point $x(0) = p$ are at all times $T > 0$ constrained to lie in a proper affine subspace. In the nonlinear setting the role of the compound matrix of that condition is taken by the Lie algebra $L = L(f_0, f_1, \dots, f_m)$ of all finite linear combinations of iterated Lie brackets of the vector fields f_i . As an immediate consequence of the Frobenius integrability theorem, if at a point $x(0) = p$ the vector fields in L span the whole tangent space, then it is possible to reach an open neighborhood of the initial point by concatenating flows of the system (1) that correspond to piecewise constant controls. Conversely, in the case of analytic vector fields and a compact set U of admissible control values, the Hermann-Nagano theorem guarantees that if the dimension of the subspace $L(p) = \{f(p) : f \in L\} < n$ is not maximal, then all such trajectories are confined to stay in a lower-dimensional proper integral submanifold of L through the point p . For a comprehensive introduction, see, e.g., the textbooks Bressan and Piccoli (2007), Isidori (1995), and Sontag (1998).

Controllability

Define the reachable set $\mathcal{R}_T(p)$ as the set of all terminal points $x(T; u, p)$ at time T

of trajectories of (1) that start at the initial point $x(0) = p$ and correspond to admissible controls. Commonly known as the *Lie algebra rank condition* (LARC), the above condition determines whether the system is accessible from the point p , which means that for arbitrarily small time $T > 0$, the reachable set $\mathcal{R}_T(p)$ has nonempty n -dimensional interior. For most applications one desires stronger controllability properties. Most amenable to Lie algebraic methods, and practically relevant, is small-time local controllability (STLC): The system is STLC from p if p lies in the interior of $\mathcal{R}_T(p)$ for every $T > 0$. In the case that there is no drift vector field f_0 , accessibility is equivalent to STLC. However, in general, the situation is much more intricate, and a rich literature is devoted to various necessary or sufficient conditions for STLC. A popular such condition is the Hermes condition. For this define the subspaces $S^1 = \text{span}\{(\text{ad}^j f_0, f_i) : 1 \leq j \leq m, i \in \mathbb{Z}^+\}$, and recursively $S^{k+1} = \text{span}\{[g_1, g_k] : g_1 \in S^1, g_k \in S^k\}$. Here $(\text{ad}^0 f, g) = g$, and recursively $(\text{ad}^{k+1} f, g) = [f, (\text{ad}^k f, g)]$. The Hermes condition guarantees in the case of analytic vector fields and, e.g., $U = [-1, 1]^m$ that if the system satisfies the (LARC) and for every $k \geq 1$, $S^{2k}(p) \subseteq S^{2k-1}(p)$, then the system is (STLC). For more general conditions, see Sussmann (1987) and also Kawski (1990) for a broader discussion.

The importance and value of Lie algebraic conditions may in large part be ascribed to their geometric character, their being invariant under coordinate changes and feedback. In particular, in the analytic case, the Lie relations completely determine the local properties of the system, in the sense that Lie algebra homomorphism between two systems gives rise to a local diffeomorphism that maps trajectories to trajectories (Sussmann 1974).

Exponential Lie Series

A central analytic tool in Lie algebraic methods that takes the role of Taylor expansions in classical analysis of dynamical system is the Chen-Fliess series which associates to every admissible

control $u: [0, T] \mapsto U$ a formal power series

$$\text{CF}(u, T) = \sum_I \int_0^T du^I \cdot X_{i_1} \dots X_{i_s} \quad (2)$$

over a set $\{X_0, X_1, \dots, X_m\}$ of noncommuting indeterminates (or *letters*). For every multi-index $I = (i_1, i_2, \dots, i_s) \in \{0, 1, \dots, m\}^s$, $s \geq 0$, the coefficient of X_I is the iterated integral defined recursively

$$\int_0^T du^{(I, j)} = \int_0^T \left(\int_0^t u^I \right) du_j(t). \quad (3)$$

Upon evaluating this series via the substitutions $X_i \longleftarrow f_i$, it becomes an *asymptotic series for the propagation of solutions of (1)*: For f_j, φ analytic, U compact, p in a compact set, and $T \geq 0$ sufficiently small, one has

$$\varphi(x(t; u, p)) = \sum_I \int_0^T du^I \cdot (f_{i_1} \dots f_{i_s} \varphi)(p). \quad (4)$$

One application of particular interest is to construct approximating systems of a given system (1) that preserve critical geometric properties, but which have a simpler structure. One such class is that of nilpotent systems, that is, systems whose Lie algebra $L = L(f_0, f_1, \dots, f_m)$ is nilpotent, and for which solutions can be found by simple quadratures. While truncations of the Chen-Fliess series never correspond to control systems of the same form, much work has been done in recent years to rewrite this series in more useful formats. For example, the infinite directed exponential product expansion in Sussmann (1986) that uses Hall trees immediately may be interpreted in terms of free nilpotent systems and consequently helps in the construction of nilpotent approximating systems. More recent work, much of it of a combinatorial algebra nature and utilizing the underlying Hopf algebras, further simplifies similar expansions and in particular yields explicit formulas for a continuous Baker-Campbell-Hausdorff formula or for the logarithm of the Chen-Fliess series (Gehrig and Kawski 2008).

Observability and Realization

In the setting of linear systems a well-defined algebraic sense dual to the concept of controllability is that of observability. Roughly speaking the system (1) is observable if knowledge of the output $y(t) = \varphi(x(t; u, p))$ over an arbitrarily small interval suffices to construct the current state $x(t; u, p)$ and indeed the past trajectory $x(\cdot; u, p)$. In the linear setting observability is equivalent to the rank condition $\text{rank}[C^T, (CA)^T, \dots, (CA^{n-1})^T] = n$ being satisfied. In the nonlinear setting, the place of the rows of this compound matrix is taken by the functions in the observation algebra, which consists of all finite linear combinations of iterated Lie derivatives $f_{i_s} \dots f_{i_1} \varphi$ of the output function.

Similar to the Hankel matrices introduced in the latter setting, in the case of a finite Lie rank, one again can use the output algebra to construct realizations in the form of (1) for systems which are initially only given in terms of input-output descriptions, or in terms of formal Fliess operators; see, e.g., Fliess (1980), Gray and Wang (2002), and Jakubczyk (1986) for further reading.

Optimal Control

In a well-defined geometric way, conditions for optimal control are dual to conditions for controllability and thus are directly amenable to Lie algebraic methods. Instead of considering a separate functional

$$J(u) = \psi(x(T; u, p)) + \int_0^T L(t, x(t; u, p), u(t)) dt \quad (5)$$

to be minimized, it is convenient for our purposes to augment the state by, e.g., defining $\dot{x}_0 = 1$ and $\dot{x}_{n+1} = L(x_0, x, u)$. For example, in the case of time-optimal control, one again obtains an enlarged system of the same form (1); else one utilizes more general Lie algebraic methods that also apply to systems not necessarily affine in the control.

The basic picture for systems with a compact set U of admissible values of the controls involves the attainable funnel $\mathcal{R}_{\leq T}(p)$ consisting of all trajectories of the system (1) starting at $x(0) = p$ that correspond to admissible controls. The trajectory corresponding to an optimal control u^* must at time T lie on the boundary of the funnel $\mathcal{R}_{\leq T}(p)$ and hence also at all prior times (using the invariance of domain property implied by the continuity of the flow). Hence one may associate a covector field along such optimal trajectory that at every time points in the direction of an outward normal. The Pontryagin Maximum Principle is a first-order characterization of such trajectory covector field pairs. Its pointwise maximization condition essentially says that if at any time $t_0 \in [0, T]$ one replaces the optimal control $u^*(\cdot)$ by any admissible control variation on an interval $[t_0, t_0 + \varepsilon]$, then such variation may be transported along the flow to yield, in the limit as $\varepsilon \searrow 0$, an *inward* pointing tangent vector to the reachable set $\mathcal{R}_T(p)$ at $x(T; u^*, p)$. To obtain stronger higher-order conditions for maximality, one may combine several such families of control variations. The effects of such combinations are again calculated in terms of iterated Lie brackets of the vector fields f_i . Indeed, necessary conditions for optimality, for a trajectory to lie on the boundary of the funnel $\mathcal{R}_{\leq T}(p)$, immediately translate into sufficient conditions for STLC, for the initial point to lie in the interior of $\mathcal{R}_T(p)$, and vice versa. For recent work employing Lie algebraic methods in optimality conditions, see, e.g., Agrachev et al. (2002).

Summary and Future Research

Lie algebraic techniques may be seen as a direct generalization of matrix linear algebra tools that have proved so successful in the analysis and design of linear systems. However, in the nonlinear case, the known algebraic rank conditions still exhibit gaps between necessary and sufficient conditions for controllability and optimality. Also, new, not yet fully understood, topological and resonance obstructions stand in the way of

controllability implying stabilizability. Systems that exhibit special structure, such as living on Lie groups, or being second order such as typical mechanical systems, are amenable to further refinements of the theory; compare, e.g., the use of affine connections and the symmetric product in Bullo et al. (2000). Other directions of ongoing and future research involve the extension of Lie algebraic methods to infinite dimensional systems and to generalize formulas to systems with less regularity; see, e.g., the work by Rampazzo and Sussmann (2007) on Lipschitz vector fields, thereby establishing closer connections with nonsmooth analysis (Clarke 1983) in control.

Cross-References

- [Differential Geometric Methods in Nonlinear Control](#)
- [Feedback Linearization of Nonlinear Systems](#)
- [Lie Algebraic Methods in Nonlinear Control](#)

Bibliography

- Agrachev AA, Stefani G, Zezza P (2002) Strong optimality for a bang-bang trajectory. *SIAM J Control Optim.* 41:991–1014 (electronic)
- Bloch AM (2003) *Nonholonomic mechanics and control*. Interdisciplinary applied mathematics, vol 24. Springer, New York. With the collaboration of J. Baillieul, P. Crouch and J. Marsden, With scientific input from P. S. Krishnaprasad, R. M. Murray and D. Zenkov, Systems and Control.
- Bressan A, Piccoli B (2007) *Introduction to the mathematical theory of control*. AIMS series on applied mathematics, vol 2. American Institute of Mathematical Sciences (AIMS), Springfield
- Bullo F, Lewis AD (2005) *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems*. Texts in applied mathematics, vol 49. Springer, New York
- Bullo F, Leonard NE, Lewis AD (2000) Controllability and motion algorithms for underactuated Lagrangian systems on Lie groups: mechanics and nonlinear control systems. *IEEE Trans Autom Control* 45:1437–1454
- Clarke FH (1983) *Optimization and nonsmooth analysis*. Canadian mathematical society series of monographs and advanced texts. Wiley, New York. A Wiley-Interscience Publication

- Fliess M (1980) Realizations of nonlinear systems and abstract transitive Lie algebras. *Bull Am Math Soc (N.S.)* 2:444–446
- Gehrig E, Kawski M (2008) A hopf-algebraic formula for compositions of noncommuting flows. In: *Proceedings IEEE conference decision and control*, Cancun, pp 156–1574
- Gray WS, Wang Y (2002) Fliess operators on L_p spaces: convergence and continuity. *Syst Control Lett* 46:67–74
- Isidori A (1995) *Nonlinear control systems*. Communications and control engineering series, 3rd edn. Springer, Berlin
- Jakubczyk B (1986) Local realizations of nonlinear causal operators. *SIAM J Control Optim* 24:230–242
- Kawski M (1990) High-order small-time local controllability. In: *Nonlinear controllability and optimal control*. Monographs and textbooks in pure and applied mathematics, vol 133. Dekker, New York, pp 431–467
- Rampazzo F, Sussmann HJ (2007) Commutators of flow maps of nonsmooth vector fields. *J Differ Equ* 232:134–175
- Sontag ED (1998) *Mathematical control theory*. Texts in applied mathematics, vol 6, 2nd edn. Springer, New York. Deterministic finite-dimensional systems
- Sussmann HJ (1974) An extension of a theorem of Nagano on transitive Lie algebras. *Proc Am Math Soc* 45:349–356
- Sussmann HJ (1986) A product expansion for the Chen series. In: *Theory and applications of nonlinear control systems* (Stockholm, 1985). North-Holland, Amsterdam, pp 323–335
- Sussmann HJ (1987) A general theorem on local controllability. *SIAM J Control Optim* 25:158–194

Linear Matrix Inequality Techniques in Optimal Control

Robert E. Skelton
Texas A&M University, College Station, TX,
USA

Abstract

LMI (linear matrix inequality) techniques offer more flexibility in the design of dynamic linear systems than techniques that minimize a scalar functional for optimization. For linear state space models, multiple goals (performance bounds) can be characterized

in terms of LMIs, and these can serve as the basis for controller optimization via finite-dimensional convex feasibility problems. LMI formulations of various standard control problems are described in this entry, including dynamic feedback stabilization, covariance control, LQR, H_∞ control, and L_∞ control. The integration of control and information architecture design (i.e., adding sensor/actuator precision selection to the control problem) is also shown to be a convex problem by formulating the feasibility constraints as LMIs.

Keywords

Control system design · Covariance control · H_∞ control · L_∞ control · LQR/LQG · Matrix inequalities · Sensor/actuator design · Integrated plant and control design

Early Optimization History

Hamilton invented state space models of nonlinear dynamic systems with his generalized momenta work in the 1800s (Hamilton 1834, 1835), but at that time the lack of computational tools prevented broad acceptance of the first-order form of dynamic equations. With the rapid development of computers in the 1960s, state space models evoked a formal control theory for minimizing a scalar function of control and state, propelled by the calculus of variations and Pontryagin's maximum principle. Optimal control has been a pillar of control theory for the last 50 years. In fact, all of the problems discussed in this entry might be solvable by minimizing a scalar functional, but an unsolved search is required to find the right functional (e.g., finding weights in an LQG formulation). Globally convergent algorithms are available to do just that for quadratic functionals, where an iterative algorithm with proven convergence can find the weights, and each iteration requires solving new Riccati equations. However, more direct methods are now available to do this just using LMIs.

Since the 1990s, a major focus for linear system design has been to pose control problems as feasibility problems, to satisfy multiple constraints. Since then, feasibility approaches have dominated design decisions, and such feasibility problems may be convex or non-convex. If the constraint problem can be reduced to a set of linear matrix inequalities (LMIs), then convexity is proven. However, failure to find such LMI formulations of the problem does not prove the problem is not convex. Computer-assisted methods for convex problems are available to avoid the search for LMIs (see Camino et al. 2003a).

In the case of linear dynamic models of stochastic processes, optimization methods led to the popularization of linear quadratic Gaussian (LQG) optimal control, which had globally optimal solutions (see Skelton 1988). The first two moments of the stochastic process (the mean and the trace of covariance) can be controlled with these methods, even if the distribution of the random variables involved is not Gaussian. Hence, LQG became just an acronym for the solution of quadratic functionals of control and state variables, even when the stochastic processes were not Gaussian. The label LQG was often used even for deterministic problems, where a time integral, rather than an expectation operator, was minimized, with given initial conditions or impulse excitations. These were formally called linear quadratic regulator (LQR) problems. Later the book Skelton (1988) gave the formal conditions under which the LQG and the LQR answers were numerically identical, and this particular version of LQR was called the *deterministic LQG*.

It was always recognized that the quadratic form of the state and control in the LQG problem was an artificial goal. The real control goals usually involved prespecified performance bounds on *each* of the outputs and bounds on *each* channel of the control and/or sensor channels. This leads to matrix inequalities (MIs) rather than a scalar minimization. While it was known early that *any* stabilizing linear controller could be obtained by some choice of weights in an LQG optimization problem (see Chap. 6 and references

in Skelton 1988), it was not known until the 1980s *what* particular choice of weights in LQG would yield a solution to multiple performance bounds on output and control (the matrix inequality (MI) problem). See early attempts in Skelton (1988), and see Zhu and Skelton (1992) and Zhu et al. (1997) for a globally convergent algorithm to find such LQG weights when the MI problem has a solution. Since then, rather than stating a minimization problem for a meaningless sum of outputs and inputs, linear control problems can now be stated simply in terms of norm bounds on *each* input vector and/or *each* output vector of the system (L_2 bounds, L_∞ bounds, or variance and covariance bounds). These advances led to the theory of covariance control. These *feasibility* problems are convex for state feedback or full-order output feedback controllers (the focus of this elementary introduction), and these can be solved using linear matrix inequalities (LMIs), as illustrated in this entry.

Perhaps the earliest book on LMI control methods was Boyd et al. (1994), but the results and notations used herein are taken from Skelton et al. (1998). Other important LMI papers and books can give the reader a broader background, including Iwasaki and Skelton (1994), Gahinet and Apkarian (1994), Scherer (1995); Scherer et al. (1997), de Oliveira et al. (2002), Li et al. (2008), de Oliveira and Skelton (2001), Camino et al. (2001, 2003a), Boyd and Vandenberghe (2004), Iwasaki et al. (2000), Khargonekar and Rotea (1991), Balakrishnan et al. (1994), Vandenberghe and Boyd (1996), Gahinet et al. (1995), and Dullerud and Paganini (2000).

Linear Matrix Equalities and Inequalities

A matrix equality is called linear matrix equality (LME) if the unknown matrix appears linearly in the equation. A linear matrix equality (LME) " $\mathbf{TGA} = \mathbf{\Theta}$ " may or may not have a solution for $\mathbf{G}$. For LMEs, there are only two questions to answer: (i) "Does there exist a solution?" and (ii) "What is the set of all solutions?" The answer to

question (i) is: “There exists a solution if and only if $\Gamma\Gamma^+ \Theta \Lambda^+ \Lambda = \Theta$.” The answer to question (ii) is: “ $\mathbf{G} = \Gamma^+ \Theta \Lambda^+ + \mathbf{Z} - \Gamma^+ \Gamma \mathbf{Z} \Lambda \Lambda^+$, where $\mathbf{Z}$ is arbitrary, and the $+$ symbol denotes Pseudo-Inverse.” LMI approaches employ the same two questions as LMEs, by formulating the necessary and sufficient conditions for the existence of an LMI solution and then to parametrize all solutions.

First, any matrix inequality is just a short-hand notation to represent a certain scalar inequality. For any square matrix $\mathbf{Q}$, the matrix inequality “ $\mathbf{Q} > \mathbf{0}$ ” means “the scalar $\mathbf{x}^T \mathbf{Q} \mathbf{x}$ is positive for all values of $\mathbf{x}$, except $\mathbf{x} = \mathbf{0}$.” Obviously this is a property of $\mathbf{Q}$, not $\mathbf{x}$, hence the abbreviated matrix notation $\mathbf{Q} > \mathbf{0}$. This matrix inequality is a linear matrix inequality (LMI), since the matrix unknown $\mathbf{Q}$ appears linearly in the inequality $\mathbf{Q} > \mathbf{0}$. Note also that any square matrix $\mathbf{Q}$ can be written as the sum of a symmetric matrix $\mathbf{Q}_s = \frac{1}{2}(\mathbf{Q} + \mathbf{Q}^T)$, and a skew-symmetric matrix $\mathbf{Q}_a = \frac{1}{2}(\mathbf{Q} - \mathbf{Q}^T)$, but $\mathbf{x}^T \mathbf{Q}_a \mathbf{x} = 0$, so only the symmetric part of the matrix $\mathbf{Q}$ affects the scalar $\mathbf{x}^T \mathbf{Q} \mathbf{x}$. We assume hereafter without loss of generality that $\mathbf{Q}$ is symmetric. The notation “ $\mathbf{Q} \geq \mathbf{0}$ ” means “the scalar $\mathbf{x}^T \mathbf{Q} \mathbf{x}$ cannot be negative for any $\mathbf{x}$.”

The answers to two questions of existence condition and set of all solutions for a particular LMI are summarized here. For any given matrices Γ , Λ , and Θ , the LMI :

$$\Gamma \mathbf{G} \Lambda + (\Gamma \mathbf{G} \Lambda)^T + \Theta < \mathbf{0}, \quad (1)$$

has a solution for matrix $\mathbf{G}$ if and only if the following two conditions hold:

$$\mathbf{U}_\Gamma^T \Theta \mathbf{U}_\Gamma < \mathbf{0}, \quad \text{or} \quad \Gamma \Gamma^T > \mathbf{0}, \quad (2)$$

$$\mathbf{V}_\Lambda^T \Theta \mathbf{V}_\Lambda < \mathbf{0}, \quad \text{or} \quad \Lambda^T \Lambda > \mathbf{0}. \quad (3)$$

where the left (right) null spaces of any matrix $\mathbf{B}$ is defined by matrices $\mathbf{U}_B$ ($\mathbf{V}_B$), where $\mathbf{U}_B^T \mathbf{B} = \mathbf{0}$, $\mathbf{U}_B^T \mathbf{U}_B > \mathbf{0}$, ($\mathbf{B} \mathbf{V}_B = \mathbf{0}$, $\mathbf{V}_B^T \mathbf{V}_B > \mathbf{0}$). If $\mathbf{G}$ exists, then one set of such $\mathbf{G}$ is given by

$$\mathbf{G} = -\rho \Gamma^T \Phi \Lambda^T (\Lambda \Phi \Lambda^T)^{-1}, \quad \Phi = (\rho \Gamma \Gamma^T - \Theta)^{-1}, \quad (4)$$

where $\rho > 0$ is an arbitrary scalar such that

$$\Phi = (\rho \Gamma \Gamma^T - \Theta)^{-1} > \mathbf{0}. \quad (5)$$

All $\mathbf{G}$ which solve the problem are given by Theorem 2.3.12 in Skelton et al. (1998). As elaborated in Chap. 9 of Skelton et al. (1998), 17 different control problems reduce to this *same* matrix inequality (1). Several examples from Skelton et al. (1998) are given in Sect. 4.

Lyapunov Equations and Related LMIs

Lyapunov proved that $\mathbf{x}(t)$ converges to zero if there exists a matrix $\mathbf{Q}$ such that two scalars have the property, $\mathbf{x}(t)^T \mathbf{Q} \mathbf{x}(t) > 0$ and $d/dt(\mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t)) < 0$, along the nonzero trajectory of a dynamic system (e.g., the system $\dot{\mathbf{x}} = \mathbf{A} \mathbf{x}$). That is, for any initial condition $\mathbf{x}(0)$ of the system $\dot{\mathbf{x}} = \mathbf{A} \mathbf{x}$, the state $\mathbf{x}(t)$ converges to zero, if and only if, there exists a matrix $\mathbf{Q}$ such that $\mathbf{Q} > \mathbf{0}$ and $\mathbf{Q} \mathbf{A} + \mathbf{A}^T \mathbf{Q} < \mathbf{0}$. The above two inequalities are LMIs (in the unknown matrix $\mathbf{Q}$), which proves the stability of the system.

LMIs are prevalent throughout the fundamental concepts of linear control theory, ranging from stability to controllability and observability, to robust and covariance control. For the linear system example, $\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u}$ and $\mathbf{y} = \mathbf{C} \mathbf{x}$, the “Controllability Gramian”: $\mathbf{X} = \int_0^\infty e^{\mathbf{A}t} \mathbf{B} \mathbf{B}^T e^{\mathbf{A}^T t} dt > \mathbf{0}$ if and only if the pair $(\mathbf{A}, \mathbf{B})$ is controllable and $\mathbf{X}$ is bounded only if the controllable modes are asymptotically stable. If $\mathbf{X}$ exists, it satisfies $\mathbf{X} \mathbf{A}^T + \mathbf{A} \mathbf{X} + \mathbf{B} \mathbf{B}^T = \mathbf{0}$, and $\mathbf{X} > \mathbf{0}$ if and only if $(\mathbf{A}, \mathbf{B})$ is a controllable pair. Likewise, if the “Observability Gramian” $\mathbf{Q} = \int_0^\infty e^{\mathbf{A}^T t} \mathbf{C}^T \mathbf{C} e^{\mathbf{A}t} dt > \mathbf{0}$ exists, it satisfies $\mathbf{Q} \mathbf{A} + \mathbf{A}^T \mathbf{Q} + \mathbf{C}^T \mathbf{C} = \mathbf{0}$ if and only if the matrix pair $(\mathbf{A}, \mathbf{C})$ is observable. Note also that the matrix pair $(\mathbf{A}, \mathbf{B})$ is controllable for any $\mathbf{A}$ if $\mathbf{B} \mathbf{B}^T > \mathbf{0}$, and the matrix pair $(\mathbf{A}, \mathbf{C})$ is observable for any $\mathbf{A}$ if $\mathbf{C}^T \mathbf{C} > \mathbf{0}$. Hence, the existence of $\mathbf{Q} > \mathbf{0}$ or $\mathbf{X} > \mathbf{0}$ satisfying either $(\mathbf{Q} \mathbf{A} + \mathbf{A}^T \mathbf{Q} < \mathbf{0})$ or $(\mathbf{A} \mathbf{X} + \mathbf{X} \mathbf{A}^T < \mathbf{0})$ is equivalent to the asymptotic stability of the

system. The same LMIs were obtained using the Lyapunov stability results.

It should now be clear that the set of all stabilizing state feedback controllers, $\mathbf{u} = \mathbf{G}\mathbf{x}$, is parametrized by the inequalities $\mathbf{Q} > \mathbf{0}$, $\mathbf{Q}(\mathbf{A} + \mathbf{B}\mathbf{G}) + (\mathbf{A} + \mathbf{B}\mathbf{G})^T\mathbf{Q} < \mathbf{0}$. This is a matrix inequality (product of the two unknowns $\mathbf{Q}$ and $\mathbf{G}$) which can be converted to an LMI using standard techniques.

Other than providing Controllability Gramian and the condition for asymptotic stability, the Lyapunov equation $\mathbf{X}\mathbf{A}^T + \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{B}^T = \mathbf{0}$ has two more interpretations.

1. **Stochastic Interpretations** : The matrix solution $\mathbf{X}$ for equation $\mathbf{X}\mathbf{A}^T + \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{B}^T = \mathbf{0}$ can represent the steady-state covariance matrix ($\mathbf{X} = \mathbb{E}[\mathbf{x}\mathbf{x}^T]$) in the presence of zero mean white noise disturbance of unit intensity $\mathbf{w}$ in $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{w}$. Moreover, the LMI $\bar{\mathbf{X}}\mathbf{A}^T + \mathbf{A}\bar{\mathbf{X}} + \mathbf{B}\mathbf{B}^T < \mathbf{0}$ can be solved to bound the steady-state covariance matrix $\mathbf{X} < \bar{\mathbf{X}}$, which in turn can be used to bound the steady-state output covariance $\mathbf{Y} = \mathbb{E}[\mathbf{y}\mathbf{y}^T] = \mathbf{C}\mathbf{X}\mathbf{C}^T < \bar{\mathbf{Y}}$ for the output equation $\mathbf{y} = \mathbf{C}\mathbf{x}$.
2. **Deterministic Interpretations** : The matrix $\mathbf{X}$ can also be interpreted as

$$\mathbf{X} = \sum_{i=1}^{n_w} \int_0^\infty \mathbf{x}(i, t) \mathbf{x}^T(i, t) dt,$$

where $\mathbf{x}(i, t)$ represents the solution of the state equation when only the i^{th} excitation ($\mathbf{w}_i = \delta(t)$) is applied, for the system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{w}$. Notice that $tr(\mathbf{X}) = \sum_{i=1}^{n_w} \|\mathbf{x}(i, \cdot)\|_{L_2}^2$, where $tr(\cdot)$ is the trace operator and $\|\mathbf{x}\|_{L_2}^2$ represents the energy in the signal $\mathbf{x}$. Also notice that $tr(\mathbf{C}\mathbf{X}\mathbf{C}^T) = \sum_{i=1}^{n_w} \|\mathbf{y}(i, \cdot)\|_{L_2}^2$, where $\mathbf{y}(i, \cdot)$ is the output signal for $\mathbf{y} = \mathbf{C}\mathbf{x}$. A time-domain interpretation of the H_2 norm of the transfer matrix $\mathbf{T}(s)$ is also given by $\|\mathbf{T}\|_{H_2}^2 = \sum_{i=1}^{n_w} \|\mathbf{y}(i, \cdot)\|_{L_2}^2 = tr(\mathbf{C}\mathbf{X}\mathbf{C}^T)$. This interpretation forms the basis to provide solutions to LQR, H_2 , and L_2 control problems.

For a rigorous equivalence of the deterministic and stochastic interpretations, see Skelton (1988). These different interpretations of the Lyapunov equation led to solve many control problems using the same LMI (Skelton et al. 1998).

Many Control Problems Reduce to the Same LMI

Consider the feedback control system

$$\begin{bmatrix} \dot{\mathbf{x}}_p \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_p & \mathbf{D}_p & \mathbf{B}_p \\ \mathbf{C}_p & \mathbf{D}_y & \mathbf{B}_y \\ \mathbf{M}_p & \mathbf{D}_z & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_p \\ \mathbf{w} \\ \mathbf{u} \end{bmatrix}, \quad (6)$$

$$\begin{bmatrix} \mathbf{u} \\ \dot{\mathbf{x}}_c \end{bmatrix} = \begin{bmatrix} \mathbf{D}_c & \mathbf{C}_c \\ \mathbf{B}_c & \mathbf{A}_c \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{x}_c \end{bmatrix} = \mathbf{G} \begin{bmatrix} \mathbf{z} \\ \mathbf{x}_c \end{bmatrix}, \quad (7)$$

where $\mathbf{z}$ is the measurement vector, $\mathbf{y}$ is the output to be controlled, $\mathbf{u}$ is the control vector, $\mathbf{x}_p$ is the plant state vector, $\mathbf{x}_c$ is the state of the controller, and $\mathbf{w}$ is the external disturbance. By defining the matrices

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} \mathbf{x}_p \\ \mathbf{x}_c \end{bmatrix}, \quad \begin{bmatrix} \mathbf{A}_{cl} & \mathbf{B}_{cl} \\ \mathbf{C}_{cl} & \mathbf{D}_{cl} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A} & \mathbf{D} \\ \mathbf{C} & \mathbf{F} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{H} \end{bmatrix} \mathbf{G} [\mathbf{M} \mathbf{E}], \end{aligned} \quad (8)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix},$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_p & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{D}_p \\ \mathbf{0} \end{bmatrix}, \mathbf{E} = \begin{bmatrix} \mathbf{D}_z \\ \mathbf{0} \end{bmatrix} \quad (9)$$

$$\mathbf{C} = [\mathbf{C}_p \mathbf{0}], \quad \mathbf{H} = [\mathbf{B}_y \mathbf{0}], \quad \mathbf{F} = \mathbf{D}_y, \quad (10)$$

one can write the closed-loop system dynamics in the form

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{cl} & \mathbf{B}_{cl} \\ \mathbf{C}_{cl} & \mathbf{D}_{cl} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{w} \end{bmatrix}. \quad (11)$$

Often we seek to characterize the set of all controllers $\mathbf{G}$ to satisfy the given upper bounds on the output and input covariances, $\mathbb{E}[\mathbf{y}\mathbf{y}^T] \leq \bar{\mathbf{Y}}$, $\mathbb{E}[\mathbf{u}\mathbf{u}^T] \leq \bar{\mathbf{U}}$. Such problems are called *covariance* control problems (see Chap. 6 of Skelton

et al. 1998), where $\mathbb{E}$ represents the steady-state expectation operator in the stochastic case (i.e., when $\mathbf{w}$ is white noise), and in the deterministic case, $\mathbb{E}$ represents the infinite integral of the matrix $[\mathbf{y}\mathbf{y}^T]$. The math remains the same in each case, with appropriate interpretations of certain matrices as explained in the previous section.

Moreover, the *same* math, the *same* matrix inequality “ $\mathbf{\Gamma}\mathbf{G}\mathbf{\Lambda} + (\mathbf{\Gamma}\mathbf{G}\mathbf{\Lambda})^T + \mathbf{\Theta} < \mathbf{0}$ ” solves 17 different control problems (using either state feedback or full-order dynamic controllers) by defining the appropriate $\mathbf{\Theta}$, $\mathbf{\Lambda}$, and $\mathbf{\Gamma}$. This includes the characterization of all stabilizing controllers, covariance control, H_∞ control, L_∞ control, LQG control, and H_2 control, including their discrete versions. Several examples from Skelton et al. (1998) follow.

Stabilizing Control

There exists a controller $\mathbf{G}$ that stabilizes the system (6) if and only if (2) and (3) hold, where the matrices are defined by

$$[\mathbf{\Gamma}|\mathbf{\Lambda}^T|\mathbf{\Theta}] = [\mathbf{B}|\mathbf{X}\mathbf{M}^T|\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T]. \quad (12)$$

One can also write such results in another way, as in Corollary 6.2.1 of Skelton et al. (1998, p. 135): There exists a control of the form $\mathbf{u} = \mathbf{G}\mathbf{x}$ that can stabilize the system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ if and only if there exists a matrix $\mathbf{X} > \mathbf{0}$ satisfying $\mathbf{B}^\perp(\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T)(\mathbf{B}^\perp)^T < \mathbf{0}$, where $\mathbf{B}^\perp$ denotes the left null space of $\mathbf{B}$. In this case all stabilizing controllers may be parametrized by $\mathbf{G} = -\mathbf{B}^T\mathbf{P} + \mathbf{L}\mathbf{Q}^{1/2}$, for any $\mathbf{Q} > \mathbf{0}$ and a $\mathbf{P} > \mathbf{0}$ satisfying $\mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P} - \mathbf{P}\mathbf{B}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = \mathbf{0}$. The matrix $\mathbf{L}$ is any matrix that satisfies the norm bound $\|\mathbf{L}\| < 1$. Youla et al. (1976) provided a parametrization of the set of all stabilizing controllers, but the parametrization was infinite dimensional (as it did not impose any restriction on the order or form of the controller). So for finite calculations, one had to truncate the set to a finite number before optimization or stabilization started. As noted above, on the other hand, all stabilizing state feedback controllers $\mathbf{G}$ can be parametrized in terms of an arbitrary but finite-dimensional norm-bounded matrix $\mathbf{L}$. Similar results apply for

the dynamic controllers of any fixed order (see Chap. 6 in Skelton et al. 1998).

Covariance Upper Bound Control

In the system (6), suppose that $\mathbf{D}_y = \mathbf{0}$, $\mathbf{B}_y = \mathbf{0}$ and that $\mathbf{w}$ is zero-mean white noise with intensity $\mathbf{I}$. Let a required upper bound $\bar{\mathbf{Y}} > \mathbf{0}$ on the steady-state output covariance $\mathbf{Y} = \mathbb{E}[\mathbf{y}\mathbf{y}^T]$ be given. The following statements are equivalent:

- (i) There exists a controller $\mathbf{G}$ that solves the covariance upper bound control problem $\mathbf{Y} < \bar{\mathbf{Y}}$.
- (ii) There exists a matrix $\mathbf{X} > \mathbf{0}$ such that $\mathbf{Y} = \mathbf{C}\mathbf{X}\mathbf{C}^T < \bar{\mathbf{Y}}$ and (2) and (3) hold, where the matrices are defined by

$$[\mathbf{\Gamma}|\mathbf{\Lambda}^T|\mathbf{\Theta}] = \begin{bmatrix} \mathbf{B} & \mathbf{X}\mathbf{M}^T & \mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T & \mathbf{D} \\ \mathbf{0} & \mathbf{E}^T & \mathbf{D}^T & -\mathbf{I} \end{bmatrix}. \quad (13)$$

Proof is provided by Theorem 9.1.2 in Skelton et al. (1998).

Linear Quadratic Regulator

Consider the linear time-invariant system (6). Suppose that $\mathbf{D}_y = \mathbf{0}$, $\mathbf{D}_z = \mathbf{0}$ and that $\mathbf{w}$ is the impulsive disturbance $\mathbf{w}(t) = \mathbf{w}_0\delta(t)$. Let a performance bound $\gamma > 0$ be given, where the required performance is to keep the integral squared output less than the prespecified value $\|\mathbf{y}\|_{L_2} < \gamma$ for any vector $\mathbf{w}_0$ such that $\mathbf{w}_0^T\mathbf{w}_0 \leq 1$, and $\mathbf{x}_0 = \mathbf{0}$. This problem is labeled as linear quadratic regulator (LQR). The following statements are equivalent:

- (i) There exists a controller $\mathbf{G}$ that solves the LQR problem.
- (ii) There exists a matrix $\mathbf{Y} > \mathbf{0}$ such that $\|\mathbf{D}^T\mathbf{Y}\mathbf{D}\| < \gamma^2$ and (2) and (3) hold, where the matrices are defined by

$$[\mathbf{\Gamma}|\mathbf{\Lambda}^T|\mathbf{\Theta}] = \begin{bmatrix} \mathbf{Y}\mathbf{B} & \mathbf{M}^T & \mathbf{Y}\mathbf{A} + \mathbf{A}^T\mathbf{Y} & \mathbf{M}^T \\ \mathbf{H} & \mathbf{0} & \mathbf{M} & -\mathbf{I} \end{bmatrix}. \quad (14)$$

Proof is provided by Theorem 9.1.3 in Skelton et al. (1998).

H-Infinity Control

LMI techniques provided the first papers to solve the general H_∞ problem, without any restrictions on the plant. See Iwasaki and Skelton (1994) and Gahinet and Apkarian (1994). Let the closed-loop transfer matrix from $\mathbf{w}$ to $\mathbf{y}$ with the controller in (6) be denoted by $\mathbf{T}(s)$:

$$\mathbf{T}(s) = \mathbf{C}_{cl}(s\mathbf{I} - \mathbf{A}_{cl})^{-1}\mathbf{B}_{cl} + \mathbf{D}_{cl}. \quad (15)$$

The H_∞ control problem can be defined as follows:

Let a performance bound $\gamma > 0$ be given. Determine whether or not there exists a controller $\mathbf{G}$ in (6) which asymptotically stabilizes the system and yields the closed-loop transfer matrix (15) such that the peak value of the frequency response is less than γ . That is, $\|\mathbf{T}\|_{H_\infty} = \sup \|\mathbf{T}(j\omega)\| < \gamma$.

The following statements are equivalent:

- (i) A controller $\mathbf{G}$ solves the H_∞ control problem.
- (ii) There exists a matrix $\mathbf{X} > \mathbf{0}$ such that (2) and (3) hold, where

$$[\Gamma | \Lambda^T | \Theta] = \begin{bmatrix} \mathbf{B} & \mathbf{X}\mathbf{M}^T & \mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T & \mathbf{X}\mathbf{C}^T & \mathbf{D} \\ \mathbf{H} & \mathbf{0} & \mathbf{C}\mathbf{X} & -\gamma\mathbf{I} & \mathbf{F} \\ \mathbf{0} & \mathbf{E}^T & \mathbf{D}^T & \mathbf{F}^T & -\gamma\mathbf{I} \end{bmatrix}. \quad (16)$$

Proof is provided by Theorem 9.1.5 in Skelton et al. (1998).

L-Infinity Control

The peak value of the frequency response is controlled by the above H_∞ controller. A similar theorem can be written to control the peak in the time domain. Define $\sup[\mathbf{y}(t)^T\mathbf{y}(t)] = \|\mathbf{y}\|_{L_\infty}^2$, and let the statement $\|\mathbf{y}\|_{L_\infty} < \gamma$ mean that the peak value of $[\mathbf{y}(t)^T\mathbf{y}(t)]$ is less than γ^2 . Suppose that $\mathbf{D}_y = \mathbf{0}$ and $\mathbf{B}_y = \mathbf{0}$. There exists a controller $\mathbf{G}$ which maintains $\|\mathbf{y}\|_{L_\infty} < \gamma$ in the presence of any energy-bounded input $\mathbf{w}(t)$ (i.e., $\int_0^\infty \mathbf{w}^T\mathbf{w}dt \leq 1$) if and only if there exists a matrix $\mathbf{X} > \mathbf{0}$ such that $\mathbf{C}\mathbf{X}\mathbf{C}^T < \gamma^2\mathbf{I}$ and (2) and (3) hold, where

$$[\Gamma | \Lambda^T | \Theta] = \begin{bmatrix} \mathbf{B} & \mathbf{X}\mathbf{M}^T & \mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T & \mathbf{D} \\ \mathbf{0} & \mathbf{E}^T & \mathbf{D}^T & -\gamma\mathbf{I} \end{bmatrix}. \quad (17)$$

Proof is provided by Theorem 9.1.4 in Skelton et al. (1998). Note that the L_∞ control and covariance upper bound control have exactly the same final mathematical result as explained in Sect. 3.

Information Architecture in Estimation and Control Problems

In a typical “control problem” that occupies most research literature, the sensors and actuators have already been selected. Yet the selection of sensors and actuators and their locations greatly affect the ability of the control system to do its job efficiently. Perhaps in one location a high-precision sensor is needed, and in another location high precision is not needed, and paying for high precision in that location would, therefore, be a waste of resources. These decisions must be influenced by the control dynamics which are yet to be designed. How does one know where to effectively spend money to improve the system? To answer this question, we must optimize the information architecture (sensor/actuator precision) jointly with the control law.

Traditionally, with full-order controllers, and *prespecified* sensor/actuator instruments (with specified precisions), the problem of designing a controller to constrain the upper bounds on the covariance of output error and control signal is a well-known solved convex problem (which means it can be converted to an LMI problem if desired), see Scherer et al. (1997) and Chap. 6 of Skelton et al. (1998). If we enlarge the domain of design freedom to include sensor/actuator precisions, it is not obvious whether the feasibility problem is convex or not. The following shows that this problem of including the sensor/actuator precisions within the control design problem is indeed convex (Li et al. 2008).

Let us consider the linear control system (6)–(11). Suppose that $\mathbf{D}_y = \mathbf{0}$, $\mathbf{B}_y = \mathbf{0}$, $\mathbf{D}_p = [\mathbf{D}_{pn} \ \mathbf{D}_a \ 0]$, and $\mathbf{D}_z = [0 \ 0 \ \mathbf{D}_s]$.

The vector $\mathbf{w}^T = [\mathbf{w}_p^T \ \mathbf{w}_a^T \ \mathbf{w}_s^T]$ contains process noise $\mathbf{w}_p$, actuator noise $\mathbf{w}_a$, and sensor noise $\mathbf{w}_s$. These disturbances are modeled as independent zero mean white noises with intensities $\mathbf{W}_p$, $\mathbf{W}_a$, and $\mathbf{W}_s$, respectively. The process noise intensity $\mathbf{W}_p$ is assumed to be known and fixed. The actuator and sensor precisions are defined as the inverse of the noise intensity (or variance in the discrete-time case) in each channel:

$$\text{diag}(\boldsymbol{\gamma}_a) \triangleq \boldsymbol{\Gamma}_a \triangleq \mathbf{W}_a^{-1}, \quad \text{diag}(\boldsymbol{\gamma}_s) \triangleq \boldsymbol{\Gamma}_s \triangleq \mathbf{W}_s^{-1}. \quad (18)$$

Assume that the price of each actuator/sensor is proportional to the precision associated with that instrument. Therefore, the total cost of all actuators and sensors is

$$\$ = \mathbf{p}_a^T \boldsymbol{\gamma}_a + \mathbf{p}_s^T \boldsymbol{\gamma}_s, \quad (19)$$

where $\mathbf{p}_a$ and $\mathbf{p}_s$ are vectors containing the price per unit of actuator precision and sensor precision, respectively.

Let $\bar{\$}$ represent the budget; the allowed upper bound on sensor/actuator costs, and $\bar{\boldsymbol{\gamma}}_a$ and $\bar{\boldsymbol{\gamma}}_s$ represent the limit on available precisions of actuators and sensors. Then, there exists a dynamic controller $\mathbf{G}$ and choices of sensor/actuator precisions $\boldsymbol{\Gamma}_s$, $\boldsymbol{\Gamma}_a$ that satisfy the constraints

$$\begin{aligned} \$ &< \bar{\$}, \quad \boldsymbol{\gamma}_a < \bar{\boldsymbol{\gamma}}_a, \quad \boldsymbol{\gamma}_s < \bar{\boldsymbol{\gamma}}_s, \\ \mathbb{E}[\mathbf{u}\mathbf{u}^T] &< \bar{\mathbf{U}}, \quad \mathbb{E}[\mathbf{y}\mathbf{y}^T] < \bar{\mathbf{Y}}, \end{aligned} \quad (20)$$

if and only if there exist matrices $\mathbf{L}$, $\mathbf{F}$, $\mathbf{Q}$, $\mathbf{X}$, and $\mathbf{Z}$ and vectors $\boldsymbol{\gamma}_a$ and $\boldsymbol{\gamma}_s$ such that the following LMIs are satisfied

$$\begin{aligned} \mathbf{p}_a^T \boldsymbol{\gamma}_a + \mathbf{p}_s^T \boldsymbol{\gamma}_s &< \bar{\$}, \\ \boldsymbol{\gamma}_a &< \bar{\boldsymbol{\gamma}}_a, \quad \boldsymbol{\gamma}_s < \bar{\boldsymbol{\gamma}}_s, \\ \begin{bmatrix} \bar{\mathbf{Y}} & \mathbf{C}_p \mathbf{X} \mathbf{C}_p \\ (\mathbf{C}_p \mathbf{X})^T & \mathbf{X} & \mathbf{I} \\ \mathbf{C}_p^T & \mathbf{I} & \mathbf{Z} \end{bmatrix} &> \mathbf{0}, \end{aligned} \quad (21)$$

$$\begin{bmatrix} \bar{\mathbf{U}} & \mathbf{L} & \mathbf{0} \\ \mathbf{L}^T & \mathbf{X} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{Z} \end{bmatrix} > \mathbf{0}, \quad \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{12}^T & \Phi_{22} \end{bmatrix} < \mathbf{0}, \quad (22)$$

$$\begin{aligned} \Phi_{11} &= \boldsymbol{\phi} + \boldsymbol{\phi}^T, \\ \boldsymbol{\phi} &= \begin{bmatrix} \mathbf{A}_p \mathbf{X} + \mathbf{B}_p \mathbf{L} & \mathbf{A}_p \\ \mathbf{Q} & \mathbf{Z} \mathbf{A}_p + \mathbf{F} \mathbf{M}_p \end{bmatrix}, \\ \Phi_{12} &= \begin{bmatrix} \mathbf{D}_{pn} & \mathbf{D}_a & \mathbf{0} \\ \mathbf{Z} \mathbf{D}_{pn} & \mathbf{Z} \mathbf{D}_a & \mathbf{F} \mathbf{D}_s \end{bmatrix}, \end{aligned} \quad (23)$$

$$\Phi_{22} = \begin{bmatrix} -\mathbf{W}_p^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{\Gamma}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\boldsymbol{\Gamma}_s \end{bmatrix}. \quad (24)$$

Note that the matrix inequalities (21)–(24) are LMIs in the collection of variables $(\mathbf{L}, \mathbf{F}, \mathbf{Q}, \mathbf{X}, \mathbf{Z}, \boldsymbol{\gamma}_a, \boldsymbol{\gamma}_s)$, hence, the joint control/sensor/actuator design is a convex problem.

After a solution $(\mathbf{L}, \mathbf{F}, \mathbf{Q}, \mathbf{X}, \mathbf{Z}, \boldsymbol{\gamma}_a, \boldsymbol{\gamma}_s)$ is found using the LMIs (21)–(24), then the problem (20) is solved for the controller as

$$\begin{aligned} \mathbf{G} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{V}_l^{-1} & -\mathbf{V}_l^{-1} \mathbf{Z} \mathbf{B}_p \end{bmatrix} \begin{bmatrix} \mathbf{Q} - \mathbf{Z} \mathbf{A}_p \mathbf{X} & \mathbf{F} \\ \mathbf{L} & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{V}_r^{-1} \\ \mathbf{I} & -\mathbf{M}_p \mathbf{X} \mathbf{V}_r^{-1} \end{bmatrix}, \end{aligned} \quad (25)$$

where $\mathbf{V}_l$ and $\mathbf{V}_r$ are left and right factors of the matrix $\mathbf{I} - \mathbf{Z}\mathbf{X}$ (which can be found from the singular value decomposition $\mathbf{I} - \mathbf{Z}\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = (\mathbf{U}\boldsymbol{\Sigma}^{1/2})(\boldsymbol{\Sigma}^{1/2}\mathbf{V}^T) = (\mathbf{V}_l)(\mathbf{V}_r)$).

To emphasize the theme of this entry, to relate optimization to LMIs, we note that three optimization problems present themselves in the above problem with three constraints: control effort $\bar{\mathbf{U}}$, output performance $\bar{\mathbf{Y}}$, and instrument costs $\bar{\$}$. To solve optimization problems, one can fix any two of these prespecified upper bounds and iteratively reduce the level set value of the third “constraint” until feasibility is lost. This process minimizes the resource expressed by the third constraint, while enforcing the other two constraints.

As an example, if the cost is not a concern, one can always set large limits for $\bar{\$}$ and discover the best assignment of sensor/actuator precisions for

the specified performance requirements. The precisions produced by the algorithm are the values $\gamma_{a_i}, \gamma_{s_i}$, produced from the solution (21)–(24), where the observed rankings $\gamma_i > \gamma_j > \gamma_k > \dots$ indicate which sensors or actuators are most critical to the required performance constraints ($\bar{U}, \bar{Y}, \bar{\$}$). If any precision γ_n is essentially zero, compared to other required precisions, then the math is asserting that the information from this sensor (n) is not important for the control objectives specified, or the control signals through this actuator channel (n) are ineffective in controlling the system to these specifications. This information leads us to a technique for choosing the best sensor actuators and their location.

The previous discussion provides the precision (γ_i) required of each sensor and each actuator in the system. Our final application of this theory locates sensors and actuators in a large-scale system, by discarding the least effective ones. While the above method of selecting the precision of the instruments is proven to be a convex problem, the problem of deleting a sensor/actuator is not a convex problem. Suppose we solve the stated feasibility problem above, by starting with the entire admissible set of sensors and actuators. For example, in a flexible structure control problem, we might not know whether to place a rate sensor or displacement sensors at a given location, so we add both. We might not know whether to use torque or force actuators, so we add both. We fill up the system with the feasible set of possibilities and let the above precision rankings (available after the above LMI problem is solved) reveal how much precision is needed at each location and for each sensor/actuator. If the revealed required precision of any sensor or actuator is essentially zero, one can delete this instrument with impunity. Likewise, if the precision required of any instrument is insignificant compared to others (say $\gamma_i > \gamma_j > \gamma_k > \dots \gg \gamma_n$), then we shall take the ad-hoc decision to delete it, even though this step causes our problem of selecting which sensors and actuators to remove from the admissible set to be a non-convex problem. Our method of selecting critical sensors and actuators is summarized as follows: (i) For the entire set of admissible sensors and actuators,

solve the LMIs from the convex “Information Architecture” problem defined above. (ii) Delete the sensor or actuator requiring insignificant precision from step (i), if any. (iii) Repeat the LMI problem in step (i) with one less sensor or actuator. (iv) Continue deleting sensors/actuators in this manner until feasibility of the problem is lost. Then take the result of the previous iteration as the final answer for the specific problem specified by the allowable bounds ($\bar{\$}, \bar{U}, \bar{Y}$).

The major contribution of the above algorithm has been to extend control theory to include “Information Precision and Architecture” in control design and to prove that the information precision and control design is jointly a convex problem. This enlarges the set of solved linear control problems, from solutions of linear controllers with sensors/actuators prespecified to solutions which specify the sensor/actuator requirements jointly with the control solution.

While this section provides a framework for combined instrument and control design, it is possible to also add plant parameters into the domain of the system design problem. Let the vector α represent the plant parameters that will be optimized along with the controller matrix $\mathbf{G}$ and the sensor/actuator precision (γ) to meet certain performance requirements of input/output covariance bounds; as above, this integrated design is not a convex problem, but one can add a convexifying potential function which generates a convex subproblem (Camino et al. 2003b). Then, iteration over that convex subproblem is guaranteed to reach a stationary solution for the following problem (Goyal and Skelton 2019):

$$\mathbf{E}_p(\alpha)\dot{\mathbf{x}}_p = \mathbf{A}_p(\alpha)\mathbf{x}_p + \mathbf{D}_p(\alpha)\mathbf{w} + \mathbf{B}_p\mathbf{u},$$

$$\mathbf{y} = \mathbf{C}_p(\alpha)\mathbf{x}_p, \quad (26)$$

$$\bar{\$} < \$, \gamma_a < \bar{\gamma}_a, \gamma_s < \bar{\gamma}_s, \bar{\alpha}_L < \alpha < \bar{\alpha}_U,$$

$$\mathbb{E}[\mathbf{u}\mathbf{u}^T] < \bar{\mathbf{U}}, \quad \mathbb{E}[\mathbf{y}\mathbf{y}^T] < \bar{\mathbf{Y}}. \quad (27)$$

Summary

Feasibility constraints and LMI techniques provide more powerful tools for designing dynamic

linear systems than techniques that minimize a scalar functional for optimization, since multiple goals (bounds) can be achieved, where *each* of the output and input norms can be bounded. If the problem can be reduced to a set of LMIs to solve, then convexity is proven, and standard LMI software toolboxes are available to solve it. Failure to reduce the problem to LMIs does not imply non-convexity of the problem, but computer-assisted methods for convex problems are available to avoid the search for LMIs. These feasibility problems also allow for optimization. Optimization can also be achieved with LMI methods by reducing the level set for one of the bounds while maintaining all the other bounds. This level-set is iteratively reduced after solving a convex problem at a given level-set. Continue reducing the level-set to find the smallest level-set that allows a solution to the convex problem.

A most amazing fact is that most of the common linear control design problems all reduce to exactly the *same* LMI. Such equivalent problems include: (i) the set of all stabilizing controllers, (ii) LQR, (iii) LQG, (iv) the set of all H_∞ controllers, (v) the set of all L_∞ controllers, (vi) positive real control design, (vii) pole assignment controllers, (viii) covariance control design, and the robust and discrete versions of these problems. Seventeen different control problems have been found to be equivalent to solving exactly the same LMI, by substituting appropriate matrix coefficients.

LMI techniques extend the range of solvable system design problems that are globally solved, and many of these problems are not even control problems. For example, the following problem reduces to an LMI: *Find the best digital computer simulation of a physical system, such that the errors of the simulation (introduced by errors in model order and errors introduced by computational roundoff errors) are limited to a specified covariance upperbound.*

By integrating information architecture and control design, one can use LMIs to simultaneously choose the control gains and the precision required of all sensor/actuators and all computer computations, to satisfy the closed-loop performance constraints. These techniques

can be used to select the information (with precision requirements) required to solve a control or estimation problem, using the most economic solution. Moreover, a suboptimal solution of integrated plant, control, and information architecture design can also be found by iterating over a set of convexified LMIs. This allows for one further step toward a system design approach to integrate multiple disciplines.

For additional discussion on LMI problems in control, the reader is encouraged to consult the following: Dullerud and Paganini (2000), de Oliveira et al. (2002), Li et al. (2008), de Oliveira and Skelton (2001), Gahinet and Apkarian (1994), Iwasaki and Skelton (1994), Camino et al. (2001, 2003a), Skelton et al. (1998), Vandenberghe and Boyd (1996), Boyd et al. (1994), Boyd and Vandenberghe (2004), Balakrishnan et al. (1994), Iwasaki et al. (2000), Khargonekar and Rotea (1991), Scherer (1995), Scherer et al. (1997), and Gahinet et al. (1995).

Cross-References

- [H-Infinity Control](#)
- [H₂ Optimal Control](#)
- [Linear Quadratic Optimal Control](#)
- [LMI Approach to Robust Control](#)
- [Stochastic Linear-Quadratic Control](#)

Bibliography

- Balakrishnan V, Huang Y, Packard A, Doyle JC (1994) Linear matrix inequalities in analysis with multipliers. In: Proceedings of the 1994 American control conference, Baltimore, vol 2, pp 1228–1232
- Boyd SP, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge
- Boyd SP, El Ghaoui L, Feron E, Balakrishnan V (1994) Linear matrix inequalities in system and control theory. SIAM, Philadelphia
- Camino JF, Helton JW, Skelton RE, Ye J (2001) Analysing matrix inequalities systematically: how to get Schur complements out of your life. In: Proceedings of the 5th SIAM conference on control & its applications, San Diego
- Camino JF, Helton JW, Skelton RE, Ye J (2003a) Matrix inequalities: a symbolic procedure to determine convexity automatically. Integral Equ Oper Theory 46(4):399–454

- Camino JF, de Oliveira MC, Skelton RE (2003b) Convexifying linear matrix inequality methods for integrating structure and control design. *J Struct Eng* 129(7): 978–988
- Goyal R, Skelton RE (2019) Joint Optimization of plant, controller, and sensor/actuator design. In: *Proceedings of the 2019 American control conference (ACC)*, Philadelphia. 1507–1512
- de Oliveira MC, Skelton RE (2001) Stability tests for constrained linear systems. In: Reza Moheimani SO (ed) *Perspectives in robust control. Lecture notes in control and information sciences*. Springer, New York, pp 241–257. ISBN:1852334525
- de Oliveira MC, Geromel JC, Bernussou J (2002) Extended H_2 and H_∞ norm characterizations and controller parametrizations for discrete-time systems. *Int J Control* 75(9):666–679
- Dullerud G, Paganini F (2000) *A course in robust control theory: a convex approach*. Texts in applied mathematics. Springer, New York
- Gahinet P, Apkarian P (1994) A linear matrix inequality approach to H_∞ control. *Int J Robust Nonlinear Control* 4(4):421–448
- Gahinet P, Nemirovskii A, Laub AJ, Chilali M (1995) *LMI control toolbox user's guide*. The Mathworks Inc., Natick
- Hamilton WR (1834) On a general method in dynamics; by which the study of the motions of all free systems of attracting or repelling points is reduced to the search and differentiation of one central relation, or characteristic function. *Philos Trans R Soc (Part II)*: 247–308
- Hamilton WR (1835) Second essay on a general method in dynamics. *Philos Trans R Soc (Part I)*:95–144
- Iwasaki T, Skelton RE (1994) All controllers for the general H_∞ control problem – LMI existence conditions and state-space formulas. *Automatica* 30(8):1307–1317
- Iwasaki T, Meinsma G, Fu M (2000) Generalized S-procedure and finite frequency KYP lemma. *Math Probl Eng* 6:305–320
- Khargonekar PP, Rotea MA (1991) Mixed H_2/H_∞ control: a convex optimization approach. *IEEE Trans Autom Control* 39:824–837
- Li F, de Oliveira MC, Skelton RE (2008) Integrating information architecture and control or estimation design. *SICE J Control Meas Syst Integr* 1(2):120–128
- Scherer CW (1995) Mixed H_2/H_∞ control. In: Isidori A (ed) *Trends in control: a European perspective*, pp 173–216. Springer, Berlin
- Scherer CW, Gahinet P, Chilali M (1997) Multiobjective output-feedback control via LMI optimization. *IEEE Trans Autom Control* 42(7):896–911
- Skelton RE (1988) *Dynamics Systems Control: linear systems analysis and synthesis*. Wiley, New York
- Skelton RE, Iwasaki T, Grigoriadis K (1998) *A unified algebraic approach to control design*. Taylor & Francis, London
- Vandenberghe L, Boyd SP (1996) Semidefinite programming. *SIAM Rev* 38:49–95
- Youla DC, Bongiorno JJ, Jabr HA (1976) Modern Wiener-Hopf design of optimal controllers, part II: the multivariable case. *IEEE Trans Autom Control* 21: 319–338
- Zhu G, Skelton R (1992) A two-Riccati, feasible algorithm for guaranteeing output l_∞ constraints. *J Dyn Syst Meas Control* 114(3):329–338
- Zhu G, Rotea M, Skelton R (1997) A convergent algorithm for the output covariance constraint control problem. *SIAM J Control Optim* 35(1):341–361

Linear Quadratic Optimal Control

H.L. Trentelman

Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, Groningen, AV, The Netherlands

Abstract

Linear quadratic optimal control is a collective term for a class of optimal control problems involving a linear input-state-output system and a cost functional that is a quadratic form of the state and the input. The aim is to minimize this cost functional over a given class of input functions. The optimal input depends on the initial condition, but can be implemented by means of a state feedback control law independent of the initial condition. Both the feedback gain and the optimal cost can be computed in terms of solutions of Riccati equations.

Keywords

Algebraic riccati equation · Finite horizon · Infinite horizon · Linear systems · Optimal control · Quadratic cost functional · Riccati differential equation

Introduction

Linear quadratic optimal control is a generic term that collects a number of optimal control problems for linear input-state-output systems in which a quadratic cost functional is minimized

over a given class of input functions. This functional is formed by integrating a quadratic form of the state and the input over a finite or an infinite time interval. Minimizing the energy of the output over a finite or infinite time interval can be formulated in this framework and in fact provides a major motivation for this class of optimal control problems. A common feature of the solutions to the several versions of the problem is that the optimal input functions can be given in the form of a linear state feedback control law. This makes it possible to implement the optimal controllers as a feedback loop around the system. Another common feature is that the optimal value of the cost functional is a quadratic form of the initial condition on the system. This quadratic form is obtained by taking the appropriate solution of a Riccati differential equation or algebraic Riccati equation associated with the system.

Systems with Inputs and Outputs

Consider the continuous-time, linear time-invariant input-output system in state space form represented by

$$\dot{x}(t) = Ax(t) + Bu(t), \quad z(t) = Cx(t) + Du(t). \quad (1)$$

This system will be referred to as Σ . In (1), A , B , C , and D are maps between suitable spaces (or matrices of suitable dimensions) and the functions x , u , and z are considered to be defined on the real axis $\mathbb{R}$ or on any subinterval of it. In particular, one often assumes the domain of definition to be the nonnegative part of $\mathbb{R}$. The function u is called the *input*, and its values are assumed to be given from outside the system. The class of admissible input functions will be denoted $\mathbf{U}$. Often, $\mathbf{U}$ will be the class of piecewise continuous or locally integrable functions, but for most purposes, the exact class from which the input functions are chosen is not important. We will assume that input functions take values in an m -dimensional space $\mathcal{U}$, which we often identify with $\mathbb{R}^m$. The variable x is called the *state variable* and it is assumed to take values in an n -dimensional space $\mathcal{X}$. The space $\mathcal{X}$ will be called

the *state space*. It will usually be identified with $\mathbb{R}^n$. Finally, z is called the *to be controlled output* of the system and takes values in a p -dimensional space $\mathcal{Z}$, which we identify with $\mathbb{R}^p$. The solution of the differential equation of Σ with initial value $x(0) = x_0$ will be denoted as $x_u(t, x_0)$. It can be given explicitly using the variation-of-constants formula (see Trentelman et al. 2001, p. 38). The set of eigenvalues of a given matrix M is called the *spectrum* of M and is denoted by $\sigma(M)$. The system (1) is called *stabilizable* if there exists a map (matrix of suitable dimensions) F such that $\sigma(A + BF) \subset \mathbb{C}^-$. Here, $\mathbb{C}^-$ denotes the open left half complex plane, i.e., $\{\lambda \in \mathbb{C} \mid \operatorname{Re}(\lambda) < 0\}$. We often express this property by saying that the pair (A, B) is stabilizable.

The Linear Quadratic Optimal Control Problem

Assume that our aim is to keep all components of the output $z(t)$ as small as possible, for all $t \geq 0$. In the ideal situation, with initial state $x(0) = 0$, the uncontrolled system (with control input $u = 0$) evolves along the stationary solution $x(t) = 0$. Of course, the output $z(t)$ will then also be equal to zero for all t . If, however, at time $t = 0$ the state of the system is perturbed to, say, $x(0) = x_0$, with $x_0 \neq 0$, then the uncontrolled system will evolve along a state trajectory unequal to the stationary zero solution, and we will get $z(t) = Ce^{At}x_0$. To remedy this, from time $t = 0$ on, we can apply an input function u , so that for $t \geq 0$ the corresponding output becomes equal to $z(t) = Cx_u(t, x_0) + Du(t)$. Keeping in mind that we want the output $z(t)$ to be as small as possible for all $t \geq 0$, we can measure its size by the quadratic cost functional

$$J(x_0, u) = \int_0^\infty \|z(t)\|^2 dt, \quad (2)$$

where $\|\cdot\|$ denotes the Euclidean norm. Our aim to keep the values of the output as small as possible can then be expressed as requiring this integral to be as small as possible by suitable

choice of input function u . In this way we arrive at the *linear quadratic optimal control problem*:

Problem 1 Consider the system $\Sigma : \dot{x}(t) = Ax(t) + Bu(t)$, $z(t) = Cx(t) + Du(t)$. Determine for every initial state x_0 an input $u \in \mathbf{U}$ (a space of functions $[0, \infty) \rightarrow \mathcal{U}$) such that

$$J(x_0, u) := \int_0^\infty \|z(t)\|^2 dt \quad (3)$$

is minimal. Here $z(t)$ denotes the output trajectory $z_u(t, x_0)$ of Σ corresponding to the initial state x_0 and input function u .

Since the system is linear and the integrand in the cost functional is a quadratic function of z , the problem is called *linear quadratic*. Of course, $\|z\|^2 = x^\top C^\top Cx + 2u^\top D^\top Cx + u^\top D^\top Du$, so the integrand can also be considered as a quadratic function of (x, u) . The convergence of the integral in (3) is of course a point of concern. Therefore, one often considers the corresponding *finite-horizon problem* in a preliminary investigation. In this problem, a final time T is given and one wants to minimize the integral

$$J(x_0, u, T) := \int_0^T \|z(t)\|^2 dt. \quad (4)$$

In contrast to this, the first problem above is sometimes called the *infinite horizon problem*. An important issue is also the *convergence of the state*. Obviously, convergence of the integral does not always imply the convergence to zero of the state. Therefore, distinction is made between the problem with zero and with free endpoint. Problem 1 as stated is referred to as the *problem with free endpoint*. If one restricts the inputs u in the problem to those for which the resulting state trajectory tends to zero, one speaks about the *problem with zero endpoint*. Specifically:

Problem 2 In the situation of Problem 1, determine for every initial state x_0 an input $u \in \mathbf{U}$ such that $x_u(t, x_0) \rightarrow 0$ ($t \rightarrow \infty$) and such that under this condition, $J(x_0, u)$ is minimized.

In the literature various special cases of these problems have been considered, and names have

been associated to these special cases. In particular, Problems 1 and 2 are called *regular* if the matrix D is injective, equivalently, $D^\top D > 0$. If, in addition, $C^\top D = 0$ and $D^\top D = I$, then the problems are said to be in *standard form*. In the standard case, the integrand in the cost functional reduces to $\|z\|^2 = x^\top C^\top Cx + u^\top u$. We often write $Q = C^\top C$. The standard case is a special case, which is not essentially simpler than the general regular problem, but which gives rise to simpler formulas. The general regular problem can be reduced to the standard case by means of a suitable feedback transformation.

The Finite-Horizon Problem

The finite-horizon problem in standard form is formulated as follows:

Problem 3 Given the system $\dot{x}(t) = Ax(t) + Bu(t)$, a final time $T > 0$, and symmetric matrices N and Q such that $N \geq 0$ and $Q \geq 0$, determine for every initial state x_0 a piecewise continuous input function $u : [0, T] \rightarrow \mathcal{U}$ such that the integral

$$J(x_0, u, T) := \int_0^T x(t)^\top Qx(t) + u(t)^\top u(t) dt + x(T)^\top Nx(T) \quad (5)$$

is minimized.

In this problem, we have introduced a *weight on the final state*, using the matrix N . This generalization of the problem does not give rise to additional complications.

A key ingredient in solving this finite-horizon problem is the *Riccati differential equation* associated with the problem:

$$\begin{aligned} \dot{P}(t) &= A^\top P(t) + P(t)A - P(t)BB^\top P(t) + Q, \\ P(0) &= N. \end{aligned} \quad (6)$$

This is a quadratic differential equation on the interval $[0, \infty]$ in terms of the matrices A , B , and Q , and with initial condition given by the weight matrix N on the final state. The unknown in the differential equation is the matrix valued

function $P(t)$. The following theorem solves the finite-horizon problem. It states that the Riccati differential equation with initial condition (6) has a unique solution on $[0, \infty)$, that the optimal value of the cost functional is determined by the value of this solution at time T , and that there exists a unique optimal input that is generated by a time-varying state feedback control law:

Theorem 1 *Consider Problem 3. The following properties hold:*

1. *The Riccati differential equation with initial value (6) has a unique solution $P(t)$ on $[0, \infty)$. This solution is symmetric and positive semidefinite for all $t \geq 0$.*
2. *For each x_0 there is exactly one optimal input function, i.e., a piecewise continuous function u^* on $[0, T]$ such that $J(x_0, u^*, T) = J^*(x_0, T) := \inf\{J(x_0, u, T) \mid u \in \mathbf{U}\}$. This optimal input function u^* is generated by the time-varying feedback control law*

$$u(t) = -B^\top P(T-t)x(t) \quad (0 \leq t \leq T). \quad (7)$$

3. *For each x_0 , the minimal value of the cost functional equals*

$$J^*(x_0, T) = x_0^\top P(T)x_0.$$

4. *If $N = 0$, then the function $t \mapsto P(t)$ is an increasing function in the sense that $P(t) - P(s)$ is positive semidefinite for $t \geq s$.*

The Infinite-Horizon Problem with Free Endpoint

We consider the situation as described in Theorem 1 with $N = 0$. An obvious conjecture is that $x_0^\top P(T)x_0$ converges to the minimal cost of the infinite-horizon problem as $T \rightarrow \infty$. The convergence of $x_0^\top P(T)x_0$ for all x_0 is equivalent to the convergence of the matrix $P(T)$ for $T \rightarrow \infty$ to some matrix P^- . Such a convergence does not always take place. In order to achieve convergence, we make the following assumption: for every x_0 , there exists an input u for which the integral

$$J(x_0, u) := \int_0^\infty x(t)^\top Qx(t) + u(t)^\top u(t) dt \quad (8)$$

converges, i.e., for which the cost $J(x_0, u)$ is finite. Obviously, for the problem to make sense for all x_0 , this condition is necessary. It is easily seen that the stabilizability of (A, B) is a sufficient condition for the above assumption to hold (not necessary, take, e.g., $Q = 0$). Take an arbitrary initial state x_0 and assume that $\bar{u}$ is a function such that the integral (8) is finite. We have for every $T > 0$ that

$$x_0^\top P(T)x_0 \leq J(x_0, \bar{u}, T) \leq J(x_0, \bar{u}),$$

which implies that for every x_0 , the expression $x_0^\top P(T)x_0$ is bounded. This implies that $P(T)$ is bounded. Since $P(T)$ is increasing with respect to T , it follows that $P^- := \lim_{T \rightarrow \infty} P(T)$ exists. Since P satisfies the differential equation (6), it follows that also $\dot{P}(t)$ has a limit as $t \rightarrow \infty$. It is easily seen that this latter limit must be zero. Hence, $P = P^-$ satisfies the following equation:

$$A^\top P + PA - PBB^\top P + Q = 0. \quad (9)$$

This is called the *algebraic Riccati equation* (ARE). The solutions of this equation are exactly the constant solutions of the Riccati differential equation. The previous consideration shows that the ARE has a positive semidefinite solution P^- . The solution is not necessarily unique, not even with the extra condition that $P \geq 0$. However, P^- turns out to be the *smallest* real symmetric positive semidefinite solution of the ARE.

The following theorem now establishes a complete solution to the regular standard form version of Problem 1:

Theorem 2 *Consider the system $\dot{x}(t) = Ax(t) + Bu(t)$ together with the cost functional*

$$J(x_0, u) := \int_0^\infty x(t)^\top Qx(t) + u(t)^\top u(t) dt,$$

with $Q \geq 0$. Factorize $Q = C^\top C$. Then, the following statements are equivalent:

1. For every $x_0 \in X$, there exists $u \in \mathbf{U}$ such that $J(x_0, u) < \infty$.
2. The ARE (9) has a real symmetric positive semidefinite solution P .

Assume that one of the above conditions holds. Then, there exists a smallest real symmetric positive semidefinite solution of the ARE, i.e., there exists a real symmetric solution $P^- \geq 0$ such that for every real symmetric solution $P \geq 0$, we have $P^- \leq P$. For every x_0 , we have

$$J^*(x_0) := \inf\{J(x_0, u) \mid u \in \mathbf{U}\} = x_0^\top P^- x_0.$$

Furthermore, for every x_0 , there is exactly one optimal input function, i.e., a function $u^* \in \mathbf{U}$ such that $J(x_0, u^*) = J^*(x_0)$. This optimal input is generated by the time-invariant feedback law

$$u(t) = -B^\top P^- x(t).$$

The Infinite-Horizon Problem with Zero Endpoint

In addition to the free endpoint problem, we consider the version of the linear quadratic problem with zero endpoint. In this case the aim is to minimize for every x_0 the cost functional over all inputs u such that $x_u(t, x_0) \rightarrow 0$ ($t \rightarrow \infty$). For each x_0 such u exists if and only if the pair (A, B) is stabilizable. A solution to the regular standard form version of Problem 2 is stated next:

Theorem 3 Consider the system $\dot{x}(t) = Ax(t) + Bu(t)$ together with the cost functional

$$J(x_0, u) := \int_0^\infty x(t)^\top Q x(t) + u(t)^\top u(t) dt,$$

with $Q \geq 0$. Assume that (A, B) is stabilizable. Then:

1. There exists a largest real symmetric solution of the ARE, i.e., there exists a real symmetric solution P^+ such that for every real symmetric solution P , we have $P \leq P^+$. P^+ is positive semidefinite.

2. For every initial state x_0 , we have

$$J_0^*(x_0) = x_0^\top P^+ x_0.$$

3. For every initial state x_0 , there exists an optimal input function, i.e., a function $u^* \in \mathbf{U}$ with $x(\infty) = 0$ such that $J(x_0, u^*) = J_0^*(x_0)$ if and only if every eigenvalue of A on the imaginary axis is (Q, A) observable, i.e., $\text{rank} \begin{pmatrix} A - \lambda I \\ Q \end{pmatrix} = n$ for all $\lambda \in \sigma(A)$ with $\text{Re}(\lambda) = 0$.

Under this assumption we have:

4. For every initial state x_0 , there is exactly one optimal input function u^* . This optimal input function is generated by the time-invariant feedback law

$$u(t) = -B^\top P^+ x(t).$$

5. The optimal closed-loop system $\dot{x}(t) = (A - BB^\top P^+)x(t)$ is stable. In fact, P^+ is the unique real symmetric solution of the ARE for which $\sigma(A - BB^\top P^+) \subset \mathbb{C}^-$.

Summary and Future Directions

Linear quadratic optimal control deals with finding an input function that minimizes a quadratic cost functional for a given linear system. The cost functional is the integral of a quadratic form in the input and state variable of the system. If the integral is taken over a finite time interval the problem is called a finite-horizon problem, and the optimal cost and optimal state feedback gain can be expressed in terms of the solution of an associated Riccati differential equation. If we integrate over an infinite time interval, the problem is called an infinite-horizon problem. The optimal cost and optimal feedback gain for the free endpoint problem can be found in terms of the smallest nonnegative real symmetric solution of the associated algebraic Riccati equation. For the zero endpoint problem, these are given in

terms of the largest real symmetric solution of the algebraic Riccati equation.

Cross-References

- [Finite-Horizon Linear-Quadratic Optimal Control with General Boundary Conditions](#)
- [H-Infinity Control](#)
- [H₂ Optimal Control](#)
- [Linear State Feedback](#)

Recommended Reading

The linear quadratic regulator problem and the Riccati equation were introduced by R.E. Kalman in the early 1960s (see Kalman (1960)). Extensive treatments of the problem can be found in the textbooks Brockett (1969), Kwakernaak and Sivan (1972), and Anderson and Moore (1971). For a detailed study of the Riccati differential equation and the algebraic Riccati equation, we refer to Wonham (1968). Extensions of the linear quadratic regulator problem to linear quadratic optimization problems, where the integrand of the cost functional is a possibly indefinite quadratic function of the state and input variable, were studied in the classical paper of Willems (1971). A further reference for the geometric classification of all real symmetric solutions of the algebraic Riccati equation is Coppel (1974). For the question what level of system performance can be obtained if, in the cost functional, the weighting matrix of the control input is singular or nearly singular leading to singular and nearly singular linear quadratic optimal control problems and “cheap control” problems, we refer to Kwakernaak and Sivan (1972). An early reference for a discussion on the singular problem is the work of Clements and Anderson (1978). More details can be found in Willems (1971) and Schumacher (1983). In singular problems, in general one allows for distributions as inputs. This approach was worked out in detail in Hautus and Silverman (1983) and Willems et al. (1986). For a more

recent reference, including an extensive list of references, we refer to the textbook of Trentelman et al. (2001).

Bibliography

- Anderson BDO, Moore JB (1971) Linear optimal control. Prentice Hall, Englewood Cliffs
- Brockett RW (1969) Finite dimensional linear systems. Wiley, New York
- Clements DJ, Anderson BDO (1978) Singular optimal control: the linear quadratic problem. Volume 5 of lecture notes in control and information sciences. Springer, New York
- Coppel WA (1974) Matrix quadratic equations. Bull Aust Math Soc 10:377–401
- Hautus MLJ, Silverman LM (1983) System structure and singular control. Linear Algebra Appl 50:369–402
- Kalman RE (1960) Contributions to the theory of optimal control. Bol Soc Mat Mex 5:102–119
- Kwakernaak H, Sivan R (1972) Linear optimal control theory. Wiley, New York
- Schumacher JM (1983) The role of the dissipation matrix in singular optimal control. Syst Control Lett 2:262–266
- Trentelman HL, Hautus MLJ, Stoorvogel AA (2001) Control theory for linear systems. Springer, London
- Willems JC (1971) Least squares stationary optimal control and the algebraic Riccati equation. IEEE Trans Autom Control 16:621–634
- Willems JC, Kitapçı A, Silverman LM (1986) Singular optimal control: a geometric approach. SIAM J Control Optim 24:323–337
- Wonham WM (1968) On a matrix Riccati equation of stochastic control. SIAM J Control Optim 6(4): 681–697

Linear Quadratic Zero-Sum Two-Person Differential Games

Pierre Bernhard

INRIA-Sophia Antipolis-Méditerranée, Sophia Antipolis, France

Abstract

As in optimal control theory, linear quadratic (LQ) differential games (DG) can be solved, even in high dimension, via a Riccati equation.

However, contrary to the control case, existence of the solution of the Riccati equation is not necessary for the existence of a closed-loop saddle point. One may “survive” a particular, nongeneric, type of conjugate point. An important application of LQDGs is the so-called H_∞ -optimal control, appearing in the theory of robust control.

Keywords

Differential games · Finite horizon ·
H-infinity control · Infinite horizon

Perfect State Measurement

Linear quadratic differential games are a special case of differential games (DG). See the article ► [Pursuit-Evasion Games and Zero-Sum Two-Person Differential Games](#). They were first investigated by Ho et al. (1965), in the context of a linearized pursuit-evasion game. This subsection is based upon Bernhard (1979, 1980). A linear quadratic DG is defined as

$$\dot{x} = Ax + Bu + Dv, \quad x(t_0) = x_0,$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $v \in \mathbb{R}^\ell$, $u(\cdot) \in L^2([0, T], \mathbb{R}^m)$, $v(\cdot) \in L^2([0, T], \mathbb{R}^\ell)$. Final time T is given, there is no terminal constraint, and using the notation $x^T K x = \|x\|_K^2$,

$$J(t_0, x_0; u(\cdot), v(\cdot)) = \|x(T)\|_K^2 + \int_{t_0}^T (x^t u^t v^t)$$

$$\begin{pmatrix} Q & S_1 & S_2 \\ S_1^t & R & 0 \\ S_2^t & 0 & -\Gamma \end{pmatrix} \begin{pmatrix} x \\ u \\ v \end{pmatrix} dt.$$

The matrices of appropriate dimensions, A , B , D , Q , S_i , R , and Γ , may all be measurable functions of time. R and Γ must be positive definite with inverses bounded away from zero. To get the most complete results available, we assume also that K and Q are nonnegative definite, although this is only necessary for some of the following results. Detailed results without that assumption were

obtained by Zhang (2005) and Delfour (2005). We chose to set the cross term in uv in the criterion null; this is to simplify the results and is not necessary. This problem satisfies Isaacs' condition (see article DG) even with nonzero such cross terms.

Using the change of control variables

$$u = \tilde{u} - R^{-1} S_1^t x, \quad v = \tilde{v} + \Gamma^{-1} S_2^t x,$$

yields a DG with the same structure, with modified matrices A and Q , but without the cross terms in xu and xv . (This extends to the case with nonzero cross terms in uv .) Thus, without loss of generality, we will proceed with $(S_1 \ S_2) = (0 \ 0)$.

The existence of open-loop and closed-loop solutions to that game is ruled by two Riccati equations for symmetric matrices P and P^* , respectively, and by a pair of canonical equations that we shall see later:

$$\begin{aligned} \dot{P} + PA + A^t P - PBR^{-1}B^t P + PD\Gamma^{-1}D^t P \\ + Q = 0, \quad P(T) = K, \end{aligned} \quad (1)$$

$$\begin{aligned} \dot{P}^* + P^*A + A^t P^* + P^*D\Gamma^{-1}D^t P^* + Q = 0, \\ P^*(T) = K. \end{aligned} \quad (2)$$

When both Riccati equations have a solution over $[t, T]$, it holds that in the partial ordering of definiteness,

$$0 \leq P(t) \leq P^*(t).$$

When the saddle point exists, it is represented by the state feedback strategies

$$u = \varphi^*(t, x) = -R^{-1}B^t P(t)x,$$

$$v = \psi^*(t, x) = \Gamma^{-1}D^t P(t)x. \quad (3)$$

The control functions generated by this pair of feedbacks will be noted $\hat{u}(\cdot)$ and $\hat{v}(\cdot)$.

Theorem 1

- A sufficient condition for the existence of a closed-loop saddle point, then given by

- (φ^*, ψ^*) in (3), is that Eq. (1) has a solution $P(t)$ defined over $[t_0, T]$.
- A necessary and sufficient condition for the existence of an open-loop saddle point is that Eq. (2) has a solution over $[t_0, T]$ (and then so does (1)). In that case, the pairs $(\hat{u}(\cdot), \hat{v}(\cdot))$, $(\hat{u}(\cdot), \psi^*)$, and (φ^*, ψ^*) are saddle points.
 - A necessary and sufficient condition for $(\varphi^*, \hat{v}(\cdot))$ to be a saddle point is that Eq. (1) has a solution over $[t_0, T]$.
 - In all cases where a saddle point exists, the Value function is $V(t, x) = \|x\|_{P(t)}^2$.

However, Eq. (1) may fail to have a solution and a closed-loop saddle point still exists. The precise necessary condition is as follows: let $X(\cdot)$ and $Y(\cdot)$ be two square matrix function solutions of the canonical equations

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = \begin{pmatrix} A & -BR^{-1}B^t + D\Gamma^{-1}D^t \\ -Q & -A^t \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix},$$

$$\begin{pmatrix} X(T) \\ Y(T) \end{pmatrix} = \begin{pmatrix} I \\ K \end{pmatrix}.$$

The matrix $P(t)$ exists for $t \in [t_0, T]$ if and only if $X(t)$ is invertible over that range, and then, $P(t) = Y(t)X^{-1}(t)$. Assume that the rank of $X(t)$ is piecewise constant, and let $X^\dagger(t)$ denote the pseudo-inverse of $X(t)$ and $\mathcal{R}(X(t))$ its range.

Theorem 2 *A necessary and sufficient condition for a closed-loop saddle point to exist, which is then given by (3) with $P(t) = Y(t)X^\dagger(t)$, is that*

1. $x_0 \in \mathcal{R}(X(t_0))$.
2. For almost all $t \in [t_0, T]$, $\mathcal{R}(D(t)) \subset \mathcal{R}(X(t))$.
3. $\forall t \in [t_0, T]$, $Y(t)X^\dagger(t) \geq 0$.

In a case where $X(t)$ is only singular at an isolated instant t^* (then conditions 1 and 2 above are automatically satisfied), called a *conjugate point* but where YX^{-1} remains positive definite on both sides of it, the conjugate point is called *even*. The feedback gain $F = -R^{-1}B^tP$

diverges upon reaching t^* , but on a trajectory generated by this feedback, the control $u(t) = F(t)x(t)$ remains finite. (See an example in Bernhard 1979.)

If $T = \infty$, with all system and payoff matrices constant and $Q > 0$, Mageirou (1976) has shown that if the algebraic Riccati equation obtained by setting $\dot{P} = 0$ in (1) admits a positive definite solution P , the game has a Value $\|x\|_P^2$, but (3) may not be a saddle point. (ψ^* may not be an equilibrium strategy.)

H_∞ -Optimal Control

This subsection is entirely based upon Başar and Bernhard (1995). It deals with imperfect state measurement, using Bernhard's nonlinear *minimax certainty equivalence principle* (Bernhard and Rapaport 1996).

Several problems of robust control may be brought to the following one: a linear, time-invariant system with two inputs (control input $u \in \mathbb{R}^m$ and disturbance input $w \in \mathbb{R}^\ell$) and two outputs (measured output $y \in \mathbb{R}^p$ and controlled output $z \in \mathbb{R}^q$) is given. One wishes to control the system with a nonanticipative controller $u(\cdot) = \phi(y(\cdot))$ in order to minimize the induced linear operator norm between spaces of square-integrable functions, of the resulting operator $w(\cdot) \mapsto z(\cdot)$.

It turns out that the problem which has a tractable solution is a kind of dual one: given a positive number γ , is it possible to make this norm no larger than γ ? The answer to this question is yes if and only if

$$\inf_{\phi \in \Phi} \sup_{w(\cdot) \in L^2} \int_{-\infty}^{\infty} (\|z(t)\|^2 - \gamma^2 \|w(t)\|^2) dt \leq 0.$$

We shall extend somewhat this classical problem by allowing either a time variable system, with a finite horizon T , or a time-invariant system with an infinite horizon.

The dynamical system is

$$\dot{x} = Ax + Bu + Dw, \quad (4)$$

$$y = Cx + Ew, \quad (5)$$

$$z = Hx + Gu. \quad (6)$$

We let

$$\begin{pmatrix} D \\ E \end{pmatrix} (D^t \ E^t) = \begin{pmatrix} M \ L \\ L^t \ N \end{pmatrix},$$

$$\begin{pmatrix} H^t \\ G^t \end{pmatrix} (H \ G) = \begin{pmatrix} Q \ S \\ S^t \ R \end{pmatrix},$$

and we assume that E is onto, $\Leftrightarrow N > 0$, and G is one-to-one $\Leftrightarrow R > 0$.

Finite Horizon

In this part, we consider a time-varying system, with all matrix functions measurable. Since the state is not known exactly, we assume that the initial state is not known either. The issue is therefore to decide whether the criterion

$$J_\gamma = \|x(T)\|_K^2 + \int_{t_0}^T (\|z(t)\|^2 - \gamma^2 \|w(t)\|^2) dt - \gamma^2 \|x_0\|_{\Sigma_0}^2 \quad (7)$$

may be kept finite and with which strategy. Let

$$\gamma^* = \inf\{\gamma \mid \inf_{\phi \in \Phi} \sup_{x_0 \in \mathbb{R}^n, w(\cdot) \in L^2} J_\gamma < \infty\}.$$

Theorem 3 $\gamma \leq \gamma^*$ if and only if the following three conditions are satisfied:

1. The following Riccati equation has a solution over $[t_0, T]$:

$$\begin{aligned} -\dot{P} = & PA + A^t P - (PB + S)R^{-1}(B^t P + S^t) \\ & + \gamma^{-2} PMP + Q, \quad P(T) = K. \end{aligned} \quad (8)$$

2. The following Riccati equation has a solution over $[t_0, T]$:

$$\begin{aligned} \dot{\Sigma} = & A\Sigma + \Sigma A^t - (\Sigma C^t + L)N^{-1}(C\Sigma + L^t) \\ & + \gamma^{-2} \Sigma Q \Sigma + M, \quad \Sigma(t_0) = \Sigma_0. \end{aligned} \quad (9)$$

3. The following spectral radius condition is satisfied:

$$\forall t \in [t_0, T], \quad \rho(\Sigma(t)P(t)) < \gamma^2. \quad (10)$$

In that case, the optimal controller ensuring $\inf_{\phi} \sup_{x_0, w} J_\gamma$ is given by a “worst case state” $\hat{x}(\cdot)$ satisfying $\hat{x}(0) = 0$ and

$$\begin{aligned} \dot{\hat{x}} = & [A - BR^{-1}(B^t P + S^t) + \gamma^{-2} D(D^t P + L^t)] \\ & \hat{x} + (I - \gamma^{-2} \Sigma P)^{-1} (\Sigma C^t + L)(y - C\hat{x}), \end{aligned} \quad (11)$$

and the certainty equivalent controller

$$\phi^*(y(\cdot))(t) = -R^{-1}(B^t P + S^t)\hat{x}(t). \quad (12)$$

Infinite Horizon

The infinite horizon case is the traditional H_∞ -optimal control problem reformulated in a state space setting. We let all matrices defining the system be constant. We take the integral in (7) from $-\infty$ to $+\infty$, with no initial or terminal term of course. We add the hypothesis that the pairs (A, B) and (A, D) are stabilizable and the pairs (C, A) and (H, A) detectable. Then, the theorem is as follows:

Theorem 4 $\gamma \leq \gamma^*$ if and only if the following conditions are satisfied: The algebraic Riccati equations obtained by placing $\dot{P} = 0$ and $\dot{\Sigma} = 0$ in (8) and (9) have positive definite solutions, which satisfy the spectral radius condition (10). The optimal controller is given by Eqs. (11) and (12), where P and Σ are the minimal positive definite solutions of the algebraic Riccati equations, which can be obtained as the limit of the solutions of the differential equations as $t \rightarrow -\infty$ for P and $t \rightarrow \infty$ for Σ .

Conclusion

The similarity of the H_∞ -optimal control theory with the LQG, stochastic, theory is in many respects striking, as is the duality observation control. Yet, the “observer” of H_∞ -optimal control does *not* arise from some estimation theory but from the analysis of a “worst case.” The best explanation might be in the duality of

the ordinary, or $(+, \times)$, algebra with the idempotent $(\max, +)$ algebra (see Bernhard 1996). The complete theory of H_∞ -optimal control in that perspective has yet to be written.

Mageirou EF (1976) Values and strategies for infinite time linear quadratic games. *IEEE Trans Autom Control* AC-21:547–550

Zhang P (2005) Some results on zero-sum two-person linear quadratic differential games. *SIAM J Control Optim* 43:2147–2165

Cross-References

- [Dynamic Noncooperative Games](#)
- [Finite-Horizon Linear-Quadratic Optimal Control with General Boundary Conditions](#)
- [Game Theory: A General Introduction and a Historical Overview](#)
- [H-Infinity Control](#)
- [H₂ Optimal Control](#)
- [Linear Quadratic Optimal Control](#)
- [Optimization-Based Robust Control](#)
- [Pursuit-Evasion Games and Zero-Sum Two-Person Differential Games](#)
- [Robust Control of Infinite-Dimensional Systems](#)
- [Robust \$\mathcal{H}_2\$ Performance in Feedback Control](#)
- [Sampled-Data H-Infinity Optimization](#)
- [Stochastic Dynamic Programming](#)
- [Stochastic Linear-Quadratic Control](#)

Bibliography

- Başar T, Bernhard P (1995) H_∞ -optimal control and related minimax design problems: a differential games approach, 2nd edn. Birkhäuser, Boston
- Bernhard P (1979) Linear quadratic zero-sum two-person differential games: necessary and sufficient condition. *JOTA* 27(1):51–69
- Bernhard P (1980) Linear quadratic zero-sum two-person differential games: necessary and sufficient condition: comment. *JOTA* 31(2):283–284
- Bernhard P (1996) A separation theorem for expected value and feared value discrete time control. *COCV* 1:191–206
- Bernhard P, Rapaport A (1996) Min-max certainty equivalence principle and differential games. *Int J Robust Nonlinear Control* 6:825–842
- Delfour M (2005) Linear quadratic differential games: saddle point and Riccati equation. *SIAM J Control Optim* 46:750–774
- Ho Y-C, Bryson AE, Baron S (1965) Differential games and optimal pursuit-evasion strategies. *IEEE Trans Autom Control* AC-10:385–389

Linear State Feedback

Panos J. Antsaklis¹ and Alessandro Astolfi^{2,3}

¹Department of Electrical Engineering,
University of Notre Dame, Notre Dame,
IN, USA

²Department of Electrical and Electronic
Engineering, Imperial College London,
London, UK

³Dipartimento di Ingegneria Civile e Ingegneria
Informatica, Università di Roma Tor Vergata,
Rome, Italy

Abstract

Feedback is a fundamental mechanism in nature and central in the control of systems. The state of a system contains important information about the system; hence feeding back the state is a powerful control policy. To illustrate the effect of feedback in linear systems, continuous- and discrete-time state-variable descriptions are used: these allow the resulting closed-loop descriptions to be explicitly written and the effect of feedback on the eigenvalues of the closed-loop system to be studied. The eigenvalue assignment problem is also discussed.

Keywords

Linear systems · Feedback · Stabilization

Introduction

Feedback is a fundamental mechanism arising in nature. Feedback is also common in engineered systems and is essential in the automatic control of dynamic processes with uncertainties in their

model descriptions and in their interactions with the environment. When feedback is used, the values of the system variables are sensed, fed back, and used to control the system. That is, a control law decision process is based not only on predictions about the system behavior derived from a process model (as in open-loop control), but also on information about the actual behavior (closed-loop feedback control). If not all state variables are sensed, one could resort to a control architecture which relies on available measurements, such as the output of the system, and on an estimation mechanism, which generates an asymptotically converging estimate of the state to be used for feedback.

Linear Continuous-Time Systems

Consider, to begin with, time-invariant systems described by the state-variable description

$$\dot{x} = Ax + Bu, \quad y = Cx + Du, \quad (1)$$

in which $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the input, $y(t) \in \mathbb{R}^p$ is the output, and $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, and $D \in \mathbb{R}^{p \times m}$ are constant matrices. In this case the linear state feedback (lsf) control law is selected as

$$u(t) = Fx(t) + r(t), \quad (2)$$

where $F \in \mathbb{R}^{m \times n}$ is the constant lsf gain matrix and $r(t) \in \mathbb{R}^m$ is a new external input.

Substituting (2) into (1) yields the closed-loop state variable description, namely

$$\begin{aligned} \dot{x} &= (A + BF)x + Br, \\ y &= (C + DF)x + Dr. \end{aligned} \quad (3)$$

Appropriately selecting F , primarily to modify $A + BF$, one affects and improves the behavior of the system.

A number of comments are in order.

- Feeding back the information from the state x of the system is expected to be, and it is,

an effective way to alter the system behavior. This is because knowledge of the (initial) state and the input, in the absence of measurement noise and disturbances, uniquely determines the system's future behavior and, intuitively, using the state information should be a good way to control the system.

- In a state feedback control law the input u can be any function of the state, that is, $u = f(x, r)$, not necessarily linear with constant gain F as in (2). Typically, given (1), (2) is selected as the lsf primarily because the resulting closed-loop description (3) is also a linear time-invariant system. However, depending on the application needs, the state feedback control law can be more complex than (2).
- Although the equations (3) that describe the closed-loop behavior are different from equations (1), this does not imply that the system parameters have changed. The way feedback control acts is not by changing the system parameters A , B , C , and D , but by changing u so that the closed-loop system behaves as if the parameters were changed. When one applies say a step via $r(t)$ in the closed-loop system, then u in (2) is modified appropriately so the system behaves in a desired way.
- It is possible to implement u in (2) as an open-loop control law, namely

$$\begin{aligned} \hat{u} &= F[sI - (A + BF)]^{-1}x(0) \\ &\quad + [I - F(sI - A)^{-1}B]^{-1}\hat{r}, \end{aligned} \quad (4)$$

where the Laplace transforms $\hat{u} = \hat{u}(s)$ and $\hat{r} = \hat{r}(s)$ have been used for notational convenience. Equation (4) produces exactly the same input as equation (2), but it has the serious disadvantage that it is based exclusively on prior knowledge on the system (notably $x(0)$). As a result when there are uncertainties (and there always are), the open-loop control law (4) may fail, while the closed-loop control law (2) succeeds.

Analogous definitions exist for continuous-time, time-varying, systems described by the equations

$$\dot{x} = A(t)x + B(t)u, \quad y = C(t)x + D(t)u. \quad (5)$$

In this framework the lsf control law is given by

$$u = F(t)x + r, \quad (6)$$

and the resulting closed-loop system is

$$\begin{aligned} \dot{x} &= [A(t) + B(t)F(t)]x + B(t)r, \\ y &= [C(t) + D(t)F(t)]x + D(t)r. \end{aligned} \quad (7)$$

Linear Discrete-Time Systems

For the discrete-time, time-invariant, case the system description is

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k) + Du(k), \end{aligned} \quad (8)$$

the lsf control law is defined as

$$u(k) = Fx(k) + r(k), \quad (9)$$

and the closed-loop system is described by

$$\begin{aligned} x(k+1) &= (A + BF)x(k) + Br(k), \\ y(k) &= (C + DF)x(k) + Dr(k). \end{aligned} \quad (10)$$

Similarly, for the discrete-time, time-varying, case

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k), \\ y(k) &= C(k)x(k) + D(k)u(k), \end{aligned} \quad (11)$$

the lsf control law is defined as

$$u(k) = F(k)x(k) + r(k), \quad (12)$$

and the resulting closed-loop system is

$$\begin{aligned} x(k+1) &= [A(k) + B(k)F(k)]x(k) \\ &\quad + B(k)r(k), \\ y(k) &= [C(k) + D(k)F(k)]x(k) \\ &\quad + D(k)r(k). \end{aligned} \quad (13)$$

Selecting the Gain F

We select F (or $F(t)$) so that the closed-loop system has certain desirable properties. Stability is of course of major importance, although several control requirements can be addressed using lsf including, for example, tracking and regulation, diagonal decoupling, and disturbance rejection. In what follows we shall focus on stability. Stability can be achieved under appropriate controllability assumptions. In the time-varying case, one way to determine such a stabilizing $F(t)$ (or $F(k)$) is to use results from the optimal Linear Quadratic Regulator (LQR) theory which yields the “best” $F(t)$ in some sense.

In the time-invariant case, stabilization is equivalent to the problem of assigning the n eigenvalues of $(A + BF)$ in the stable region of the complex plane. If $\lambda_i, i = 1, \dots, n$, are the eigenvalues of $A + BF$, then F should be chosen so that, for all $i = 1, \dots, n$, the real part of λ_i is negative, that is, $Re(\lambda_i) < 0$ in the continuous time case, and the magnitude of λ_i is smaller than one, that is, $|\lambda_i| < 1$, in the discrete time case. Eigenvalue assignment is therefore an important objective which is discussed hereafter.

Eigenvalue Assignment Problem

For continuous-time and discrete-time, time-invariant, systems the eigenvalue assignment problem can be posed as follows. Given matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, find $F \in \mathbb{R}^{m \times n}$ such that the eigenvalues of $A + BF$ are assigned to arbitrary, complex conjugate, locations. Note that the characteristic polynomial of $A + BF$, namely, $\det(sI - (A + BF))$, has real coefficients, which implies that any complex eigenvalue is one of a pair of complex conjugate eigenvalues.

For single-input systems, that is, for systems with $m = 1$, the eigenvalue assignment problem has a simple solution, as illustrated in the following statement.

Proposition 1 Consider system (1) or (8). Let $m = 1$. Assume that

$$\text{rank } R = n,$$

where

$$R = [B, AB, \dots, A^{n-1}B],$$

that is, the system is reachable. Let $p(s)$ be a monic polynomial of degree n . Then there is a (unique) lsf gain matrix F such that the characteristic polynomial of $A + BF$ is equal to $p(s)$. Such an lsf gain matrix F is given by

$$F = -[0 \dots 0 \ 1] R^{-1} p(A). \quad (14)$$

Proposition 1 provides a constructive way to assign the characteristic polynomial, hence the eigenvalues, of the matrix $A + BF$. Note that, for low-order systems, i.e., if $n = 2$ or $n = 3$, it may be convenient to compute directly the characteristic polynomial of $A + BF$ and then compute F using the principle of identity of polynomials, i.e., F should be such that the coefficients of the polynomials $\det(sI - (A + BF))$ and $p(s)$ coincide. Equation (14) is known as Ackermann's formula.

The result summarized in Proposition 1 can be extended to multi-input systems.

Proposition 2 Consider system (1) or (8). Suppose

$$\text{rank } R = n,$$

that is the system is reachable. Let $p(s)$ be a monic polynomial of degree n . Then there is a lsf gain matrix F such that the characteristic polynomial of $A + BF$ is equal to $p(s)$.

Note that in the case $m > 1$ the lsf gain matrix F assigning the characteristic polynomial of the matrix $A + BF$ is not unique. To compute one such a lsf gain matrix one may exploit the following fact.

Lemma 1 (Heymann) Consider system (1) or (8). Suppose

$$\text{rank } R = n,$$

that is, the system is reachable. Let b_i be a nonzero column of the matrix B . Then there is a matrix G such that the single-input system (for convenience we write the system in

continuous time, but the same conclusions hold in the discrete-time case)

$$\dot{x} = (A + BG)x + b_i v, \quad (15)$$

is reachable.

Exploiting Lemma 1 it is possible to design a matrix F such that the characteristic polynomial of $A + BF$ equals some monic polynomial $p(s)$ of degree n in two steps. First we compute a matrix G such that the system (15) is reachable, and then we use Ackermann's formula to compute a lsf gain matrix F such that the characteristic polynomial of

$$A + BG + b_i F$$

is $p(s)$.

Transfer Functions

If $H_F(s)$ is the transfer function matrix of the closed-loop system (3), it is of interest to find its relation to the open-loop transfer function $H(s)$ of (1). It can be shown that

$$\begin{aligned} H_F(s) &= H(s)[I - F(sI - A)^{-1}B]^{-1} \\ &= H(s)[F(sI - (A + BF))^{-1}B + I]. \end{aligned}$$

In the single-input, single-output, case it can be readily shown that the lsf control law (2) only changes the coefficients of the denominator polynomial in the transfer function (this result is also true in the multi-input, multi-output case). Therefore if any of the (stable) zeros of $H(s)$ needs to be changed the only way to accomplish this via lsf is by pole-zero cancellation (assigning closed-loop poles at the open-loop zero locations; in the multi-input, multi-output case, closed-loop eigenvalue directions also need to be assigned for cancellations to take place).

Observer-Based Dynamic Controllers

When the state x is not available for feedback, an asymptotic estimator (a Luenberger observer) is

typically used to estimate the state. The estimate $\hat{x}$ of the state, instead of the actual x , is then used in (2) to control the system, in what is known as the *certainty equivalence* architecture.

Summary and Future Directions

The notion of state feedback for linear systems, possibly time-varying, has been discussed. It has been shown that state feedback modifies the closed-loop behavior. The related problem of eigenvalue assignment has been discussed and its connection with the reachability properties of the system has been highlighted. The class of feedback laws considered is the simplest possible one: if additional constraints on the input signal, or on the closed-loop performance, are imposed, then one has to resort to nonlinear state feedback or to dynamic state feedback. This is the case, for example, if the input signal is bounded in amplitude and/or rate or if one has to solve the so-called noninteracting control problem with stability, that is, the problem of designing a feedback law yielding asymptotic stability and such that the closed-loop system is composed of m decoupled subsystems.

Cross-References

- ▶ [Controllability and Observability](#)
- ▶ [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)
- ▶ [Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions](#)
- ▶ [Observer-Based Control](#)
- ▶ [Observers in Linear Systems Theory](#)
- ▶ [Tracking and Regulation in Linear Systems](#)

Bibliography

- Antsaklis PJ, Michel AN (2006) Linear systems. Birkhäuser, Boston
- Brockett RW (1970) Finite dimensional linear systems. Wiley, London
- Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs

- Trentelman HL, Stoorvogel AA, Hautus MLJ (2001) Control theory for linear systems. Springer, London
- Wonham WM (1967) On pole assignment in multi-input controllable linear systems, vol AC-12, pp 660–665
- Wonham WM (1985) Linear multivariable control: a geometric approach, 3rd edn. Springer, New York
- Zadeh LA, Desoer CA (1963) Linear system theory. McGraw-Hill, New York

Linear Systems: Continuous-Time Impulse Response Descriptions

Panos J. Antsaklis

Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

An important input–output description of a linear continuous-time system is its impulse response, which is the response $h(t, \tau)$ to an impulse applied at time τ . In time-invariant systems that are also causal and at rest at time zero, the impulse response is $h(t, 0)$ and its Laplace transform is the transfer function of the system. Expressions for $h(t, \tau)$ when the system is described by state-variable equations are also derived.

Keywords

Continuous-time · Impulse response descriptions · Linear systems · Time-invariant · Time-varying · Transfer function descriptions

Introduction

Consider linear continuous-time dynamical systems, the input–output behavior of which can be described by an integral representation of the form

$$y(t) = \int_{-\infty}^{+\infty} H(t, \tau) u(\tau) d\tau \quad (1)$$

where $t, \tau \in \mathbb{R}$, the output is $y(t) \in \mathbb{R}^p$, the input is $u(t) \in \mathbb{R}^m$, and $H : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^{p \times m}$ is assumed to be integrable. For instance, any system in state-variable form

$$\begin{aligned}\dot{x} &= A(t)x + B(t)u \\ y &= C(t)x + D(t)u\end{aligned}\quad (2)$$

or

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\quad (3)$$

also has a representation of the form (1) as we shall see below.

Note that it is assumed that at $\tau = -\infty$, the system is at rest. $H(t, \tau)$ is the *impulse response matrix* of the system (1). To explain, consider first a single-input single-output system:

$$y(t) = \int_{-\infty}^{+\infty} h(t, \tau)u(\tau)d\tau, \quad (4)$$

and recall that if $\delta(\hat{t} - \tau)$ denotes an impulse (delta or Dirac) function applied at time $\tau = \hat{t}$, then for a function $f(t)$,

$$f(\hat{t}) = \int_{-\infty}^{+\infty} f(\tau)\delta(\hat{t} - \tau)d\tau. \quad (5)$$

If now in (4) $u(\tau) = \delta(\hat{t} - \tau)$, that is, an impulse input is applied at $\tau = \hat{t}$, then the output $y_I(t)$ is

$$y_I(t) = h(t, \hat{t}),$$

i.e., $h(t, \hat{t})$ is the output at time t when an impulse is applied at the input at time $\hat{t}$. So in (4), $h(t, \tau)$ is the response at time t to an impulse applied at time τ . Clearly if the *impulse response* $h(t, \tau)$ is known, the response to any input $u(t)$ can be derived via (4), and so $h(t, \tau)$ is an input/output description of the system.

Equation (1) is a generalization of (4) for the multi-input, multi-output case. If we let all the components of $u(\tau)$ in (1) be zero except the j th component, then

$$y_i(t) = \int_{-\infty}^{+\infty} h_{ij}(t, \tau)u_j(\tau)d\tau, \quad (6)$$

$h_{ij}(t, \tau)$ denotes the response of the i th component of the output of system (1) at time t due to an impulse applied to the j th component of the input at time τ with all remaining components of the input being zero. $H(t, \tau) = [h_{ij}(t, \tau)]$ is called the *impulse response matrix* of the system.

If it is known that system (1) is causal, then the output will be zero before an input is applied. Therefore,

$$H(t, \tau) = 0, \quad \text{for } t < \tau, \quad (7)$$

and (1) becomes

$$y(t) = \int_{-\infty}^t H(t, \tau)u(\tau)d\tau. \quad (8)$$

Rewrite (8) as

$$\begin{aligned}y(t) &= \int_{-\infty}^{t_0} H(t, \tau)u(\tau)d\tau + \int_{t_0}^t H(t, \tau)u(\tau)d\tau \\ &= y(t_0) + \int_{t_0}^t H(t, \tau)u(\tau)d\tau.\end{aligned}\quad (9)$$

If (1) is at rest at $t = t_0$ (i.e., if $u(t) = 0$ for $t \geq t_0$, then $y(t) = 0$ for $t \geq t_0$), $y(t_0) = 0$ and (9) becomes

$$y(t) = \int_{t_0}^t H(t, \tau)u(\tau)d\tau. \quad (10)$$

If in addition system (1) is time-invariant, then $H(t, \tau) = H(t - \tau, 0)$ (also written as $H(t - \tau)$) since only the elapsed time $(t - \tau)$ from the application of the impulse is important. Then (10) becomes

$$y(t) = \int_0^t H(t - \tau)u(\tau)d\tau, \quad t \geq 0, \quad (11)$$

where we chose $t_0 = 0$ without loss of generality. Equation (11) is the description for causal, time-invariant systems, at rest at $t = 0$.

Equation (11) is a convolution integral and if we take the (one-sided or unilateral) Laplace transform of both sides,

$$\hat{y}(s) = \hat{H}(s)\hat{u}(s), \quad (12)$$

where $\hat{y}(s)$, $\hat{u}(s)$ are the Laplace transforms of $y(t)$, $u(t)$ and $\hat{H}(s)$ is the Laplace transform of the impulse response $H(t)$. $\hat{H}(s)$ is the *transfer function matrix* of the system. Note that the transfer function of a linear, time-invariant system is typically defined as the rational matrix $\hat{H}(s)$ that satisfies (12) for any input and its corresponding output assuming zero initial conditions, which is of course consistent with the above analysis.

Connection to State-Variable Descriptions

When a system is described by the state-variable description (2), then

$$y(t) = \int_{t_0}^t [C(\tau)\Phi(t, \tau)B(\tau) + D(\tau)\delta(t - \tau)]u(\tau)d\tau, \quad (13)$$

where it was assumed that $x(t_0) = 0$, i.e., the system is at rest at t_0 . Here $\Phi(t, \tau)$ is the state transition matrix of the system defined by the Peano-Baker series:

$$\begin{aligned} \Phi(t, t_0) = I &+ \int_{t_0}^t A(\tau_1)d\tau_1 \\ &+ \int_{t_0}^t A(\tau_1) \left[\int_{t_0}^{\tau_1} A(\tau_2)d\tau_2 \right] d\tau_1 + \cdots; \end{aligned}$$

see ► [Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions](#).

Comparing (13) with (10), the impulse response

$$H(t, \tau) = \begin{cases} C(t)\Phi(t, \tau)B(t) + D(t)\delta(t - \tau) & t \geq \tau, \\ 0 & t < \tau. \end{cases} \quad (14)$$

Similarly, when the system is time-invariant and is described by (3),

$$y(t) = \int_{t_0}^t [Ce^{A(t-\tau)}B + D\delta(t - \tau)]u(\tau)d\tau, \quad (15)$$

where $x(t_0) = 0$.

Comparing (15) with (11), the impulse response

$$H(t - \tau) = \begin{cases} Ce^{A(t-\tau)}B + D\delta(t - \tau) & t \geq \tau, \\ 0 & t < \tau, \end{cases} \quad (16)$$

or as it is commonly written (taking the time when the impulse is applied to be zero, $\tau = 0$)

$$H(t) = \begin{cases} Ce^{At}B + D\delta(t) & t \geq 0, \\ 0 & t < 0. \end{cases} \quad (17)$$

Take now the (one-sided or unilateral) Laplace transform of both sides in (17) to obtain

$$\hat{H}(s) = C(sI - A)^{-1}B + D, \quad (18)$$

which is the transfer function matrix in terms of the coefficient matrices in the state-variable description (3). Note that (18) can also be derived directly from (3) by assuming zero initial conditions ($x(0) = 0$) and taking Laplace transform of both sides.

Finally, it is easy to show that equivalent state-variable descriptions give rise to the same impulse responses.

Summary

The continuous-time impulse response is an external, input–output description of linear, continuous-time systems. When the system is time-invariant, the Laplace transform of the impulse response $h(t, 0)$ (which is the output response at time t due to an impulse applied at time zero with initial conditions taken to be zero) is the transfer function of the system – another very common input–output description. The relationships with the state-variable descriptions are shown.

Cross-References

- ▶ [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)
- ▶ [Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions](#)

Recommended Reading

External or input–output descriptions such as the impulse response and the transfer function (in the time-invariant case) are described in several textbooks below.

Bibliography

- Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston
- DeCarlo RA (1989) Linear systems. Prentice-Hall, Englewood Cliffs
- Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs
- Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs
- Sontag ED (1990) Mathematical control theory: deterministic finite dimensional systems. Texts in applied mathematics, vol 6. Springer, New York

Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions

Panos J. Antsaklis
Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

Continuous-time processes that can be modeled by linear differential equations with constant coefficients can also be described in a systematic way in terms of state variable descriptions of the form $\dot{x}(t) = Ax(t) + Bu(t)$, $y(t) = Cx(t) + Du(t)$. The response of such systems due to a given input and a set

of initial conditions is derived and expressed in terms of the variation of constants formula. Equivalence of state variable descriptions is also discussed.

Keywords

Continuous-time · Linear systems · State variable descriptions · Time-invariant

Synonyms

[LTI Systems](#)

Introduction

Linear, continuous-time systems are of great interest because they model, exactly or approximately, the behavior over time of many practical physical systems of interest. We are particularly interested in systems, the behavior of which is described by linear, ordinary differential equations with constant coefficients.

Such descriptions can always be rewritten as a set of first-order differential equations, typically in the following convenient state variable form:

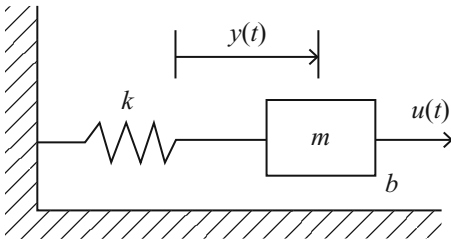
$$\begin{aligned}\dot{x} &= Ax(t) + Bu(t), & y(t) &= Cx(t) + Du(t); \\ x(0) &= x_0,\end{aligned}\tag{1}$$

where $x(t)$, the state vector, is a column vector of dimension n ($x(t) \in \mathbb{R}^n$) and $\dot{x}(t) = \frac{dx}{dt}$ with the derivative being taken element by element. $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$ are matrices with real entries (these are the constant coefficients that make the system time invariant); and $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$ are the inputs and outputs of the system. The vector differential equation is the *state equation* and the algebraic equation is the *output equation*.

The advantage of the above state variable description is that for given input $u(t)$ and initial condition $x(0)$, its solution (state and output motions or trajectories) can be conveniently and systematically characterized. This is shown below.

Deriving State Variable Descriptions

Description (1) may be derived directly, by modeling the behavior of a linear, continuous-time, time-invariant system, but more often it is derived either from the linearization of a nonlinear equation around an operating point or a trajectory or from higher-order differential equations that model the system's behavior. The example below illustrates the latter case.



Consider a spring-mass example, where a mass m slides horizontally on a surface with damping coefficient b due to friction and it is attached to a wall by a linear spring of spring constant k . If $y(t)$ denotes the distance of the center of the mass from a position of rest of the spring, by applying Newton's law the following second-order linear ordinary differential equation with constant coefficients is obtained:

$$m\ddot{y}(t) + b\dot{y}(t) + ky(t) = u(t). \quad (2)$$

Here $\dot{y}(t) = \frac{dy(t)}{dt}$. The motion of the mass $y(t)$, $t > 0$ is uniquely determined if the applied force $u(t)$, $t \geq 0$ is known and at $t = 0$ the initial position $y(0) = y_0$ and initial velocity $\dot{y}(0) = y_1$ are given. To obtain a state variable description, introduce the state variables x_1 and x_2 as

$$x_1(t) = y(t), \quad x_2(t) = \dot{y}(t)$$

to obtain the set of first-order differential equations $m\dot{x}_2(t) + bx_2(t) + kx_1(t) = u(t)$ and $\dot{x}(t) = \begin{bmatrix} x_2(t) \\ \frac{1}{m}(-kx_1(t) - bx_2(t) + u(t)) \end{bmatrix}$ which can be rewritten in the form of (1)

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t) \quad (3)$$

and

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

with $\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix}$ as initial conditions. This is of the form (1) where $x(t)$ is a 2-dimensional column vector; A is a 2×2 matrix; B and C are 2-dimensional column and row vectors, respectively; and $x(0) = x_0$.

Notes:

1. It is always possible to obtain a state variable description which is equivalent to a given set of higher-order differential equations
2. The choice of the state variables, here x_1 and x_2 , is not unique. Different choices will lead to different A , B , and C .
3. The number of the state variables is typically equal to the order of the set of the higher-order differential equations and equals the number of initial conditions needed to derive the unique solution; in the above example this number is 2.
4. In time-invariant systems, it can be assumed without loss of generality that the starting time is $t = 0$ and so the initial conditions are taken to be $x(0) = x_0$.

Solving $\dot{x} = A(t)x$; $x(0) = x_0$

Consider the homogeneous equation

$$\dot{x} = A(t)x; \quad x(0) = x_0 \quad (4)$$

where $x(t) = [x_1(t), \dots, x_n(t)]^T$ is the state vector of dimension n and A is an $n \times n$ matrix with entries real numbers (i.e., $A \in \mathbb{R}^{n \times n}$).

Equation (4) is a special case of (1) where there are no inputs and outputs, u and y . The homogeneous vector differential equation (4) will

be solved first, and its solution will be used to find the solution of (1).

Solving (4) is an initial value problem. It can be shown that there always exists a unique solution $\varphi(t)$ such that

$$\dot{\varphi}(t) = A\varphi(t); \quad \varphi(0) = x_0.$$

To find the unique solution, consider first the one-dimensional case, namely,

$$\dot{y}(t) = ay(t); \quad y(0) = y_0$$

the unique solution of which is

$$y(t) = e^{at} y_0, \quad t \geq 0.$$

The scalar exponential e^{at} can be expressed in a series form

$$e^{at} = \sum_{k=0}^{\infty} \frac{t^k}{k!} a^k (= 1 + \frac{1}{1}ta + \frac{1}{2}t^2a^2 + \frac{1}{6}t^3a^3 + \dots)$$

The generalization to the $n \times n$ matrix exponential (A is $n \times n$) is given by

$$e^{At} = \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k \quad (= I_n + At + \frac{1}{2}A^2t^2 + \dots) \quad (5)$$

By analogy, let the solution to (4) be

$$(x(t) =) \varphi(t) = e^{At} x_0 \quad (6)$$

It is a solution since if it is substituted into (4),

$$\begin{aligned} \dot{\varphi}(t) &= [A + At + \frac{1}{2}A^2t^2 + \dots]x_0 \\ &= Ae^{At}x_0 = A\varphi(t) \end{aligned}$$

and $\varphi(0) = e^{A \cdot 0}x_0 = x_0$, that is, it satisfies the equation and the initial condition. Since the solution of (4) is unique, (6) is the unique solution of (4).

The solution (6) can be derived more formally using the Peano-Baker series (see ► [Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions](#)), which in the present time-invariant case becomes the defining series for the matrix exponential (5).

System Response

Based on the solution of the homogeneous equation (4), shown in (6), the solution of the state equation in (1) can be shown to be

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau. \quad (7)$$

The following properties for the matrix exponential e^{At} can be shown directly from the defining series:

1. $Ae^{At} = e^{At}A$.
2. $(e^{At})^{-1} = e^{-At}$.

Equation (7) which is known as the *variation of constants formula* can be derived as follows:

Consider $\dot{x} = Ax + Bu$ and let $z(t) = e^{-At}x(t)$. Then $x(t) = e^{At}z(t)$ and substituting

$$Ae^{At}z(t) + e^{At}\dot{z}(t) = Ae^{At}z(t) + Bu(t)$$

or $\dot{z}(t) = e^{-At}Bu(t)$ from which

$$z(t) - z(0) = \int_0^t e^{-A\tau}Bu(\tau)d\tau$$

or

$$e^{-At}x(t) - x(0) = \int_0^t e^{-A\tau}Bu(\tau)d\tau$$

or

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$$

which is the variation of constants formula (7).

Equation (7) is the sum of two parts, the *state response* (when $u(t) = 0$ and the system is driven only by the initial state conditions) and the *input response* (when $x_0 = 0$ and the system is driven only by the input $u(t)$); this illustrates the linear system principle of superposition.

If the output equation $y(t) = Cx(t) + Du(t)$ is considered, then in view of (7),

$$\begin{aligned}
 y(t) &= Ce^{At}x_0 + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t) \\
 &= Ce^{At}x_0 + \int_0^t [Ce^{A(t-\tau)}B \\
 &\quad + D\delta(t-\tau)]u(\tau)d\tau
 \end{aligned} \tag{8}$$

The second expression involves the Dirac (or impulse or delta) function $\delta(t)$, and it is derived based on the basic property for $\delta(t)$, namely,

$$f(t) = \int_{-\infty}^{\infty} \delta(t-\tau)f(\tau)d\tau$$

It is clear that the matrix exponential e^{At} plays a central role in determining the response of a linear continuous-time, time-invariant system described by (1).

Given A , e^{At} may be determined using several methods including its defining series, diagonalization of A using a similarity transformation (PAP^{-1}), the Cayley-Hamilton theorem, using expressions involving the modes of the system ($e^{At} = \sum_{i=1}^n A_i e^{\lambda_i t}$ when A has n distinct eigenvalues λ_i ; $A_i = v_i \tilde{v}_i$ with $v_i, \tilde{v}_i$ the right and left eigenvectors of A that correspond to λ_i ($\tilde{v}_i v_j = 1, i = j$ and $\tilde{v}_i v_j = 0, i \neq j$)), or using Laplace transform ($e^{At} = \mathcal{L}^{-1}[(sI - A)^{-1}]$). See references below for detailed algorithms.

Equivalent State Variable Descriptions

Given

$$\dot{x} = Ax + Bu, \quad y = Cx + Du \tag{9}$$

consider the new state vector $\tilde{x}$ where

$$\tilde{x} = Px$$

with P a real nonsingular matrix. Substituting $x = P^{-1}\tilde{x}$ in (9), we obtain

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}u, \quad y = \tilde{C}\tilde{x} + \tilde{D}u, \tag{10}$$

where

$$\tilde{A} = PAP^{-1}, \quad \tilde{B} = PB, \quad \tilde{C} = CP^{-1}, \quad \tilde{D} = D$$

The state variable descriptions (9) and (10) are called equivalent and P is the equivalence transformation. This transformation corresponds to a change in the basis of the state space, which is a vector space. Appropriately selecting P , one can simplify the structure of $\tilde{A}(= PAP^{-1})$; the matrices $\tilde{A}$ and A are called similar. When the eigenvectors of A are all linearly independent (this is the case, e.g., when all eigenvalues λ_i of A are distinct), then P may be found so that $\tilde{A}$ is diagonal. When e^{At} is to be determined, and $\tilde{A} = PAP^{-1} = \text{diag}[\lambda_i]$ ($\tilde{A}$ and A have the same eigenvalues), then

$$e^{At} = e^{P^{-1}\tilde{A}Pt} = P^{-1}e^{\tilde{A}t}P = P^{-1}\text{diag}[e^{\lambda_i t}]P.$$

Note that it can be easily shown that equivalent state space representations give rise to the same impulse response and transfer function (see [► Linear Systems: Continuous-Time Impulse Response Descriptions](#)).

Summary

State variable descriptions for continuous-time time-invariant systems are introduced and the state and output responses to inputs and initial conditions are derived. Equivalence of state variable representations is also discussed.

Cross-References

- [Linear Systems: Continuous-Time Impulse Response Descriptions](#)
- [Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions](#)
- [Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions](#)

Recommended Reading

The state variable description of systems received wide acceptance in systems theory beginning in the late 1950s. This was primarily due to the work of R.E. Kalman and others in filtering theory and quadratic control theory and to the

work of applied mathematicians concerned with the stability theory of dynamical systems. For comments and extensive references on some of the early contributions in these areas, see Kailath (1980) and Sontag (1990). The use of state variable descriptions in systems and control opened the way for the systematic study of systems with multi-inputs and multi-outputs.

Bibliography

- Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston
- Brockett RW (1970) Finite dimensional linear systems. Wiley, New York
- Chen CT (1984) Linear system theory and design. Holt, Rinehart and Winston, New York
- DeCarlo RA (1989) Linear systems. Prentice-Hall, Englewood Cliffs
- Hespanha JP (2009) Linear systems theory. Princeton Press, Princeton
- Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs
- Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs
- Sontag ED (1990) Mathematical control theory: deterministic finite dimensional systems. Texts in applied mathematics, vol 6. Springer, New York
- Zadeh LA, Desoer CA (1963) Linear system theory: the state space approach. McGraw-Hill, New York

Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions

Panos J. Antsaklis
Department of Electrical Engineering, University
of Notre Dame, Notre Dame, IN, USA

Abstract

Continuous-time processes that can be modeled by linear differential equations with time-varying coefficients can be written in terms of state variable descriptions of the form $\dot{x}(t) = A(t)x(t) + B(t)u(t)$, $y(t) = C(t)x(t) + D(t)u(t)$. The response of such systems due to

a given input and initial conditions is derived using the Peano-Baker series. Equivalence of state variable descriptions is also discussed.

Keywords

Continuous-time · Linear systems · State variable descriptions · Time-varying

Introduction

Dynamical processes that can be described or approximated by linear high-order ordinary differential equations with time-varying coefficients can also be described, via a change of variables, by state variable descriptions of the form

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) + B(t)u(t); & x(t_0) &= x_0 \\ y(t) &= C(t)x(t) + D(t)u(t),\end{aligned}\quad (1)$$

where $x(t)$ ($t \in \mathbb{R}$, the set of reals) is a column vector of dimension n ($x(t) \in \mathbb{R}^n$) and $A(t)$, $B(t)$, $C(t)$, $D(t)$ are matrices with entries functions of time t . $A(t) = [a_{ij}(t)]$, $a_{ij}(t) : \mathbb{R} \rightarrow \mathbb{R}$. $A(t) \in \mathbb{R}^{n \times n}$, $B(t) \in \mathbb{R}^{n \times m}$, $C(t) \in \mathbb{R}^{p \times n}$, $D(t) \in \mathbb{R}^{p \times m}$. The input vector is $u(t) \in \mathbb{R}^m$ and the output vector is $y(t) \in \mathbb{R}^p$. The vector differential equation in (1) is the *state equation*, while the algebraic equation is the *output equation*.

The advantage of the state variable description (1) is that given an input $u(t)$, $t \geq 0$ and an initial condition $x(t_0) = x_0$, the state trajectory or motion for $t \geq t_0$ can be conveniently characterized. To derive the expressions, we first consider the homogenous state equation and the corresponding initial value problem.

Solving $\dot{x}(t) = A(t)x(t)$; $x(t_0) = x_0$

Consider the homogenous equation with the initial condition

$$x(t) = A(t)x(t); \quad x(t_0) = x_0 \quad (2)$$

where $x(t) = [x_1(t), \dots, x_n(t)]^T$ is the state (column) vector of dimension n and $A(t)$ is

an $n \times n$ matrix with entries functions of time that take on values from the field of reals ($A \in \mathbb{R}^{n \times n}$).

Under certain assumptions on the entries of $A(t)$, a solution of (2) exists and it is unique. These assumptions are satisfied, and a solution exists and is unique in the case, for example, when the entries of $A(t)$ are continuous functions of time. In the following we make this assumption.

To find the unique solution of (2), we use the *method of successive approximations* which when applied to

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0 \quad (3)$$

is described by

$$\begin{aligned} \phi_0(t) &= x_0 \\ \phi_m(t) &= x_0 + \int_{t_0}^t f(\tau, \phi_{m-1}(\tau)) d\tau, \quad m = 1, 2, \dots \end{aligned} \quad (4)$$

As $m \rightarrow \infty$, ϕ_m converges to the unique solution of (3), assuming the f satisfies certain conditions.

Applying the method of successive approximations to (2) yields

$$\begin{aligned} \phi_0(t) &= x_0 \\ \phi_1(t) &= x_0 + \int_{t_0}^t A(\tau)x_0 d\tau \\ \phi_2(t) &= x_0 + \int_{t_0}^t A(\tau)\phi_1(\tau)x_0 d\tau \\ &\vdots \\ \phi_m(t) &= x_0 + \int_{t_0}^t A(\tau)\phi_{m-1}(\tau)x_0 d\tau \end{aligned}$$

from which

$$\begin{aligned} \phi_m(t) &= \left[I + \int_{t_0}^t A(\tau_1) d\tau_1 \right. \\ &\quad \left. + \int_{t_0}^t A(\tau_1) \int_{t_0}^{\tau_1} A(\tau_2) d\tau_2 d\tau_1 + \dots \right] x_0 \end{aligned}$$

$$\begin{aligned} &+ \int_{t_0}^t A(\tau_1) \int_{t_0}^{\tau_1} A(\tau_2) \dots \int_{t_0}^{\tau_{m-1}} A(\tau_m) \\ &\quad d\tau_m \dots d\tau_1 \Big] x_0 \end{aligned}$$

When $m \rightarrow \infty$, and under the above continuity assumptions on $A(t)$, $\phi_m(t)$ converges to the unique solution of (2), i.e.,

$$\phi(t) = \Phi(t, t_0)x_0 \quad (5)$$

where

$$\begin{aligned} \Phi(t, t_0) &= I + \int_{t_0}^t A(\tau_1) d\tau_1 \\ &+ \int_{t_0}^t A(\tau_1) \left[\int_{t_0}^{\tau_1} A(\tau_2) d\tau_2 \right] d\tau_1 + \dots \end{aligned} \quad (6)$$

Note that $\Phi(t_0, t_0) = I$ and by differentiation it can be seen that

$$\dot{\phi}(t, t_0) = A(t)\phi(t, t_0), \quad (7)$$

as expected, since (5) is the solution of (2). The $n \times n$ matrix $\Phi(t, t_0)$ is called the *state transition matrix* of (2). The defining series (6) is called the Peano-Baker series.

Note that when $A(t) = A$, a constant matrix, then (6) becomes

$$\Phi(t, t_0) = I + \sum_{k=1}^{\infty} \frac{A^k(t-t_0)^k}{k!} \quad (8)$$

which is the defining series for the matrix exponential $e^{A(t-t_0)}$ (see ► [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)).

System Response

Based on the solution (5) of $\dot{x} = A(t)x(t)$, the solution of the non-homogenous equation

$$\dot{x}(t) = A(t)x(t) + B(t)u(t); \quad x(t_0) = x_0 \quad (9)$$

can be shown to be

$$\phi(t) = \Phi(t, t_0)x_0 + \int_{t_0}^t \Phi(t, \tau)B(\tau)u(\tau)d\tau. \quad (10)$$

Equation (10) is the *variation of constants formula*. This result can be shown via direct substitution of (10) into (9); note that $\phi(t) = \Phi(t_0, t_0)x_0 = x_0$. That (10) is a solution can also be shown using a change of variables in (9), namely,

$$z(t) = \Phi(t_0, t)x(t).$$

Equation (10) is the sum of two parts, the *state response* (when $u(t) = 0$ and the system is driven only by the initial state conditions) and the *input response* (when $x_0 = 0$ and the system is driven only by the input $u(t)$); this illustrates the linear system principle of superposition.

In view of (10), the output $y(t) (= C(t)x(t) + D(t)u(t))$ is

$$\begin{aligned} y(t) &= C(t)\Phi(t, t_0)x_0 \\ &\quad + \int_{t_0}^t C(t)\Phi(t, \tau)B(\tau)u(\tau)d\tau + D(t)u(t) \\ &= C(t)\Phi(t, t_0)x_0 + \int_{t_0}^t [C(t)\Phi(t, \tau)B(\tau) \\ &\quad + D(t)\delta(t - \tau)]u(\tau)d\tau \end{aligned}$$

The second expression involves the Dirac (or impulse or delta) function $\delta(t)$ and is derived based on the basic property for $\delta(t)$, namely,

$$f(t) = \int_{-\infty}^{+\infty} \delta(t - \tau)f(\tau)d\tau,$$

where $\delta(t - \tau)$ denotes an impulse applied at time $\tau = t$.

Properties of the State Transition Matrix $\Phi(t, t_0)$

In general it is difficult to determine $\Phi(t, t_0)$ explicitly; however, $\Phi(t, t_0)$ may be readily determined in a number of special cases including the cases in which $A(t) = A$, $A(t)$ is diagonal, $A(t)A(\tau) = A(\tau)A(t)$.

Consider $\dot{x} = A(t)x$. We can derive a number of important properties which are described below. It can be shown that given n linearly independent initial conditions x_{0i} , the corresponding n solutions $\phi_i(t)$ are also linearly independent. Let a *fundamental matrix* $\Psi(t)$ of $\dot{x} = A(t)x$ be an $n \times n$ matrix, the columns of which are a set of linearly independent solutions $\phi_1(t), \dots, \phi_n(t)$. The state transition matrix Φ is the fundamental matrix determined from solutions that correspond to the initial conditions $[1, 0, 0, \dots]^T, [0, 1, 0, \dots, 0]^T, \dots, [0, 0, \dots, 1]^T$ (recall that $\Phi(t_0, t_0) = I$). The following are properties of $\Phi(t, t_0)$:

- (i) $\Phi(t, t_0) = \Psi(t)\Psi^{-1}(t_0)$ with $\Psi(t)$ any fundamental matrix.
- (ii) $\Phi(t, t_0)$ is nonsingular for all t and t_0 .
- (iii) $\Phi(t, \tau) = \Phi(t, \sigma)\Phi(\sigma, \tau)$ (semigroup property).
- (iv) $[\Phi(t, t_0)]^{-1} = \Phi(t_0, t)$.

In the special case of time-invariant systems and $\dot{x} = Ax$, the above properties can be written in terms of the matrix exponential since

$$\Phi(t, t_0) = e^{A(t-t_0)}.$$

Equivalence of State Variable Descriptions

Given the system

$$\begin{aligned} \dot{x} &= A(t)x + B(t)u \\ y &= C(t)x + D(t)u \end{aligned} \quad (11)$$

consider the new state vector $\tilde{x}$

$$\tilde{x}(t) = P(t)x(t)$$

where $P^{-1}(t)$ exists and P and P^{-1} are continuous. Then the system

$$\begin{aligned}\dot{\tilde{x}} &= \tilde{A}(t)\tilde{x} + \tilde{B}(t)u \\ y &= \tilde{C}(t)\tilde{x} + \tilde{D}(t)u\end{aligned}$$

where

$$\begin{aligned}\tilde{A}(t) &= [P(t)A(t) + \dot{P}(t)]P^{-1}(t) \\ \tilde{B}(t) &= P(t)B(t) \\ \tilde{C}(t) &= C(t)P^{-1}(t) \\ \tilde{D}(t) &= D(t)\end{aligned}$$

is equivalent to (1). It can be easily shown that equivalent descriptions give rise to the same impulse responses.

Summary

State variable descriptions for continuous-time time-varying systems were introduced and the state and output responses to inputs and initial conditions were derived. The equivalence of state variable representations was also discussed.

Cross-References

- [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)
- [Linear Systems: Continuous-Time Impulse Response Descriptions](#)
- [Linear Systems: Discrete-Time, Time-Varying, State Variable Descriptions](#)

Recommended Reading

Additional information regarding the time-varying case may be found in Brockett (1970), Rugh (1996), and Antsaklis and Michel (2006). For historical comments and extensive references on some of the early contributions, see Sontag (1990) and Kailath (1980).

Bibliography

- Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston
- Brockett RW (1970) Finite dimensional linear systems. Wiley, New York
- Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs
- Miller RK, Michel AN (1982) Ordinary differential equations. Academic, New York
- Rugh WJ (1996) Linear system theory, 2nd edn. Prentice-Hall, Englewood Cliffs
- Sontag ED (1990) Mathematical control theory: deterministic finite dimensional systems. Texts in applied mathematics, vol 6. Springer, New York

Linear Systems: Discrete-Time Impulse Response Descriptions

Panos J. Antsaklis

Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

An important input-output description of a linear discrete-time system is its (discrete-time) impulse response (or pulse response), which is the response $h(k, k_0)$ to a discrete impulse applied at time k_0 . In time-invariant systems that are also causal and at rest at time zero, the impulse response is $h(k, 0)$, and its z -transform is the transfer function of the system. Expressions for $h(k, k_0)$ when the system is described by state variable equations are derived.

Keywords

At rest · Causal · Discrete-time · Discrete-time impulse response descriptions · Linear systems · Pulse response descriptions · Time-invariant · Time-varying · Transfer function descriptions

Introduction

Consider linear, discrete-time dynamical systems that can be described by

$$y(k) = \sum_{l=-\infty}^{+\infty} H(k, l)u(l) \quad (1)$$

where $k, l \in \mathbb{Z}$ is the set of integers, the output is $y(k) \in \mathbb{R}^p$, the input is $u(k) \in \mathbb{R}^m$, and $H(k, l) : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{R}^{p \times m}$. For instance, any system that can be written in state variable form

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) \\ y(k) &= C(k)x(k) + D(k)u(k) \end{aligned} \quad (2)$$

or

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (3)$$

can be represented by (1). Note that it is assumed that at $l = -\infty$, the system is at rest, i.e., no energy is stored in the system at time $-\infty$.

Define the discrete-time impulse (or unit pulse) as

$$\delta(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}, k \in \mathbb{Z}$$

and consider a single-input, single-output system:

$$y(k) = \sum_{l=-\infty}^{+\infty} h(k, l)u(l) \quad (4)$$

If $u(l) = \delta(\hat{l} - l)$, that is, the input is a unit pulse applied at $l = \hat{l}$, then the output is

$$y_I(k) = h(k, \hat{l}),$$

i.e., $h(k, \hat{l})$ is the output at time k when a unit pulse is applied at time $\hat{l}$.

So in (4) $h(k, l)$ is the response at time k to a discrete-time impulse (unit pulse) applied at time l . $h(k, l)$ is the *discrete-time impulse response* of the system. Clearly if $h(k, l)$ is known, the response of the system to any input can be

determined via (4). So $h(k, l)$ is an input/output description of the system.

Equation (1) is a generalization of (4) for the multi-input, multi-output case. If we let all the components of $u(l)$ in (1) be zero except for the j th component, then

$$y_i(k) = \sum_{l=-\infty}^{+\infty} h_{ij}(k, l)u_j(l) \quad (5)$$

$h_{ij}(k, l)$ denotes the response of the i th component of the output of system (1) at time k due to a discrete impulse applied to the j th component of the input at time l with all remaining components of the input being zero. $H(k, l) = [h_{ij}(k, l)]$ is called the *impulse response matrix* of the system.

If it is known that system (1) is *causal*, then the output will be zero before an input is applied. Therefore,

$$H(k, l) = 0, \quad \text{for } k < l, \quad (6)$$

and so when causality is present, (1) becomes

$$y(k) = \sum_{l=-\infty}^k H(k, l)u(l). \quad (7)$$

A system described by (1) is at rest at $k = k_0$ if $u(k) = 0$ for $k \geq k_0$ implies $y(k) = 0$ for $k \geq k_0$. For a system at rest at $k = k_0$, (7) becomes

$$y(k) = \sum_{l=k_0}^k H(k, l)u(l). \quad (8)$$

If system (1) is time-invariant, then $H(k, l) = H(k - l, 0)$ (also written as $H(k - l)$) since only the time elapsed $(k - l)$ from the application of the discrete-time impulse is important. Then (8) becomes

$$y(k) = \sum_{l=0}^k H(k - l)u(l), \quad k \geq 0, \quad (9)$$

where we chose $k_0 = 0$ without loss of generality. Equation (9) is the description for casual, time-invariant systems, at rest at $k = 0$.

Equation (9) is a convolution sum and if we take the (one-sided or unilateral) z -transform of both sides,

$$\hat{y}(z) = \hat{H}(z)\hat{u}(z), \quad (10)$$

where $\hat{y}(z)$, $\hat{u}(z)$ are the z -transforms of $y(k)$, $u(k)$ and $\hat{H}(z)$ is the z -transform of the discrete-time impulse response $H(k)$. $\hat{H}(z)$ is the *transfer function matrix* of the system. Note that the transfer function of a linear, time-invariant system is defined as the rational matrix $\hat{H}(z)$ that satisfies (10) for any input and its corresponding output assuming zero initial conditions.

Connections to State Variable Descriptions

When a system is described by (2), then

$$y(k) = \sum_{l=k_0}^{k-1} C(k)\Phi(k, l+1)B(l)u(l) + D(k)u(k), \quad k > k_0 \quad (11)$$

where it was assumed that $x(k_0) = 0$, i.e., the system is at rest at k_0 . Here $\Phi(k, l) (= A(k-1) \cdots A(l))$ is the state transition matrix of the system.

Comparing (11) with (8), the discrete-time impulse response of the system is

$$H(k, l) = \begin{cases} C(k)\Phi(k, l+1)B(l) & k > l \\ D(k) & k = l \\ 0 & k < l \end{cases} \quad (12)$$

Similarly, when the system is time-invariant and is described by (3),

$$y(k) = \sum_{l=k_0}^{k-1} CA^{k-(l+1)}Bu(l) + Du(k), \quad k > k_0 \quad (13)$$

where $x(k_0) = 0$ and

$$H(k, l) = H(k-l) = \begin{cases} CA^{k-(l+1)}B & k > l \\ D & k = l \\ 0 & k < l \end{cases} \quad (14)$$

When $l = 0$ (taking the time when the discrete impulse is applied to be zero, $l = 0$), the discrete-time impulse response is

$$H(k) = \begin{cases} CA^{k-1}B & k > 0 \\ D & k = 0 \\ 0 & k < 0 \end{cases} \quad (15)$$

Taking (one-sided or unilateral) z -transforms of both sides in (15),

$$\hat{H}(z) = C(zI - A)^{-1}B + D \quad (16)$$

which is the transfer function matrix in terms of the coefficient matrices in the state variable description (3). Note that (16) can also be derived directly from (3) by assuming zero initial conditions ($x(0) = 0$) and taking z -transforms of both sides.

Finally, it is easy to show that equivalent state variable descriptions give rise to the same discrete-impulse response.

Summary

The discrete-time impulse response is an external, input-output description of linear, discrete-time systems. When the system is time-invariant, the z -transform of the impulse response $h(k, 0)$ (which is the output response at time k due to a discrete impulse applied at time zero with initial conditions taken to be zero) is the transfer function – another very common input-output description. The relationships to the state variable descriptions were shown.

Cross-References

- [Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions](#)
- [Linear Systems: Discrete-Time, Time-Varying, State Variable Descriptions](#)

Recommended Reading

External or input-output descriptions such as the impulse response and the transfer function (in the time-invariant case) are described in several textbooks below.

Bibliography

- Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston
 Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs
 Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs

Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions

Panos J. Antsaklis
 Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

Discrete-time processes that can be modeled by linear difference equations with constant coefficients can also be described in a systematic way in terms of state variable descriptions of the form $x(k+1) = Ax(k) + Bu(k)$, $y(k) = Cx(k) + Du(k)$. The response of such systems due to a given input and subject to initial conditions is derived. Equivalence of state variable descriptions is also discussed.

Keywords

Discrete-time · Linear systems · State variable descriptions · Time-invariant

Introduction

Discrete-time systems arise in a variety of ways in the modeling process. There are systems that

are inherently defined only at discrete points in time; examples include digital devices, inventory systems, economic systems such as banking where interest is calculated and added to savings accounts at discrete time interval, etc. There are also systems that describe continuous-time systems at discrete points in time; examples include simulations of continuous processes using digital computers and feedback control systems that employ digital controllers and give rise to sampled-data systems.

Linear, discrete-time, time-invariant systems can be modeled via state variable equations, namely,

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k); \quad x(0) = x_0 \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (1)$$

where $k \in \mathbb{Z}$, the set of integers, the state vector $x \in \mathbb{R}^n$, i.e., an n dimensional column vector; $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$ are matrices with entries of real numbers; and $y(k) \in \mathbb{R}^p$, $u(k) \in \mathbb{R}^m$ the output and the input, respectively. The vector difference equation in (1) is the *state equation* and the algebraic equation is the *output equation*.

Note that (1) could have been equivalently written as $x(l) = Ax(l-1) + Bu(l-1)$ where $l = k+1$ and $x(l-1)$ is an easily visualized delayed version of $x(l)$; this is a form more common in signal processing (where a two-sided or bilateral z -transform is used). In control where we assume a known initial condition at time equal to zero (and one-sided or unilateral z -transform is taken), the form in (1) is common.

Similar to the continuous-time case, (1) can be derived from a set of high-order difference equations by introducing the state variables $x(k) = [x_1(k), \dots, x_n(k)]^T$. Description (1) can also be derived from continuous-time system descriptions by sampling (see ► [Sampled-Data Systems](#)).

The advantage of the above state variable description is that given any input $u(k)$ and initial conditions $x(0)$, its solution (state trajectory or motion) can be conveniently and systematically characterized. This is done below. We first

consider the solutions of the homogenous equation $x(k+1) = Ax(k)$.

Solving $x(k+1) = Ax(k)$; $x(0) = x_0$

Consider the homogenous equation

$$x(k+1) = Ax(k); \quad x(0) = x_0 \quad (2)$$

where $k \in \mathbb{Z}^+$ is a nonnegative integer, $x(k) = [x_1(k), \dots, x_n(k)]^T$ is the state column vectors of dimension n , and A is an $n \times n$ matrix with entries real numbers (i.e., $A \in \mathbb{R}^{n \times n}$).

Write (2) for $k = 0, 1, 2, \dots$, namely, $x(1) = Ax(0)$, $x(2) = Ax(1) = A^2x(0)$, $\dots$ to derive the solution

$$x(k) = A^k x(0), \quad k \geq 0 \quad (3)$$

This result can be shown formally by induction. Note that $A^0 = I$ by convention and so (3) also satisfies the initial condition.

If the initial time were some (integer) k_0 instead of zero, then the solution would be

$$x(k) = A^{k-k_0} x(k_0), \quad k \geq k_0 \quad (4)$$

The solution can be written as

$$\begin{aligned} x(k) &= \Phi(k, k_0)x(k_0) \\ &= \Phi(k - k_0, 0)x(k_0), \quad k \geq k_0 \end{aligned} \quad (5)$$

where $\Phi(k, k_0)$ is the state transition matrix and it equals $\Phi(k, k_0) = A^{k-k_0}$. Note that for time-invariant systems, the initial time k_0 can always be taken to be zero without loss of generality; this is because the behavior depends only on the time elapsed ($k - k_0$) and not on the actual initial time k_0 .

In view of (3), it is clear that A^k plays an important role in the solutions of the difference state equations that describe linear, discrete-time, time-invariant systems; it is actually analogous to the role e^{At} plays in the solutions of the linear differential state equations that describe linear, continuous-time, time-invariant systems.

Notice that in (3), $k \geq 0$. This is so because A^k for $k < 0$ may not exist; this is the case, for example, when A is a singular matrix – it has at least one eigenvalue at the origin. In contrast, e^{At} exists for any t positive or negative. The implication is that in discrete-time systems we may not be able to determine uniquely the initial past state $x(0)$ from a current state value $x(k)$; in contrast, in continuous-time systems, it is always possible to go backwards in time.

There are several methods to calculate A^k that mirror the methods to calculate e^{At} . One could, for example, use similarity transformations, or the z -transform. When all eigenvectors of A are linearly independent (this is the case, e.g., when all eigenvalues λ_i of A are distinct), then a similarity transformation exists so that

$$PAP^{-1} = \tilde{A} = \text{diag}[\lambda_i].$$

Then

$$A^k = P^{-1} \tilde{A}^k P = P^{-1} \begin{bmatrix} \lambda_1^k & & \\ & \ddots & \\ & & \lambda_n^k \end{bmatrix} P.$$

Alternatively, using the z -transforms, $A^k = z^{-1}\{z(zI - A)^{-1}\}$. Also when the eigenvalues λ_i of A are distinct, then

$$A^k = \sum_{i=1}^n A_i \lambda_i^k,$$

where $A_i = v_i \tilde{v}_i$ with $v_i, \tilde{v}_i$ the right and left eigenvectors of A that correspond to λ_i . Note that

$$\begin{bmatrix} \tilde{v}_1 \\ \vdots \\ \tilde{v}_n \end{bmatrix} = [v_1 \cdots v_n]^{-1},$$

$A_i \lambda_i^k$ are the modes of the system. One could also use the Cayley-Hamilton theorem to determine A^k .

System Response

Consider the description (1). The response can be easily derived by writing the equation for $k = 0, 1, 2, \dots$ and substituting or formally by induction. It is

$$x(k) = A^k x(0) + \sum_{j=0}^{k-1} A^{k-(j+1)} B u(j), \quad k > 0 \quad (6)$$

and

$$\begin{aligned} y(k) &= C A^k x(0) + \sum_{j=0}^{k-1} C A^{k-(j+1)} B u(j) \\ &\quad + D u(k), \quad k > 0 \\ y(0) &= C x(0) + D u(0). \end{aligned} \quad (7)$$

Note that (6) can also be written as

$$x(k) = A^k x(0) + [B, AB, \dots, A^{k-1}B] \begin{bmatrix} u(k-1) \\ \vdots \\ u(0) \end{bmatrix}. \quad (8)$$

Clearly the response is the sum of two components, one due to the initial condition (state response) and one due to the input (input response). This illustrates the linear system principle of superposition.

If the initial time is k_0 and (4) is used, then

$$\begin{aligned} y(k) &= C A^{k-k_0} x(k_0) + \sum_{j=k_0}^{k-1} C A^{k-(j+1)} B u(j) \\ &\quad + D u(k), \quad k > k_0 \\ y(k_0) &= C x(k_0) + D u(k_0). \end{aligned} \quad (9)$$

Equivalence of State Variable Descriptions

Given description (1), consider the new state vector $\tilde{x}$ where

$$\tilde{x}(k) = P x(k)$$

with $P \in \mathbb{R}^{n \times n}$ a real nonsingular matrix.

Substituting $x = P^{-1} \tilde{x}$ in (1), we obtain

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{A} \tilde{x}(k) + \tilde{B} u(k) \\ y(k) &= \tilde{C} \tilde{x}(k) + \tilde{D} u(k) \end{aligned} \quad (10)$$

where

$$\tilde{A} = P A P^{-1}, \quad \tilde{B} = P B, \quad \tilde{C} = C P^{-1}, \quad \tilde{D} = D$$

The state variable descriptions (1) and (9) are called equivalent and P is the equivalence transformation matrix. This transformation corresponds to a change in the basis of the state space, which is a vector space. Appropriately selecting P one can simplify the structure of $\tilde{A}$ ($= P A P^{-1}$). It can be easily shown that equivalent description gives rise to the same discrete impulse response and transfer function.

Summary

State variable descriptions for discrete-time, time-invariant systems were introduced and the state and output responses to inputs and initial conditions were derived. The equivalence of state variable representations was also discussed.

Cross-References

- [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)
- [Linear Systems: Discrete-Time Impulse Response Descriptions](#)
- [Linear Systems: Discrete-Time, Time-Varying, State Variable Descriptions](#)
- [Sampled-Data Systems](#)

Recommended Reading

The state variable descriptions received wide acceptance in systems theory beginning in the late 1950s. This was primarily due to the work of R.E. Kalman. For historical comments and extensive references, see Kailath (1980). The

use of state variable descriptions in systems and control opened the way for the systematic study of systems with multi-inputs and multi-outputs.

Bibliography

- Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston
- Franklin GF, Powell DJ, Workman ML (1998) Digital control of dynamic systems, 3rd edn. Addison-Wesley, Longman, Inc., Menlo Park, CA
- Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs
- Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs

Linear Systems: Discrete-Time, Time-Varying, State Variable Descriptions

Panos J. Antsaklis

Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

Discrete-time processes that can be modeled by linear difference equations with time-varying coefficients can be written in terms of state variable descriptions of the form $x(k+1) = A(k)x(k) + B(k)u(k)$, $y(k) = C(k)x(k) + D(k)u(k)$. The response of such systems due to a given input and initial conditions is derived. Equivalence of state variable descriptions is also discussed.

Keywords

Discrete-time · Linear systems · State variable descriptions · Time-varying

Introduction

Discrete-time systems arise in a variety of ways in the modeling process. There are systems that are inherently defined only at discrete points in

time; examples include digital devices, inventory systems, and economic systems such as banking where interest is calculated and added to savings accounts at discrete time interval. There are also systems that describe continuous-time systems at discrete points in time; examples include simulations of continuous processes using digital computers and feedback control systems that employ digital controllers and give rise to sampled-data systems.

Dynamical processes that can be described or approximated by linear difference equations with time-varying coefficients can also be described, via a change of variables, by state variable descriptions of the form

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k); \quad x(k_0) = x_0 \\ y(k) &= C(k)x(k) + D(k)u(k). \end{aligned} \quad (1)$$

Above, the state vector $x(k)$ ($k \in \mathbb{Z}$, the set of integers) is a column vector of dimension n ($x(k) \in \mathbb{R}^n$); the output is $y(k) \in \mathbb{R}^m$ and the input is $u(k) \in \mathbb{R}^m$. $A(k)$, $B(k)$, $C(k)$, and $D(k)$ are matrices with entries functions of time k , $A(k) = [a_{ij}(k)]$, $a_{ij}(k) : \mathbb{Z} \rightarrow \mathbb{R}$ ($A(k) \in \mathbb{R}^{n \times n}$, $B(k) \in \mathbb{R}^{n \times m}$, $C(k) \in \mathbb{R}^{p \times n}$, $D(k) \in \mathbb{R}^{p \times m}$). The vector difference equation in (1) is the state equation, while the algebraic equation is the output equation. Note that in the time-invariant case, $A(k) = A$, $B(k) = B$, $C(k) = C$, and $D(k) = D$.

The advantage of the state variable description (1) is that given an input $u(k)$, $k \geq k_0$ and an initial condition $x(k_0) = x_0$, the state trajectories or motions for $k \geq k_0$ can be conveniently characterized. To determine the expressions, we first consider the homogeneous state equation and the corresponding initial value problem.

Solving $x(k+1) = A(k)x(k)$; $x(k_0) = x_0$

Consider the homogenous equation

$$x(k+1) = A(k)x(k); \quad x(k_0) = x_0 \quad (2)$$

Note that

$$\begin{aligned}
 x(k_0 + 1) &= A(k_0)x(k_0) \\
 x(k_0 + 2) &= A(k_0 + 1)A(k_0)x(k_0) \\
 &\vdots \\
 x(k) &= A(k-1)A(k-2) \cdots A(k_0)x(k_0) \\
 &= \prod_{j=k_0}^{k-1} A(j)x(k_0), \quad k > k_0
 \end{aligned}$$

This result can be shown formally by induction. The solution of (2) is then

$$x(k) = \Phi(k, k_0)x(k_0), \quad (3)$$

where $\Phi(k, k_0)$ is the *state transition matrix* of (2) given by

$$\Phi(k, k_0) = \prod_{j=k_0}^{k-1} A(j), \quad k > k_0; \quad \Phi(k_0, k_0) = I. \quad (4)$$

Note that in the time-invariant case, $\Phi(k, k_0) = A^{k-k_0}$.

System Response

Consider now the state equation in (1). It can be easily shown that the solution is

$$\begin{aligned}
 x(k) &= \Phi(k, k_0)x(k_0) \\
 &+ \sum_{j=k_0}^{k-1} \Phi(k, j+1)B(j)u(j), \quad k > k_0,
 \end{aligned} \quad (5)$$

and the response $y(k)$ of (1) is

$$\begin{aligned}
 y(k) &= C(k)\Phi(k, k_0)x(k_0) \\
 &+ C(k) \sum_{j=k_0}^{k-1} \Phi(k, j+1)B(j)u(j) \\
 &+ D(k)u(k), \quad k > k_0,
 \end{aligned} \quad (6)$$

and

$$y(k_0) = C(k_0)x(k_0) + D(k_0)u(k_0).$$

Equation (5) is the sum of two parts, the *state response* (when $u(k) = 0$ and the system is driven only by the initial state conditions) and the *input response* (when $x(k_0) = 0$ and the system is driven only by the input $u(k)$); this illustrates the linear systems principle of superposition.

Equivalence of State Variable Descriptions

Given (1), consider the new state vector $\tilde{x}$ where

$$\tilde{x}(k) = P(k)x(k)$$

where $P^{-1}(k)$ exists. Then

$$\begin{aligned}
 \tilde{x}(k+1) &= \tilde{A}(k)\tilde{x}(k) + \tilde{B}(k)u(k) \\
 y(k) &= \tilde{C}(k)\tilde{x}(k) + \tilde{D}(k)u(k)
 \end{aligned} \quad (7)$$

where

$$\begin{aligned}
 \tilde{A}(k) &= P(k+1)A(k)P^{-1}(k), \\
 \tilde{B}(k) &= P(k+1)B(k), \\
 \tilde{C}(k) &= C(k)P^{-1}(k), \\
 \tilde{D}(k) &= D(k)
 \end{aligned}$$

is equivalent to (1). It can be easily shown that equivalent descriptions give rise to the same discrete impulse responses.

Summary

State variable descriptions for linear discrete-time time-varying systems were introduced and the state and output responses to inputs and initial conditions were derived. The equivalence of state variable representations was also discussed.

Cross-References

- [Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions](#)
- [Linear Systems: Discrete-Time Impulse Response Descriptions](#)

- [Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions](#)
- [Sampled-Data Systems](#)

Recommended Reading

The state variable descriptions received wide acceptance in systems theory beginning in the late 1950s. This was primarily due to the work of R.E. Kalman. For historical comments and extensive references, see Kailath (1980). The use of state variable descriptions in systems and control opened the way for the systematic study of systems with multi-inputs and multi-outputs.

Bibliography

- Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston
- Astrom KJ, Wittenmark B (1997) Computer controlled systems: Theory and Design, 3rd edn. Prentice Hall, Upper Saddle River, NJ
- Franklin GF, Powell DJ, Workman ML (1998) Digital control of dynamic systems, 3rd edn. Addison-Wesley, Menlo Park, CA
- Jury EI (1958) Sampled-data control systems. Wiley, New York
- Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs
- Ragazzini JR, Franklin GF (1958) Sampled-data control systems. McGraw-Hill, New York
- Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs

LMI Approach to Robust Control

Kang-Zhi Liu

Department of Electrical and Electronic Engineering, Chiba University, Chiba, Japan

Abstract

In the analysis and design of robust control systems, LMI method plays a fundamental role. This article gives a brief introduction to

this topic. After the introduction of LMI, it is illustrated how a control design problem is related with matrix inequality. Then, two methods are explained on how to transform a control problem characterized by matrix inequalities to LMIs, which is the core of the LMI approach. Based on this knowledge, the LMI solutions to various kinds of robust control problems are illustrated. Included are $\mathcal{H}_\infty$ and $\mathcal{H}_2$ control, regional pole placement, and gain-scheduled control.

Keywords

Gain-scheduled control · $\mathcal{H}_\infty$ and $\mathcal{H}_2$ control · LMI · Multi-objective control · Regional pole placement · Robust control

Introduction of LMI

A matrix inequality in a form of

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0 \quad (1)$$

is called an LMI (linear matrix inequality). Here, $x = [x_1 \cdots x_m]$ is the unknown vector and $F_i (i = 1, \dots, m)$ is a symmetric matrix. $F(x)$ is an affine function of x . The inequality means that $F(x)$ is positive definite.

LMI can be solved effectively by numerical algorithms such as the famous interior point method (Nesterov and Nemirovskii 1994). MATLAB has an LMI toolbox (Gahinet et al. 1995) tailored for solving the related control problems. Boyd et al. (1994) provide detailed theoretic fundamentals of LMI. A comprehensive and up-to-date treatment on the applications of LMI in robust control is covered in Liu and Yao (2014).

The notation $\text{He}(A) = A + A^T$ is used to simplify the presentation of large matrices; $A_\perp$ is a matrix whose columns form the basis of the kernel space of A , i.e., $AA_\perp = 0$. Further, $A \otimes B$ denotes the Kronecker product of matrices (A, B) .

Control Problems and LMI

In control problems, it is often the case that the variables are matrices. For example, the necessary and sufficient condition for the stability of a linear system $\dot{x}(t) = Ax(t)$ is that there exists a positive-definite matrix P satisfying the inequality $AP + PA^T < 0$. Although this is different from the LMI of Eq. (1) in form, it can be converted to Eq. (1) equivalently by using a basis of symmetric matrices.

Next, consider the stabilization of system $\dot{x} = Ax + Bu$ by a state feedback $u = Fx$. The closed-loop system is $\dot{x} = (A + BF)x$. Therefore, the stability condition is that there exist a positive-definite matrix P and a feedback gain matrix F satisfying the inequality

$$(A + BF)P + P(A + BF)^T < 0. \quad (2)$$

In this inequality, FP , the product of unknown variables F and P , appears. Such matrix inequality is called a *bilinear matrix inequality*, or *BMI* for short. BMI problem is non-convex and difficult to solve. There are mainly two methods for transforming a BMI into an LMI: variable elimination and variable change.

From BMI to LMI: Variable Elimination

The method of variable elimination is good at optimizing single-objective problems. This method is based on the theorem below (Gahinet and Apkarian 1994).

Lemma 1 *Given real matrices E, F, G with G being symmetric, the inequality*

$$E^T XF + F^T X^T E + G < 0 \quad (3)$$

has a solution X if and only if the following two inequalities hold simultaneously

$$E_{\perp}^T G E_{\perp} < 0, \quad F_{\perp}^T G F_{\perp} < 0. \quad (4)$$

Application of this theorem to the previous stabilization problem (2) yields $(B^T)_{\perp}^T (AP + PA^T)(B^T)_{\perp} < 0$, which is an LMI about P . Once P is obtained, it may be substituted back into the inequality (2) and solve for F .

For output feedback problems, it is often needed to construct a new matrix from two given matrices in solving a control problem with LMI approach. The method is given by the following lemma.

Lemma 2 *Given two n -dimensional positive-definite matrices X and Y , a $2n$ -dimensional positive-definite matrix $\mathbb{P}$ satisfying the conditions*

$$\mathbb{P} = \begin{bmatrix} Y & * \\ * & * \end{bmatrix}, \quad \mathbb{P}^{-1} = \begin{bmatrix} X & * \\ * & * \end{bmatrix}$$

can be constructed if and only if

$$\begin{bmatrix} X & I \\ I & Y \end{bmatrix} > 0. \quad (5)$$

Factorizing $Y - X^{-1}$ as FF^T , a solution is given by

$$\mathbb{P} = \begin{bmatrix} Y & F \\ F^T & I \end{bmatrix}.$$

As an example of output feedback control design, let us consider the stabilization of the plant

$$\dot{x}_p = Ax_p + Bu, \quad y = Cx_p \quad (6)$$

with a full-order dynamic controller

$$\dot{x}_K = A_K x_K + B_K y, \quad u = C_K x_K + D_K y. \quad (7)$$

The closed-loop system is

$$\begin{bmatrix} \dot{x}_p \\ \dot{x}_K \end{bmatrix} = A_c \begin{bmatrix} x_p \\ x_K \end{bmatrix}, \quad A_c = \begin{bmatrix} A + BD_K C & BC_K \\ B_K C & A_K \end{bmatrix}. \quad (8)$$

The stability condition is that the matrix inequality

$$A_c^T \mathbb{P} + \mathbb{P} A_c < 0 \quad (9)$$

has a solution $\mathbb{P} > 0$. To apply the variable elimination method, we need to put all coefficient matrices of the controller into a single matrix.

This is done as follows:

$$A_c = \bar{A} + \bar{B}\mathcal{K}\bar{C}, \quad \mathcal{K} = \begin{bmatrix} D_K & C_K \\ B_K & A_K \end{bmatrix} \quad (10)$$

in which $\bar{A} = \text{diag}(A, 0)$, $\bar{B} = \text{diag}(B, I)$, and $\bar{C} = \text{diag}(C, I)$, all being block diagonal. Then, based on Lemma 1, the stability condition reduces to the existence of symmetric matrices X , Y satisfying LMIs

$$(B^T)_{\perp}^T (AX + XA^T) (B^T)_{\perp} < 0 \quad (11)$$

$$(C_{\perp})^T (YA + A^T Y) C_{\perp} < 0. \quad (12)$$

Meanwhile, the positive definiteness of matrix $\mathbb{P}$ is guaranteed by Eq. (5) in Lemma 2.

From BMI to LMI: Variable Change

We may also use the method of variable change to transform a BMI into an LMI. This method is good at multi-objective optimization.

The detail is as follows (Gahinet 1996). A positive-definite matrix can always be factorized as the quotient of two triangular matrices, i.e.,

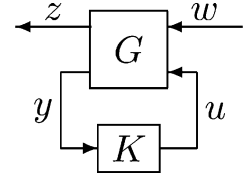
$$\mathbb{P}\Pi_1 = \Pi_2, \quad \Pi_1 = \begin{bmatrix} X & I \\ M^T & 0 \end{bmatrix}, \quad \Pi_2 = \begin{bmatrix} I & Y \\ 0 & N^T \end{bmatrix}. \quad (13)$$

$\mathbb{P} > 0$ is guaranteed by Eq. (5) for a full-order controller. Further, the matrices M , N are computed from $MN^T = I - XY$. Consequently, they are nonsingular.

An equivalent inequality $\Pi_1^T A_c^T \Pi_2 + \Pi_2^T A_c \Pi_1 < 0$ is obtained by multiplying Eq. (9) with Π_1^T and Π_1 . After a change of variables, this inequality reduces to an LMI

$$\text{He} \begin{bmatrix} AX + BC & A + B\mathbb{D}C + \mathbb{A}^T \\ 0 & YA + \mathbb{B}C \end{bmatrix} < 0. \quad (14)$$

LMI Approach to Robust Control, Fig. 1
Generalized feedback system



The new variables $\mathbb{A}$, $\mathbb{B}$, $\mathbb{C}$, $\mathbb{D}$ are set as

$$\begin{aligned} \mathbb{A} &= NA_K M^T + NB_K CX + YBC_K M^T \\ &\quad + Y(A + BD_K C)X \\ \mathbb{B} &= NB_K + YBD_K, \quad \mathbb{C} = C_K M^T \\ &\quad + D_K CX, \quad \mathbb{D} = D_K. \end{aligned} \quad (15)$$

The coefficient matrices of the controller become

$$\begin{aligned} D_K &= \mathbb{D}, \quad C_K = (\mathbb{C} - D_K CX)M^{-T}, \\ B_K &= N^{-1}(\mathbb{B} - YBD_K) \\ A_K &= N^{-1}(\mathbb{A} - NB_K CX - YBC_K M^T \\ &\quad - Y(A + BD_K C)X)M^{-T}. \end{aligned} \quad (16)$$

$\mathcal{H}_2$ and $\mathcal{H}_\infty$ Control

In system optimization, $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms are the most popular and effective performance indices. $\mathcal{H}_2$ norm of a transfer function is closely related with the squared area of its impulse response. So, a smaller $\mathcal{H}_2$ norm implies a faster response. Meanwhile, $\mathcal{H}_\infty$ norm of a transfer function is the largest magnitude of its frequency response. Hence, for a transfer function from the disturbance to the controlled output, a smaller $\mathcal{H}_\infty$ norm guarantees a better disturbance attenuation.

Usually $\mathcal{H}_2$ and $\mathcal{H}_\infty$ optimization problems are treated in the generalized feedback system of Fig. 1. Here, the generalized plant $G(s)$ includes the nominal plant, the performance index, and the weighting functions.

Let the generalized plant $G(s)$ be

$$G(s) = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} (sI - A)^{-1} \begin{bmatrix} B_1 & B_2 \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & 0 \end{bmatrix}. \quad (17)$$

Further, the stabilizability of (A, B_2) and the detectability of (C_2, A) are assumed. The closed-loop transfer matrix from the disturbance

w to the performance output z is denoted by

$$H_{zw}(s) = C_c(sI - A_c)^{-1}B_c + D_c. \quad (18)$$

The condition for $H_{zw}(s)$ to have an $\mathcal{H}_2$ norm less than γ , i.e., $\|H_{zw}\|_2 < \gamma$, is that there are symmetric matrices $\mathbb{P}$ and W satisfying

$$\begin{bmatrix} \mathbb{P}A_c + A_c^T\mathbb{P} & C_c^T \\ C_c & -I \end{bmatrix} < 0, \quad \begin{bmatrix} W & B_c^T\mathbb{P} \\ \mathbb{P}B_c & \mathbb{P} \end{bmatrix} > 0 \quad (19)$$

as well as $\text{Tr}(W) < \gamma^2$. Here, $\text{Tr}(W)$ denotes the trace of matrix W , i.e., the sum of its diagonal entries.

The LMI solution is derived via the application of the variable change method, as given below.

Theorem 1 Suppose that $D_{11} = 0$. The $\mathcal{H}_2$ control problem is solvable if and only if there exist symmetric matrices X, Y, W and matrices $\mathbb{A}, \mathbb{B}, \mathbb{C}$ satisfying the following LMIs:

$$\text{He} \begin{bmatrix} AX + B_2\mathbb{C} & 0 & 0 \\ A^T + \mathbb{A} & YA + \mathbb{B}C_2 & 0 \\ C_1X + D_{12}\mathbb{C} & C_1 & -\frac{1}{2}I \end{bmatrix} < 0 \quad (20)$$

$$\begin{bmatrix} W & B_1^T & B_1^TY \\ B_1 & X & I \\ YB_1 & I & Y \end{bmatrix} > 0, \quad \text{Tr}(W) < \gamma^2. \quad (21)$$

When the LMI Eqs. (20) and (21) have solutions, an $\mathcal{H}_2$ controller is given by Eq. (16) by setting $\mathbb{D} = 0$.

The $\mathcal{H}_\infty$ control problem is to design a controller so that $\|H_{zw}\|_\infty < \gamma$. The starting point of $\mathcal{H}_\infty$ control is the famous bounded real lemma, which states that $H_{zw}(s)$ has an $\mathcal{H}_\infty$ normless

than γ if and only if there is a positive-definite matrix $\mathbb{P}$ satisfying

$$\begin{bmatrix} A_c^T\mathbb{P} + \mathbb{P}A_c & \mathbb{P}B_c & C_c^T \\ B_c^T\mathbb{P} & -\gamma I & D_c^T \\ C_c & D_c & -\gamma I \end{bmatrix} < 0. \quad (22)$$

There are two kinds of LMI solutions to this control problem: one based on variable elimination and one based on variable change.

To state the first solution, define the following matrices first:

$$N_Y = [C_2 \ D_{21}]_\perp, \quad N_X = [B_2^T \ D_{12}^T]_\perp. \quad (23)$$

Theorem 2 The $\mathcal{H}_\infty$ control problem has a solution if and only if Eq. (5) and the following LMIs have positive-definite solutions X, Y :

$$\begin{bmatrix} N_X^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} AX + XA^T & XC_1^T & B_1 \\ C_1X & -\gamma I & D_{11} \\ B_1^T & D_{11}^T & -\gamma I \end{bmatrix} \times \begin{bmatrix} N_X & 0 \\ 0 & I \end{bmatrix} < 0 \quad (24)$$

$$\begin{bmatrix} N_Y^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} YA + A^TY & YB_1 & C_1^T \\ B_1^TY & -\gamma I & D_{11}^T \\ C_1 & D_{11} & -\gamma I \end{bmatrix} \times \begin{bmatrix} N_Y & 0 \\ 0 & I \end{bmatrix} < 0. \quad (25)$$

Once a matrix $\mathbb{P}$ is computed according to Lemma 2, Eq. (22) becomes an LMI and its solution yields the controller.

The second solution is given below.

Theorem 3 The $\mathcal{H}_\infty$ control problem has a solution if and only if Eq. (5) and the following LMI have solutions X, Y and $\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}$:

$$\text{He} \begin{bmatrix} AX + B_2\mathbb{C} & A + B_2\mathbb{D}C_2 & B_1 + B_2\mathbb{D}D_{21} & 0 \\ \mathbb{A} & YA + \mathbb{B}C_2 & YB_1 + \mathbb{B}D_{21} & 0 \\ 0 & 0 & -\frac{\gamma}{2}I & 0 \\ C_1X + D_{12}\mathbb{C} & C_1 + D_{12}\mathbb{D}C_2 & D_{11} + D_{12}\mathbb{D}D_{21} & -\frac{\gamma}{2}I \end{bmatrix} < 0. \quad (26)$$

The controller is given by Eq. (16).

Regional Pole Placement

The location of system poles determines the response quality. However, for uncertain systems it is impossible to place the closed-loop poles at fixed points because they move with the variation of the plant. Nevertheless, it is still possible to place the closed-loop poles inside a region. For convex regions characterized by LMI, the design method is mature and proven effective in practice.

Let us see how to characterize a convex region. It is easy to know that a complex number z is inside the disk of Fig. 2a if and only if it satisfies

$$\begin{bmatrix} -r & z + c \\ \bar{z} + c & -r \end{bmatrix} < 0.$$

Similarly, z is inside the sector of Fig. 2b if and only if

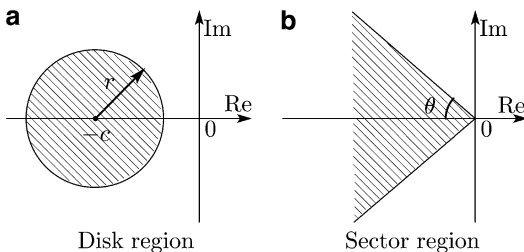
$$\begin{bmatrix} (z + \bar{z}) \sin \theta & (z - \bar{z}) \cos \theta \\ -(z - \bar{z}) \cos \theta & (z + \bar{z}) \sin \theta \end{bmatrix} < 0.$$

Generally, the set of complex number z characterized by

$$D = \{z \in \mathbb{C} | L + zM + \bar{z}M^T < 0\} \quad (27)$$

is called an LMI region, in which L is a symmetric matrix. For the dynamic system

$$\dot{x} = Ax, \quad (28)$$



LMI Approach to Robust Control, Fig. 2 Typical examples of LMI region

all of its poles are in the LMI region D if and only if there is a positive-definite matrix $\mathbb{P}$ satisfying the LMI

$$L \otimes \mathbb{P} + M \otimes (A\mathbb{P}) + M^T \otimes (A\mathbb{P})^T < 0. \quad (29)$$

This forms the basis for the regional pole placement design.

For the disk region in Fig. 2a, the condition becomes

$$\begin{bmatrix} -r\mathbb{P} & c\mathbb{P} + A\mathbb{P} \\ c\mathbb{P} + (A\mathbb{P})^T & -r\mathbb{P} \end{bmatrix} < 0. \quad (30)$$

Meanwhile, for the sector region in Fig. 2b, the corresponding LMI is

$$\begin{bmatrix} (A\mathbb{P} + \mathbb{P}A^T) \sin \theta & (A\mathbb{P} - \mathbb{P}A^T) \cos \theta \\ -(A\mathbb{P} - \mathbb{P}A^T) \cos \theta & (A\mathbb{P} + \mathbb{P}A^T) \sin \theta \end{bmatrix} < 0. \quad (31)$$

Moreover, for a composite LMI region, such as the intersection of the disk and the sector, the pole placement is guaranteed by enforcing a common solution $\mathbb{P}$ to all the corresponding LMIs.

In the pole placement design, only the variable change method is applicable. For example, in the nominal closed-loop system Eq. (8), the pole placement condition is that the LMI

$$L \otimes \begin{bmatrix} X & I \\ I & Y \end{bmatrix} + \text{He} \left(M \otimes \begin{bmatrix} AX + BC & A + B\mathbb{D}C \\ \mathbb{A} & Y A + \mathbb{B}C \end{bmatrix} \right) < 0 \quad (32)$$

and Eq. (5) are solvable (Chilali and Gahinet 1996).

For systems with norm-bounded parameter uncertainty, a robust pole placement method is provided in Chilali et al. (1999).

Multi-objective Control

It is noted that all of the preceding control designs involve a positive-definite matrix $\mathbb{P}$. Therefore, a multi-objective control design is easily realized

by enforcing a common solution $\mathbb{P}$ to the corresponding matrix inequality conditions.

Gain-Scheduled Control

In practice, many nonlinear systems can be expressed as linear systems with state-dependent coefficients in form, which is known as the LPV (linear parameter-varying) form. For example, in the model of a robot arm $J\ddot{\theta} + mgl \sin \theta = u$, if we define a parameter as $p(t) = \sin \theta / \theta$, then it can be written as an LPV model $J\ddot{\theta}(t) + mglp(t)\theta(t) = u(t)$. In this class of systems, when the parameter $p(t)$ is available online and its range is finite, one may tune the controller parameters based on the information of $p(t)$, so as to achieve a higher performance. This is referred to as gain-scheduled control.

Consider the following affine model:

$$\dot{x} = A(p(t))x + B_1(p(t))d + B_2(p(t))u \quad (33)$$

$$z = C_1(p(t))x + D_{11}d + D_{12}u \quad (34)$$

$$y = C_2(p(t))x + D_{21}d \quad (35)$$

where $A(p) \sim C_2(p)$ are affine functions of the time-varying parameter vector $p(t)$, such as $A(p) = A_0 + \sum_{i=1}^q p_i(t)A_i$. The gain-scheduled control is to impose, on the coefficient matrices of the controller, the same affine structure about $p(t)$ such as $A_K(p) = A_{K0} + \sum_{i=1}^q p_i(t)A_{Ki}$.

To simplify the design, it is desirable that the coefficient matrices of the closed-loop system become affine functions of the parameter vector $p(t)$. This may be satisfied by restraining some of the matrices of the controller to constant ones. The easy-to-design structure of a gain-scheduled controller is summarized as follows:

- Both $B_2(p)$ and $C_2(p)$ depend on $p(t)$: (B_K, C_K) must be constant matrices besides $D_K = 0$.
- Constant (B_2, C_2) : All coefficient matrices of the controller can be affine functions of the parameter vector $p(t)$.

- Constant B_2 : (B_K, D_K) must be constant matrices.
- Constant C_2 : (C_K, D_K) must be constant matrices.

When the structure of the gain-scheduled controller is chosen as summarized above, the solvability conditions reduce to those at all vertices θ_i of the scheduling parameter vector $p(t)$. Further, a multi-objective is achieved by imposing a common solution $\mathbb{P}$ to all LMI conditions. Some concrete examples are illustrated below:

$\mathcal{H}_\infty$ Norm Spec: The conditions of Theorem 3 are satisfied at all vertices θ_j of the parameter vector $p(t)$.

$\mathcal{H}_2$ Norm Spec: The conditions of Theorem 1 are satisfied at all vertices θ_j of the parameter vector $p(t)$.

Regional Pole Placement: Eq. (32) is satisfied at all vertices θ_j of the parameter vector $p(t)$ and Eq. (5) holds.

Moreover, a different gain-scheduled method is proposed in Packard (1994) for parametric systems with norm-bounded uncertainty.

Summary and Future Direction

LMI approach is a very powerful method that can be applied to solve most of the robust control problems smartly and effectively. In particular, its capability of handling the multi-objective control problems is very attractive and proven useful in industrial applications.

Further study is needed in the following directions.

- New method of variable change is desired in order to deal with the robust performance design of parametric systems.
- Almost all robust performance designs are carried out based on sufficient conditions. It is very important to discover less conservative design methods.

Cross-References

- ▶ [H-Infinity Control](#)
- ▶ [Linear Matrix Inequality Techniques in Optimal Control](#)
- ▶ [Optimization-Based Robust Control](#)
- ▶ [Optimization-Based Control Design Techniques and Tools](#)
- ▶ [Robust Synthesis and Robustness Analysis Techniques and Tools](#)

Bibliography

- Apkarian P, Gahinet P (1995) A convex characterization of gain-scheduled $\mathcal{H}_\infty$ controllers. *IEEE Trans Autom Control* 40(5):853–864
- Boyd SP et al (1994) *Linear matrix inequalities in system and control*. SIAM, Philadelphia
- Chilali M, Gahinet P (1996) H_∞ design with pole placement constraints: an LMI approach. *IEEE Trans Autom Control* 41(3):358–367
- Chilali M, Gahinet P, Apkarian P (1999) Robust pole placement in LMI regions. *IEEE Trans Autom Control* 44(12):2257–2270
- Gahinet P (1996) Explicit controller formulas for LMI-based $\mathcal{H}_\infty$ synthesis. *Automatica* 32(7):1007–1014
- Gahinet P, Apkarian P (1994) A linear matrix inequality approach to $\mathcal{H}_\infty$ control. *Int J Robust Nonlinear Control* 4:421–448
- Gahinet P, Nemirovski A, Laub AJ, Chilali M (1995) *LMI control toolbox*. The MathWorks, Inc., Natick
- Liu KZ, Yao Y (2014, to appear) *Robust control: theory and applications*. Wiley, New York
- Nesterov Y, Nemirovskii A (1994) *Interior-point polynomial methods in convex programming*. SIAM, Philadelphia
- Packard A (1994) Gain scheduling via linear fractional transformations. *Syst Control Lett* 22: 79–92

Low-Power High-Gain Observers

Daniele Astolfi¹ and Lorenzo Marconi²
¹CNRS, LAGEPP UMR 5007, Villeurbanne, France
²C.A.SY. – DEI, University of Bologna, Bologna, Italy

Abstract

This entry deals with a class of nonlinear high-gain observers referred to as “low-power” due

to the fact that the high-gain parameter is powered only up to 2 regardless of the dimension of the observed system, on the contrary of conventional high-gain observers in which the high-gain parameter is powered up to the dimension of the observed system. This kind of observers preserves the main properties of conventional high-gain observers in terms of tunability of the speed of convergence and robustness to exogenous disturbances, by outperforming those in terms of numerical implementation and sensitivity to high-frequency measurement noise. The drawback of low-power high-gain observers over the conventional ones is given by the dimension of its dynamics that is larger.

Keywords

Nonlinear Observers · High-gain observers · Noise Sensitivity

Introduction

Observing the state of dynamical systems by processing available measurements is a problem ubiquitous in control. Output feedback stabilization (Atassi and Khalil 1999; Teel and Praly 1995), output regulation (Isidori 2017), and fault detection and isolation (Edwards et al. 2000) are only a few examples of research areas in which state observers play a key role. The problem of designing an observer becomes very challenging when the underlying dynamics are nonlinear and possibly affected by uncertainties, and a robust, possibly practical, estimation of the state is needed. In this regard a special role in the literature of robust nonlinear observers is played by the so-called high-gain observers (Khalil and Praly 2014). The adjective high-gain places emphasis on a feature of the observer whose structure distinguishes for a design parameter, referred to as the high-gain parameter, whose value is typically taken large to dominate the effect of uncertainties and to tune the speed of convergence of the estimation and the residual bound of the error that can be arbitrarily fixed.

A pretty common framework where high-gain observers are studied is the one of systems ful-

filling a *uniform observability assumption* (see Definition 1.2 in Gauthier and Kupka (2001)). This class of systems is described, possibly after a coordinates change, in the so-called canonical observability form (see Theorem 4.1 in Gauthier and Kupka (2001))

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\vdots \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= \varphi(x) \\ y &= x_1\end{aligned}\quad (1)$$

in which $x = \text{col}(x_1, \dots, x_n) \in \mathbb{R}^n$ is the state and $\varphi(\cdot)$ is nonlinear function. We observe that $x = \text{col}(y, \dot{y}, \dots, y^{(n-1)})$, namely, in this coordinate frame the state coincides with the output and its first $n-1$ time derivatives. The state is typically assumed to satisfy $x \in X$, in which X is a compact set of $\mathbb{R}^n$, whereas the function $\varphi(\cdot)$ is assumed to be *uniformly locally Lipschitz* in X , namely, there exists a constant $\bar{\varphi} > 0$ such that

$$\|\varphi(x') - \varphi(x'')\| \leq \bar{\varphi} \|x' - x''\| \quad \forall x', x'' \in X.$$

In this framework, conventional high-gain observers take the form

$$\begin{aligned}\dot{\hat{x}}_1 &= \hat{x}_2 + \ell c_1 (y - \hat{x}_1) \\ \dot{\hat{x}}_2 &= \hat{x}_3 + \ell^2 c_2 (y - \hat{x}_1) \\ &\vdots \\ \dot{\hat{x}}_{n-1} &= \hat{x}_n + \ell^{n-1} c_{n-1} (y - \hat{x}_1) \\ \dot{\hat{x}}_n &= \varphi_s(\hat{x}) + \ell^n c_n (y - \hat{x}_1)\end{aligned}\quad (2)$$

in which the c_i 's are chosen so that the polynomial $\lambda^n + c_n \lambda^{n-1} + \dots + c_2 \lambda + c_1$ is Hurwitz, $\varphi_s(\cdot)$ is a bounded function that agrees with $\varphi(\cdot)$ in X , and ℓ is the high-gain design parameter. It turns out (see Khalil and Praly (2014)) that there exists a $\ell^*(\bar{\varphi}) > 0$ such that for all $\ell \geq \ell^*$ the following estimate holds true

$$\|\hat{x}(t) - x(t)\| \leq \alpha \ell^{n-1} \exp(-\beta \ell t) \quad (3)$$

for all $\hat{x}(0) \in X$ and all the system trajectories $x(t) \in X$, in which (α, β) are positive constants (not dependent on ℓ). The previous bound shows that asymptotic estimation is achieved with an arbitrary fast convergence rate tuneable through the design high-gain parameter ℓ . The arbitrary fast convergence rate, though, is obtained at the expense of a “peaking phenomenon” leading to large estimation errors (proportional to ℓ^{n-1}) in the initial part of the transient. A clear drawback of this kind of observers when ℓ is picked large is also at implementation level due to the fact that numerical value of the coefficients scales up to ℓ^n .

High-gain observers can be shown also to be robust to disturbances entering anywhere in the chain of integrators of (1) and also affecting the measured output y . In particular, by adopting the observer (1) to an observed system of the form

$$\begin{aligned}\dot{x}_1 &= x_2 + d_1 \\ \dot{x}_2 &= x_3 + d_2 \\ &\vdots \\ \dot{x}_{n-1} &= x_n + d_{n-1} \\ \dot{x}_n &= \varphi(x) + d_n \\ y &= x_1 + v\end{aligned}\quad (4)$$

in which $d := \text{col}(d_1, \dots, d_n) \in D_d \subset \mathbb{R}^n$ and $v \in D_v \subset \mathbb{R}$ are bounded disturbances (with D_d and D_v compact sets of $\mathbb{R}^n$ and $\mathbb{R}$), it turns out that for a sufficiently large value of ℓ , the following estimate holds (see Astolfi and Marconi (2015))

$$\|\hat{x}(t) - x(t)\| \leq \max \left\{ \alpha \ell^{n-1} \exp(-\beta \ell t), \frac{\gamma}{\ell} \|\Gamma(\ell) d\|_\infty, \ell^{n-1} \theta \|v\|_\infty \right\} \quad (5)$$

for some positive constants $\alpha, \beta, \gamma, \theta$ (independent of ℓ) and with $\Gamma(\ell) := \text{diag}(\ell^{n-1}, \dots, \ell, 1)$. The form of $\Gamma(\ell)$, in particular, shows how the high-gain parameter affects the asymptotic gain of the disturbances on the estimation errors. As for disturbances entering in the first $n-1$ integrators, it turns out that ℓ affects the asymptotic gain in an unfavorable way, namely, the larger is the high-gain parameter, the larger is the effect of the disturbance d

on the asymptotic estimation error (with the exception of d_{n-1} in which the asymptotic gain is not affected by ℓ). As for the disturbance d_n , on the other hand, (5) shows that practical observation is achieved with an estimation error that can be steered to arbitrarily small values in arbitrarily small time by increasing the values of ℓ . Similarly, the measurements noise is amplified on the asymptotic estimation error of a quantity proportional to ℓ^{n-1} . The latter property, in turn, practically limits the convergence rate achievable from this kind of observers in noisy environments.

The Low-Power Observer Structure

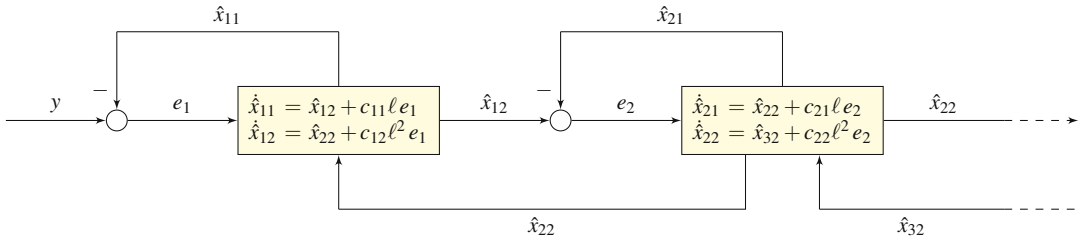
The same asymptotic properties (5) guaranteed by the high-gain observer (2) could be obtained by a different observer structure, referred to as “low power,” in which the high-gain parameter is powered just up to the order two regardless of the dimension n of the observed system. The “low-power” high-gain observer is given by properly interconnecting $n - 1$ subsystems, each of dimension 2 and with state $(\hat{x}_{i1}, \hat{x}_{i2})$, $i = 1, \dots, n - 1$, according to the following structure proposed in Astolfi and Marconi (2015)

$$\begin{aligned}
 \dot{\hat{x}}_{11} &= \hat{x}_{12} + \ell c_{11}(y - \hat{x}_{11}) \\
 \dot{\hat{x}}_{12} &= \hat{x}_{22} + \ell^2 c_{12}(y - \hat{x}_{11}) \\
 &\vdots \\
 \dot{\hat{x}}_{i1} &= \hat{x}_{i2} + \ell c_{i1}(\hat{x}_{(i-1)2} - \hat{x}_{i1}) \\
 \dot{\hat{x}}_{i2} &= \hat{x}_{(i+1)2} + \ell^2 c_{i2}(\hat{x}_{(i-1)2} - \hat{x}_{i1}) \quad i = 2, \dots, n - 2 \\
 &\vdots \\
 \dot{\hat{x}}_{(n-1)1} &= \hat{x}_{(n-1)2} + \ell c_{(n-1)1}(\hat{x}_{(n-2)2} - \hat{x}_{(n-1)1}) \\
 \dot{\hat{x}}_{(n-1)2} &= \varphi_s(\hat{x}) + \ell^2 c_{(n-1)2}(\hat{x}_{(n-2)2} - \hat{x}_{(n-1)1})
 \end{aligned} \tag{6}$$

in which $\hat{x} = \text{col}(\hat{x}_{11}, \hat{x}_{21}, \dots, \hat{x}_{(n-1)1}, \hat{x}_{(n-2)2}) \in \mathbb{R}^n$, the $c_{i,j}$, $i = 1, 2$, $j = 1, \dots, n - 1$, are design coefficients, ℓ is the high-gain design parameter and $\varphi_s(\hat{x})$ is defined as in the canonical high-gain observer. A graphical sketch of the observer structure is shown in Fig. 1.

The intuition behind the observer structure is along the following lines. The first subsystem, with state $(\hat{x}_{11}, \hat{x}_{12})$, resembles the structure of a canonical high-gain observer aiming to estimate $(y, \dot{y})$. In this respect the term $\hat{x}_{22}$ acts as “consistency term” whose asymptotic value is expected to converge to $\ddot{y}$. This consistency term is provided by the state of the second subsystem, with state $(\hat{x}_{21}, \hat{x}_{22})$ still relying on a canonical high-gain structure and aiming to provide an estimation of $(\dot{y}, \ddot{y})$. The innovation term of this system, given by $\hat{x}_{12} - \hat{x}_{21}$, is in fact constructed by using the state variable $\hat{x}_{12}$ of the “previous” subsystem, acting as a “proxy” of $\dot{y}$, and the estimation of $\dot{y}$ provided by the state $\hat{x}_{21}$ of

the “actual” subsystem. Similar to the above, the consistency term is given by $\hat{x}_{32}$ provided by the “next” subsystem and expected to provide an estimation of $y^{(3)}$. This intuition continues all along the chain of the $(n - 1)$ subsystems of (6), with the state $(\hat{x}_{i1}, \hat{x}_{i2})$ of the generic i -th subsystem providing an estimation of $(y^{(i-1)}, y^{(i)})$ and with the innovation and the consistency terms constructed by using the state, respectively, of the “previous” and “next” subsystem. The fact that each subsystem has dimension 2 allows one to limit the power of the high-gain parameter to 2 regardless of the value of n , with the drawback of increasing the dimension of the overall observer to the dimension $2n - 2$. As the previous analysis reveals, a potential benefit of this redundancy is that the low-power observer provides a “double” estimate of the state $(y, \dot{y}, \dots, y^{(n-1)})$ of system (1). As a matter of fact, each variable $y^{(i)}$ is estimated twice, for $i = 1, \dots, n - 2$, by $\hat{x}_{i2}$ and $\hat{x}_{(i+1)2}$, with y and $y^{(n-1)}$ estimated



Low-Power High-Gain Observers, Fig. 1 Block diagram representation of the low-power high-gain observer (6)

once, respectively, by the state variables $\hat{x}_{11}$ and $\hat{x}_{(n-2)2}$. This fact justifies the introduction of a new vector

$$\hat{x}' := \text{col}(\hat{x}_{11}, \hat{x}_{12}, \hat{x}_{22}, \dots, \hat{x}_{(n-1)2}) \in \mathbb{R}^n$$

that, besides $\hat{x}$, represents an additional estimation of the state x .

Of course, the envisioned asymptotic properties of (6) rely on an appropriate choice of the coefficients $c_{i,j}$. With respect to canonical high-gain observers, this choice is slightly more complicated. In particular, define the following set of matrices

$$A_2 := \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},$$

$$C_2 := (1, 0) \quad B_i := \text{col}(0, \dots, 0, 1) \in \mathbb{R}^i$$

$K_i := \text{col}(c_{i1}, c_{i2})$, and $E_i := A_2 - K_i C_2$, $i = 1, \dots, n-1$, and define recursively the matrices $M_i \in \mathbb{R}^{2i \times 2i}$, $i = 1, \dots, n-1$, as $M_1 := E_1$,

$$M_i := \begin{pmatrix} M_{i-1} & B_{2(i-1)} B_2^T \\ K_i B_{2(i-1)}^T & E_i \end{pmatrix},$$

$$i = 2, \dots, n-1$$

where $B_{2(i-1)}$ denotes the zero column vector of dimension $i-1$ with just a 1 in the last position. It turns out that the coefficients $c_{i,j}$ must be chosen such that the matrix M_{n-1} defined above is Hurwitz. As a matter of fact, in this case standard Lyapunov arguments (the same used also in the case of canonical high-gain observers) can be used to show that there exists a $\ell^*(\bar{\varphi})$ such that for all $\ell \geq \ell^*(\bar{\varphi})$ the state of the observer (6)

driven by the observed system (4) fulfills

$$\left\| \begin{pmatrix} \hat{x}(t) - x(t) \\ \hat{x}'(t) - x'(t) \end{pmatrix} \right\| \leq \max \left\{ \alpha \ell^{n-1} \exp(-\beta \ell t), \frac{\gamma}{\ell} \|\Gamma(\ell)d\|_\infty, \ell^{n-1} \theta \|v(t)\|_\infty \right\} \quad (7)$$

for some positive constants $(\alpha, \beta, \gamma, \theta)$ independent of ℓ and with $\Gamma(\ell)$ defined as before. Namely, the same nominal properties of the canonical high-gain observer, in terms of convergence speed and asymptotic properties in presence of disturbances, are preserved. As said, with respect to observer (2), the proposed structure (6) has larger dimension equal to $2n-2$, but the remarkable advantage of having the high-gain parameter ℓ powered just up to the order 2 regardless of the dimension n of the observed system.

We conclude the section by observing that a selection of the design parameters c_{ij} so that M_{n-1} is Hurwitz always exists. As a matter of fact, the Algorithm 1 presented next can be used (see Astolfi et al. 2016a).

It is worth remarking that the eigenvalues of M_{n-1} can be also arbitrarily assigned by following a more complex algorithm; see Lemma 1 in Astolfi and Marconi (2015).

Sensitivity to High-Frequency Noise

With reference to the sensitivity to measurement noise, it is worth noting that the bounds (5) and (7) refer to the so-called $\mathcal{L}_\infty$ gain, namely, characterizes the sensitivity to the class of *bounded* disturbances $v(t)$. It is a well-known

Algorithm 1 Computation of $c_{i,j}$ to obtain M_{n-1} Hurwitz

```

if  $i = 1$  then
    Select  $c_{11}, c_{12} > 0$ .
end if
for  $i = 2, \dots, n-1$  do
    Compute  $\gamma_i := \max_{\omega \in \mathbb{R}} \|G_i(j\omega)\|$  with  $G_i(s) := B_{2(i-1)}^\top (sI_{2(i-1)} - M_{i-1})^{-1} B_{2(i-1)}$ .
    Select  $c_{i1} := c_{(i-1)1}$ .
    Select any  $0 < c_{i2} < \frac{c_{i1}}{\gamma_i}$ .
end for

```

fact that measurement noises typically belong to a more restricted class of signals, namely, the one in which the harmonic content is confined at high frequency. In this respect, when the restricted class of *high-frequency* measurement noises is dealt with, the previous bound can be further refined by highlighting low-pass filtering properties of the observers (2) and (6), with the latter outperforming the former as detailed next. We consider, in particular, system (4) in which $d = 0$ and the measurement noise v is modelled as a quasi-periodic signal of the form

$$v(t) = \sum_{i=1}^{n_v} v_i^c \cos\left(\frac{\omega_i}{\varepsilon} t\right) + v_i^s \sin\left(\frac{\omega_i}{\varepsilon} t\right)$$

where n_v , v_i^c , v_i^s , ω_i are positive numbers and where $\varepsilon \in (0, 1)$ is a small number parametrizing the frequencies of the signal $v(t)$.

As for the canonical high-gain observer (2), it is possible to show (see Astolfi et al. 2016b) that the i -th estimation error fulfills the following bound

$$\limsup_{t \rightarrow \infty} \|\hat{x}_i(t) - x_i(t)\| \leq \varepsilon \mu \ell^i \|v\|_\infty$$

$$i = 1, \dots, n,$$

for all $0 < \varepsilon \leq \varepsilon^*(\ell)$, where $\varepsilon^*(\ell) > 0$ is positive constants depending on ℓ and μ is a positive constant independent of ℓ . This bound shows that once ℓ is fixed, the sensitivity of the estimation error to measurement noise decreases as ε takes smaller values, namely, as higher-frequency noise signals are considered, with an asymptotic gain proportional to ε . For linear

systems, this property immediately comes by frequency response arguments using the fact that the relative degree between the measurement noise v and the estimation error $\hat{x}_i - x_i$ for (4)–(2) is unitary for all $i = 1, \dots, n$.

Concerning the low-power high-gain observer (6), we observe that the relative degree between the “input” v and the i -th estimation error $\hat{x}_i - x_i$ is one for $i = 1$ (as for (2)) and then increases for higher values of i . More precisely, by defining $m := \lceil (n+1)/2 \rceil$ and considering a general case in which the function $\varphi(x)$ is affected by x_1 , the relative degree in question is i for $i = 1, \dots, m$ and $n - i + 2$ for $i = m+1, \dots, n$. As a consequence, it can be shown (see Astolfi et al. 2018) that the i -th estimation error associated with the low-power observer (6) fulfills the following bound

$$\limsup_{t \rightarrow \infty} \|\hat{x}_i(t) - x_i(t)\| \leq \varepsilon^i \mu \ell^{2i-1} \|v\|_\infty$$

for $i = 1, \dots, m$ and

$$\limsup_{t \rightarrow \infty} \|\hat{x}_i(t) - x_i(t)\| \leq \varepsilon^{n-i+2} \mu \ell \|v\|_\infty$$

for $i = m+1, \dots, n$, holding for $0 < \varepsilon \leq \varepsilon^*(\ell)$, where $\varepsilon^*(\ell) > 0$ is a constant depending on ℓ and $\mu > 0$ is a constant independent of ℓ . The previous bounds highlight the “low-pass” filter behaviors of the low-power high-gain observer (6) and moreover the remarkable feature of having an asymptotic gain between the measurement noise and the i -th error component that is proportional to ε powered at a value that increases as long as “higher” components of the errors are considered, as opposed to the standard case in which the asymptotic gain depends on ε regardless of the value of i . This fact, which is strongly related to the relative degree properties mentioned above, clearly shows that the new observer behaves better than the standard one as far as ε tends to zero, namely, as far as high-frequency noise is concerned. To better realize the improvements of the low-power high-gain observer over the standard one in relation to the sensitivity to high-frequency measurement noise, we recall the example reported in Astolfi and

Low-Power High-Gain Observers, Table 1

Normalized asymptotic errors in presence of measurement noise

Standard high-gain observer $\hat{x}$	Low-power high-gain observer $\hat{x}$	Low-power high-gain observer $\hat{x}'$
$\ \hat{x}_1 - x_1\ _a = 0.15$	$\ \hat{x}_{11} - x_1\ _a = 0.06$	$\ \hat{x}_{11} - x_1\ _a = 0.06$
$\ \hat{x}_2 - x_2\ _a = 8$	$\ \hat{x}_{21} - x_2\ _a = 0.2$	$\ \hat{x}_{12} - x_1\ _a = 3$
$\ \hat{x}_3 - x_3\ _a = 2 \cdot 10^2$	$\ \hat{x}_{31} - x_3\ _a = 0.2$	$\ \hat{x}_{22} - x_3\ _a = 3$
$\ \hat{x}_4 - x_4\ _a = 2.5 \cdot 10^3$	$\ \hat{x}_{41} - x_4\ _a = 0.1$	$\ \hat{x}_{32} - x_4\ _a = 2$
$\ \hat{x}_5 - x_5\ _a = 10^4$	$\ \hat{x}_{42} - x_5\ _a = 0.3$	$\ \hat{x}_{42} - x_5\ _a = 0.3$

Marconi (2015). Therein it is shown that the Van der Pol equation

$$\ddot{z} = -\rho^2 z + \sigma(1 - z^2)\dot{z}, \quad y = z + v(t),$$

with uncertain constant positive parameters ρ, σ can be immersed into a system of the form (1) with $n = 5$ with a known expression of φ that can be found in Astolfi and Marconi (2015). The performances of the two observers (2) and (6) have been tested with the design parameters chosen as indicated in Astolfi and Marconi (2015) and with high-frequency measurement noise $v(t) = a_N \sin(\omega_N t)$ with $a_N = 10^{-2}$ and $\omega_N = 10^3$. Table 1 shows the normalized asymptotic error magnitudes of the two observers, where the normalized asymptotic error for the i -th estimate is defined as $\|\hat{x}_i - x_i\|_a = \lim_{t \rightarrow \infty} \sup \|\hat{x}_i(t) - x_i(t)\|/a_N$. The numerical results show a remarkable improvement of the sensitivity to high-frequency measurement noise of the new observer with respect to the standard one confirming the analysis of this section. More examples can be also found in Astolfi et al. (2018).

Variants of the Low-Power Observer

Some variants of the low-power high-gain observer (6) are possible in order to deal with a more general class of systems or to improve performances. For instance, the low-power construction can be extended to the class of observed systems that, possibly after a change of coordinates, can be expressed in the following *generalized observability canonical form* (see Gauthier and Kupka 2001)

$$\begin{aligned} \dot{x}_1 &= f_1(\mathbf{x}_1, x_2) \\ &\vdots \\ \dot{x}_i &= f_i(\mathbf{x}_i, x_{i+1}) \\ &\vdots \\ \dot{x}_n &= f_n(\mathbf{x}_n) \\ y &= h(x_1) \end{aligned} \quad (8)$$

where $x = \text{col}(x_1, \dots, x_n) \in X$ is the state, $\mathbf{x}_i = \text{col}(x_1, \dots, x_i) \in \mathbb{R}^i$, and the functions f_i, h are Lipschitz functions in X fulfilling the following

$$\begin{aligned} \frac{\partial h(x_1)}{\partial x_1} &\neq 0, \quad \frac{\partial f_i(\mathbf{x}_i, x_{i+1})}{\partial x_{i+1}} \neq 0, \\ i &= 1, \dots, n-1 \end{aligned}$$

for all $x \in X$. This form clearly encompasses the one in (1). As shown in Wang et al. (2017), the low-power high-gain observer designed in these coordinates takes the form

$$\begin{aligned} \dot{\hat{x}}_{11} &= f_1(\hat{\mathbf{x}}_1, \hat{x}_{12}) + \ell c_{11}(y - h(\hat{x}_1)), \\ \dot{\hat{x}}_{12} &= f_2(\hat{\mathbf{x}}_2, \hat{x}_{22}) + \ell^2 c_{12}(y - h(\hat{x}_1)) \\ &\vdots \\ \dot{\hat{x}}_{i1} &= f_i(\hat{\mathbf{x}}_i, \hat{x}_{i2}) + \ell c_{i1}(\hat{x}_{(i-1)2} - \hat{x}_{i1}) \\ \dot{\hat{x}}_{i2} &= f_{i+1}(\hat{\mathbf{x}}_{i+1}, \hat{x}_{(i+1)2}) \\ &\quad + \ell^2 c_{i2}(\hat{x}_{(i-1)2} - \hat{x}_{i1}) \\ i &= 2, \dots, n-2 \\ &\vdots \\ \dot{\hat{x}}_{(n-1)1} &= f_{n-1}(\hat{\mathbf{x}}_{n-1}, \hat{x}_{(n-1)2}) \\ &\quad + \ell c_{(n-1)1}(\hat{x}_{(n-2)2} - \hat{x}_{(n-1)1}) \\ \dot{\hat{x}}_{(n-1)2} &= f_n(\hat{\mathbf{x}}_n) + \ell^2 c_{(n-1)2}(\hat{x}_{(n-2)2} - \hat{x}_{(n-1)1}) \end{aligned} \quad (9)$$

in which the estimate is given by $\hat{x} = \text{col}(\hat{x}_{11}, \hat{x}_{21}, \dots, \hat{x}_{(n-1)1}, \hat{x}_{(n-2)2})$, the variables

$\hat{\mathbf{x}}_i$ are used to denote the compact notation $\hat{\mathbf{x}}_i := \text{col}(\hat{x}_1, \dots, \hat{x}_i) \in \mathbb{R}^i$ and the $c_{i,j}$, $i = 1, 2$, $j = 1, \dots, n-1$ are positive coefficients to be properly chosen, according to Wang et al. (2017). Similar to the above, it can be proved that if ℓ is taken sufficiently large, then the state of the cascade (8)–(9) fulfills

$$\|\hat{x}(t) - x(t)\| \leq \alpha \ell^{n-1} \exp(-\beta \ell t),$$

for some positive α and β not dependent on ℓ . In presence of state disturbances d and measurement noise v , asymptotic bounds similar to the ones presented in sections “The Low-Power Observer Structure” and “Sensitivity to High-Frequency Noise” can be given.

$$\begin{aligned}
 \dot{\hat{x}}_{11} &= \hat{x}_{12} + \ell c_{11}(y - \hat{x}_{11}), \\
 \dot{\hat{x}}_{12} &= \text{sat}_{r_{i+2}}(\hat{x}_{22}) + \ell^2 c_{12}(y - \hat{x}_{11}) \\
 &\vdots \\
 \dot{\hat{x}}_{i1} &= \hat{x}_{i2} + \ell c_{i1}(\text{sat}_{r_i}(\hat{x}_{(i-1)2}) - \hat{x}_{i1}) \\
 \dot{\hat{x}}_{i2} &= \text{sat}_{r_{i+2}}(\hat{x}_{(i+1)2}) + \ell^2 c_{i2}(\text{sat}_{r_i}(\hat{x}_{(i-1)2}) - \hat{x}_{i1}) \quad i = 2, \dots, n-2 \\
 &\vdots \\
 \dot{\hat{x}}_{(n-1)1} &= \hat{x}_{(n-1)2} + \ell c_{(n-1)1}(\text{sat}_{r_{n-1}}(\hat{x}_{(n-2)2}) - \hat{x}_{(n-1)1}) \\
 \dot{\hat{x}}_{(n-1)2} &= \varphi_s(\hat{x}) + \ell^2 c_{(n-1)2}(\text{sat}_{r_{n-1}}(\hat{x}_{(n-2)2}) - \hat{x}_{(n-1)1}) \\
 \dot{\hat{x}}_{n1} &= \varphi_s(\hat{x}) + \ell c_{n1}(\text{sat}_{r_n}(\hat{x}_{(n-1)2}) - \hat{x}_{n1})
 \end{aligned} \tag{10}$$

where the state is now of dimension $2n-1$, the coefficients $c_{i,j}$ can be designed by following the procedure in the algorithm above with $c_{n1} > 0$ arbitrarily taken, and $\varphi_s(\cdot)$ is defined as before. By selecting as estimate $\hat{x} = \text{col}(\hat{x}_{11}, \hat{x}_{21}, \dots, \hat{x}_{(n-1)1}, \hat{x}_{n1})$, it turns out that for sufficiently large value of ℓ , the state of the cascade (1)–(10) fulfills the following asymptotic bound

$$\|\hat{x}(t) - x(t)\| \leq \min \left\{ \alpha \ell^{n-1} \exp(-\beta \ell t), \delta \right\} \tag{11}$$

where (α, β, δ) are positive constants independent of ℓ . The bound (11) guarantees that the estimation error $\|\hat{x} - x\|$ is bounded, for all time, by a constant which is independent of ℓ , preventing therefore the peaking phenomenon.

A further interesting extension of the low-power observer is the one proposed in Astolfi et al. (2018) in order to eliminate the peaking phenomenon associated with the high-gain structures, both standard and low-power, presented before. In particular, consider the canonical observability form (1), define $r_i > 0$ as

$$r_i := \max_{x \in X} |x_i| \quad i = 1, \dots, n,$$

and let $\text{sat}_{r_i}(\cdot)$ be the piecewise linear saturation function with saturation level r_i , $i = 1, \dots, n$. Then, the low-power peaking-free observer takes the following form (Astolfi et al. 2018)

Furthermore, the asymptotic properties of the estimation error $\|\hat{x} - x\|$ of the low-power high-gain observer (7) are retained.

Summary

In this entry we presented low-power high-gain observers as potential effective alternative to conventional high-gain solutions. With respect to the latter, the low-power version has the advantage of having the high-gain parameter just powered up to the order to 2 regardless of the dimension of the observed system and of having substantial better performances in terms of sensitivity to high-frequency measurement noises. The low-power solution preserves all the important features of conventional high-gain observers that

make them an important tool in many control contexts. Application contexts of the low-power can be found in Astolfi et al. (2017) where output regulation and output feedback stabilization problems are addressed. It is worth also observing that the low-power idea can be extended to the problem of recursive stabilization through backstepping in order to limit the power of the stabilizing high-gain parameter as shown in Astolfi et al. (2017).

Cross-References

- [Observer-Based Control](#)
- [Observers for Nonlinear Systems](#)
- [Observers in Linear Systems Theory](#)

Bibliography

- Astolfi D, Marconi L (2015) A high-gain nonlinear observer with limited gain power. *IEEE Trans Autom Control* 60(11):3059–3064
- Astolfi D, Marconi L, Teel AR (2016a) Low-power peaking-free high-gain observers for nonlinear systems. In: 2016 European control conference (ECC), 1424–1429
- Astolfi D, Marconi L, Praly L, Teel A (2016b) Sensitivity to high-frequency measurement noise of nonlinear high-gain observers. *IFAC-PapersOnLine* 49(18): 862–866
- Astolfi D, Isidori A, Marconi L (2017) Output regulation via low-power construction. In: *Feedback stabilization of controlled dynamical systems*. Springer, Cham, pp 143–165
- Astolfi D, Marconi L, Praly L, Teel AR (2018) Low-power peaking-free high-gain observers. *Automatica* 98: 169–179
- Atassi AN, Khalil HK (1999) A separation principle for the stabilization of a class of nonlinear systems. *IEEE Trans Autom Control* 44(9):1672–1687
- Edwards C, Spurgeon S, Patton RJ (2000) Sliding mode observers for fault detection and isolation. *Automatica* 36(4):541–553
- Gauthier JP, Kupka I (2001) *Deterministic observation theory and applications*. Cambridge University Press, Cambridge
- Isidori A (2017) Regulation and tracking in nonlinear systems. In: *Lectures in feedback design for multivariable systems*. Springer, Cham, pp 341–364
- Khalil HK, Praly L (2014) High-gain observers in nonlinear feedback control. *Int J Robust Nonlinear Control* 24(6):993–1015
- Teel AR, Praly L (1995) Tools for semiglobal stabilization by partial state and output feedback. *SIAM J Control Optim* 33(5):1443–1488
- Wang L, Astolfi D, Su H, Marconi L (2017) High-gain observers with limited gain power for systems with observability canonical form. *Automatica* 75:16–23

LTl Systems

- [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)

Lyapunov Methods in Power System Stability

Hsiao-Dong Chiang

School of Electrical and Computer Engineering,
Cornell University, Ithaca, NY, USA

Abstract

Energy functions, an extension of the Lyapunov functions, have been practically used in electric power systems for several applications. A comprehensive energy function theory for general nonlinear autonomous dynamical systems, along with its applications to electric power systems, will be summarized in this entry. Several schemes for constructing numerical energy functions for power system transient stability models expressed as a set of general differential-algebraic equations are presented. A practical application of energy functions for direct stability assessment of power system transient stability is also presented.

Keywords

Lyapunov functions · Energy functions · Direct stability analysis · Electric power systems · Large-scale nonlinear systems

Introduction

Energy functions, an extension of the Lyapunov functions, have been practically used in electric power systems for several applications. A comprehensive energy function theory for general nonlinear autonomous dynamical systems along with its applications to electric power systems will be summarized in this entry.

We consider a general nonlinear autonomous dynamical system described by the following equation:

$$\dot{x}(t) = f(x(t)) \quad (1)$$

We say a function $V : R^n \rightarrow R$ is an energy function for the system (1) if the following three conditions are satisfied (Chiang et al. 1987):

- (E1): The derivative of the energy function $V(x)$ along any system trajectory $x(t)$ is nonpositive, i.e., $\dot{V}(x(t)) \leq 0$.
- (E2): If $x(t)$ is a nontrivial trajectory (i.e., $x(t)$ is not an equilibrium point), then along the nontrivial trajectory $x(t)$, the set $\{t \in R : V(x(t)) = 0\}$ has measure zero in R .
- (E3): That a trajectory $x(t)$ has a bounded value of $V(x(t))$ for $t \in R^+$ implies that the trajectory $x(t)$ is also bounded.

Condition (E1) indicates that the value of an energy function is nonincreasing along its trajectory, but does not imply that the energy function is strictly decreasing along any trajectory. Conditions (E1) and (E2) imply that the energy function is strictly decreasing along any system trajectory. Property (E3) states that the energy function is a proper map along any system trajectory but need not be a proper map for the entire state space. Obviously, an energy function may not be a Lyapunov function.

The task of constructing an energy function for a (post-fault) transient stability model is essential to direct stability analysis of power systems. The role of the energy function is to make feasible a direct determination of whether a given point (such as the initial point of a post-fault power system) lies inside the stability region of a post-fault SEP without performing numerical integration. It has been shown that a general (analytical) energy

function for power systems with losses does not exist (Chiang 1989). One key implication is that any general procedure attempting to construct an energy function for a lossy power system transient stability model must include a step that checks for the existence of an energy function. This step essentially plays the same role as the Lyapunov equation in determining the stability of an equilibrium point.

Several schemes for constructing numerical energy functions for power system transient stability models expressed as a set of general differential-algebraic equations (DAEs) are available (Chu and Chiang 1999, 2005). An energy function for an AC/DC power system represented by a structure-preserving model considering detailed DC dynamics was developed in Jiang and Chiang (2013). The function was theoretically shown to satisfy the required conditions (E1) and (E2). The monotonic decreasing property of the energy function was numerically demonstrated in several examples.

Energy Function Theory

In general, the dynamical behaviors of trajectories of general nonlinear systems can be very complicated. The asymptotical behaviors (i.e., the ω -limit set) of trajectories can be quasiperiodic trajectories or chaotic trajectories. However, as shown below, every trajectory of system (1) having an energy function has only two modes of behaviors: its trajectory either converges to an equilibrium point or goes to infinity (becomes unbounded) as time increases. This result is explained in the following theorem.

Theorem 1 (Global Behavior of Trajectories)

If there exists a function satisfying condition (E1) and condition (E2) of the energy function for system (1), then every bounded trajectory of system (1) converges to one of the equilibrium points.

Theorem 1 asserts that there does not exist any limit cycle (oscillatory behavior) or bounded complicated behavior such as an almost periodic trajectory, chaotic motion, etc. in the sys-

tem. We next show a sharper result, asserting that every trajectory on the stability boundary must converge to one of the unstable equilibrium points (UEPs) on the stability boundary. Recall that for a hyperbolic equilibrium point, it is an (asymptotically) *stable equilibrium point* if all the eigenvalues of its corresponding Jacobian have negative real parts; otherwise, it is an *unstable equilibrium point*. Let $\hat{x}$ be a hyperbolic equilibrium point. Its stable and unstable manifolds, $W^s(\hat{x})$ and $W^u(\hat{x})$, are well defined. There are many physical systems such as electric power systems containing multiple stable equilibrium points. A useful concept for these kinds of systems is that of the *stability region* (also called the *region of attraction*). The stability region of a stable equilibrium point x_s is defined as $A(x_s) := \{x \in R^n : \lim_{t \rightarrow \infty} \Phi_t = x_s\}$.

Theorem 2 (Trajectories on the Stability Boundary Chiang et al. 1987) *If there exists an energy function for system (1), then every trajectory on the stability boundary $\partial A(x_s)$ converges to one of the equilibrium points on the stability boundary.*

The significance of this theorem is that it offers an effective way to characterize the stability boundary. In fact, Theorem 2 asserts that the stability boundary $\partial A(x_s)$ is contained in the union of stable manifolds of the UEPs on the stability boundary, i.e.,

$$\partial A(x_s) \subseteq \bigcup_{x_i \in \{E \cap \partial A(x_s)\}} W^s(x_i)$$

The above two theorems give interesting results on the structure of the equilibrium points on the stability boundary. They lead to the development of a controlling UEP method for direct assessment of power system stability (Chiang et al. 1987; Chiang 2011). Moreover, Theorem 2 presents a necessary condition for the existence of certain types of equilibrium points on a *bounded* stability boundary.

Theorem 3 (Unbounded Stability Region Chiang et al. 1987) *If there exists an energy function for system (1) which has an asymptotically*

stable equilibrium point x_s and if $\partial A(x_s)$ contains no source, then the stability region $A(x_s)$ is unbounded.

A direct application of Theorem 3 is that the stability region of a power system stability model which admits an energy function is unbounded.

Optimally Estimating the Stability Region Using Energy Functions

In this section, we focus on how to optimally determine the critical level value of an energy function for estimating the stability boundary $\partial A(x_s)$. We consider the following set:

$$S_v(k) = \{x \in R^n : V(x) < k\} \quad (2)$$

where $V(\bullet) : R^n \rightarrow R$ is an energy function. We shall call the boundary of set (2) $\partial S(k) := \{x \in R^n : V(x) = k\}$ the level set (or constant energy surface) and k the level value. Generally speaking, this set $S(k)$ can be very complicated with several connected components, even for the two-dimensional case. We use the notation $S_k(x_s)$ to denote the only component of the several disjoint connected components of S_k that contains the stable equilibrium point x_s .

Theorem 4 (Optimal Estimation) *Consider the nonlinear system (1) which has an energy function $V(x)$. Let x_s be an asymptotically stable equilibrium point whose stability region $A(x_s)$ is not dense in R^n . Let E_1 be the set of type one equilibrium points and $\hat{c} = \min_{x_i \in \partial A(x_s) \cap E_1} V(x_i)$. Then,*

1. $S_{\hat{c}}(x_s) \subset A(x_s)$
2. *The set $\{S_b(x_s) \cap \bar{A}^c(x_s)\}$ is nonempty for any number $b > \hat{c}$.*

This theorem leads to an optimal estimation of the stability region $A(x_s)$ via an energy function $V(\cdot)$ (Chiang 1989). For the purpose of illustration, we consider the following simple example:

$$\begin{aligned}\dot{x}_1 &= -\sin x_1 - 0.5 \sin(x_1 - x_2) + 0.01 \\ \dot{x}_2 &= -0.5 \sin x_2 - 0.5 \sin(x_2 - x_1) + 0.05\end{aligned}\quad (3)$$

It is easy to show that the following function is an energy function for system (3):

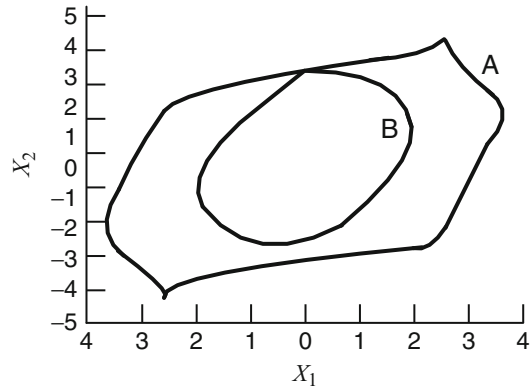
$$\begin{aligned}V(x_1, x_2) &= -2 \cos x_1 - \cos x_2 - \cos(x_1 - x_2) \\ &\quad - 0.02x_1 - 0.1x_2\end{aligned}\quad (4)$$

The point $x_s(x_1^s, x_2^s) = (0.02801, 0.06403)$ is the stable equilibrium point whose stability region is to be estimated. Applying the optimal scheme to system (3), we have the critical level value of 0.31329. Curve A in Fig. 1 is the exact stability boundary $\partial A(x_s)$, while Curve B is the stability boundary estimated by the connected component (containing the s.e.p. x_s) of the constant energy surface. It can be seen that the critical level value, 0.31329, is indeed the optimal value.

Application of Energy Functions to Electric Power Grids

After decades of research and development in the energy-function-based direct methods and the time-domain simulation approach, it has become clear that the capabilities of direct methods and that of the time-domain approach complement each other. The current direction of development is to include appropriate direct methods and time-domain simulation programs within the body of overall power system stability simulation programs (Chiang et al. 1995; Chiang 1999, 2011; Fouad and Vittal 1991; Sauer and Pai 1998). In this development, the BCU direct method performs the function of fast screening and ranking of a large contingency list to identify unstable and critical contingencies and screen out stable contingencies which represent the majority ones. Only the identified unstable or potentially unstable contingencies are sent to the time-domain simulation program for detailed simulation/analysis.

The PJM Interconnection has successfully designed and implemented a Transient Stability



Lyapunov Methods in Power System Stability, Fig. 1 Curve A is the exact stability boundary $\partial A(x_s)$ of system (5), while Curve B is the stability boundary estimated by the constant energy surface (with a level value of 0.31329) of the energy function

Analysis & Control (TSA&C) system at its energy management system (EMS). The EMS provides a real-time snapshot including power flow case data from the state estimator, dynamic data, and a contingency list to the TSA&C application. For each real-time snapshot, TSA&C performed a transient stability assessment against the list of 3000 contingencies and calculated the stability limits on key transfer interfaces (Chiang et al. 2013). TEPCO-BCU was selected as the leading fast screening tool for improving the performance of the PJM TSA system (Tong et al. 2010). The TEPCO-BCU software was evaluated in the PJM TSA Test System environment. The evaluation period was chosen to include the scheduled outage season and the summer peak load season.

During this evaluation period, both the PJM TSA software and the TEPCO-BCU software were run periodically in parallel, every 15 min in the PJM TSA Test System environment. The goal was to evaluate TEPCO-BCU in a real-time environment as a transient stability analysis screening tool. The total number of contingencies involved in this evaluation is about 5.3 million. It is the ability to perform dynamic contingency screening on a large number of contingencies and filter out a much smaller number of contingencies

Lyapunov Methods in Power System Stability, Table 1 Overall performance of TEPCO-BCU for online contingency screening

Reliability measure	Screening measurement	Computation speed	Online computation
100%	92% to 99.5%	1.3 s	Yes

requiring further analysis that would make online TSA feasible.

A summary of the overall performance of TEPCO-BCU indicates that TEPCO-BCU is a fast and accurate screening tool for online transient stability analysis of large-scale power systems. Of the 5.3 million contingencies, all the unstable contingencies verified by detailed time domain simulation were classified by TEPCO-BCU as unstable. These unstable contingencies exhibit first-swing instability as well as multi-swing instability. The percentage of screening for stable contingencies is shown in Table 1. For a total of 5.29 million contingencies, on average, TEPCO-BCU consumes about 1.3556 s for the stability assessment of each contingency. Depending on the loading conditions and network topologies, the screening rate ranges from 92% to 99.5%, meaning the TEPCO-BCU quickly screens out, for every 100 stable contingencies, between 92 contingencies and 99.5 contingencies which are definitely stable.

This evaluation study confirms the author’s belief that theory-based solution methods can lead to practical applications in large-scale non-linear systems.

Summary and Future Directions

After decades of research and development in energy-function-based direct methods, it has become clear that the capabilities of the controlling UEP method and those of the time-domain simulation approach complement each other. The current direction of development is to incorporate the BCU-based controlling UEP method and time-domain simulation programs into the body of overall online transient stability assessments (online TSA). Online TSA has been

designed to provide power system operators with critical information, including the following:

- (i) Assessment of the transient stability of a power system subject to a list of contingencies.
- (ii) Available (power) transfer limits at key interfaces subject to transient stability constraints.

With the proliferation of renewable energy resources in the electric power grid, it becomes clear that online TSA needs to include the uncertainties brought by renewable energy into the overall assessment. The uncertainties of renewable energy can be expressed as a set of scenarios and put into a list of renewable contingencies (i.e., uncertainties). Hence, item (i) needs to extend into the following:

- (i) Assessment of the transient stability of a power system subject to a list of (network) contingencies and a list of renewable contingencies.

Cross-References

- ▶ [Lyapunov’s Stability Theory](#)
- ▶ [Model Order Reduction: Techniques and Tools](#)
- ▶ [Power System Voltage Stability](#)
- ▶ [Small Signal Stability in Electric Power Systems](#)

Recommended Reading

A recent book which contains a comprehensive treatment of energy functions theory and applications is Chiang (2011).
A recent book which contains a comprehensive treatment of stability region theory and applications is Chiang and Alberto (2015).

Bibliography

- Chiang HD (1989) Study of the existence of energy functions for power systems with losses. *IEEE Trans Circuits Syst CAS-36*(11):1423–1429
- Chiang HD (1999) Power system stability. In: Webster JG (ed) *Wiley encyclopedia of electrical and electronics engineering*. Wiley, New York, pp 104–137
- Chiang HD (2011) Direct methods for stability analysis of electrical power system: theoretical foundation, BCU methodologies and application. Wiley, Hoboken
- Chiang HD, Alberto LFC (2015) *Stability regions of nonlinear dynamical systems: theory, estimation, and applications*. Cambridge University Press, Cambridge
- Chiang HD, Thorp JS (1989) Stability regions of nonlinear dynamical systems: a constructive methodology. *IEEE Trans Autom Control* 34(12):1229–1241
- Chiang HD, Wu FF, Varaiya PP (1987) Foundations of direct methods for power system transient stability analysis. *IEEE Trans Circuits Syst CAS-34*(2):160–173
- Chiang HD, Chu CC, Cauley G (1995) Direct stability analysis of electric power systems using energy functions: theory, applications and perspective. Invited paper, *Proc. IEEE* (83)11:1497–1529
- Chiang HD, Wang CS, Li H (1999) Development of BCU classifiers for online dynamic contingency screening of electric power systems. *IEEE Trans Power Syst* 14(2):660–666
- Chiang HD, Li H, Tong J, Tada Y (2013) On-line transient stability screening of a practical 14,500-bus power system: methodology and evaluations. In: Khaitan SK, Gupta A (eds) *High performance computing in power and energy systems*. Springer, Berlin/New York, pp 335–358. ISBN:978-3-642-32682-0
- Chu CC, Chiang HD (1999) Constructing analytical energy functions for network-reduction power systems with models: framework and developments. *Circuits Syst Signal Process* 18(1):1–16
- Chu CC, Chiang HD (2005) Constructing analytical energy functions for network-preserving power system models. *Circuits Syst Signal Process* 24(4):363–383
- Fouad AA, Vittal V (1991) *Power system transient stability analysis: using the transient energy function method*. Prentice-Hall, Englewood Cliffs
- Jiang N, Chiang HD (2013) Energy function for power system with detailed DC model: construction and analysis. *IEEE Trans Power Syst* 28(4):3756–3764
- Sauer PW, Pai MA (1998) *Power system dynamics and stability*. Prentice-Hall, Upper Saddle River
- Tong J, Chiang HD, Tada Y (2010) On-line power system stability screening of practical power system models using TEPCO-BC. In: *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS 2010)*, 1:537–540

Lyapunov's Stability Theory

Hassan K. Khalil

Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA

Abstract

Lyapunov's theory for characterizing and studying the stability of equilibrium points is presented for time-invariant and time-varying systems modeled by ordinary differential equations.

Keywords

Asymptotic stability · Equilibrium point · Exponential stability · Global asymptotic stability · Hurwitz matrix · Invariance principle · Linearization · Lipschitz condition · Lyapunov function · Lyapunov surface · Negative (semi-) definite function · Perturbed system · Positive (semi-) definite function · Region of attraction · Stability · Time-invariant system · Time-varying system

Introduction

Stability theory plays a central role in systems theory and engineering. For systems represented by state models, stability is characterized by studying the asymptotic behavior of the state variables near steady-state solutions, like equilibrium points or periodic orbits. In this article, Lyapunov's method for determining the stability of equilibrium points is introduced. The attractive features of the method include a solid theoretical foundation, the ability to conclude stability without knowledge of the solution (no extensive simulation effort), and an analytical framework that makes it possible to study the effect of model perturbations and design feedback control. Its main drawback is the need to search for an auxiliary function that satisfies certain conditions.

Stability of Equilibrium Points

We consider a nonlinear system represented by the state model

$$\dot{x} = f(x) \quad (1)$$

where the n -dimensional **locally Lipschitz** function $f(x)$ is defined for all x in a domain $D \subset R^n$. A function $f(x)$ is locally Lipschitz at a point x_0 if it satisfies the **Lipschitz condition** $\|f(x) - f(y)\| \leq L\|x - y\|$ for all x, y in some neighborhood of x_0 , where L is a positive constant and $\|x\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$. The Lipschitz condition guarantees that Eq. (1) has a unique solution for given initial state $x(0)$. Suppose $\bar{x} \in D$ is an **equilibrium point** of Eq. (1); that is, $f(\bar{x}) = 0$. Whenever the state of the system starts at $\bar{x}$, it will remain at $\bar{x}$ for all future time. Our goal is to characterize and study the stability of $\bar{x}$. For convenience, we take $\bar{x} = 0$. There is no loss of generality in doing so because any equilibrium point $\bar{x}$ can be shifted to the origin via the change of variables $y = x - \bar{x}$. Therefore, we shall always assume that $f(0) = 0$ and study stability of the origin $x = 0$.

The equilibrium point $x = 0$ of Eq. (1) is **stable** if for each $\varepsilon > 0$, there is $\delta = \delta(\varepsilon) > 0$ such that $\|x(0)\| < \delta$ implies that $\|x(t)\| < \varepsilon$, for all $t \geq 0$. It is **asymptotically stable** if it is stable and δ can be chosen such that $\|x(0)\| < \delta$ implies that $x(t)$ converges to the origin as t tends to infinity. When the origin is asymptotically stable, the **region of attraction** (also called region of asymptotic stability, domain of attraction, or basin) is defined as the set of all points x such that the solution of Eq. (1) that starts from x at time $t = 0$ approaches the origin as t tends to ∞ . When the region of attraction is the whole space, we say that the origin is **globally asymptotically stable**. A stronger form of asymptotic stability arises when there exist positive constants c , k , and λ such that the solutions of Eq. (1) satisfy the inequality

$$\|x(t)\| \leq k\|x(0)\|e^{-\lambda t}, \quad \forall t \geq 0 \quad (2)$$

for all $\|x(0)\| < c$. In this case, the equilibrium point $x = 0$ is said to be **exponentially stable**. It is said to be **globally exponentially stable** if the inequality is satisfied for any initial state $x(0)$.

Linear Systems

For the linear time-invariant system

$$\dot{x} = Ax \quad (3)$$

the stability properties of the origin can be determined by the location of the eigenvalues of A . The origin is stable if and only if all the eigenvalues of A satisfy $\text{Re}[\lambda_i] \leq 0$ and for every eigenvalue with $\text{Re}[\lambda_i] = 0$ and algebraic multiplicity $q_i \geq 2$, $\text{rank}(A - \lambda_i I) = n - q_i$, where n is the dimension of x and q_i is the multiplicity of λ_i as a zero of $\det(\lambda I - A)$. The origin is globally exponentially stable if and only if all eigenvalues of A have negative real parts; that is, A is a **Hurwitz matrix**. For linear systems, the notions of asymptotic and exponential stability are equivalent because the solution is formed of exponential modes. Moreover, due to linearity, if the origin is exponentially stable, then the inequality of Eq. (2) will hold for all initial states.

Linearization

Suppose the function $f(x)$ of Eq. (1) is continuously differentiable in a domain D containing the origin. The Jacobian matrix $[\partial f / \partial x]$ is an $n \times n$ matrix whose (i, j) element is $\partial f_i / \partial x_j$. Let A be the Jacobian matrix evaluated at the origin $x = 0$. It can be shown that

$$f(x) = [A + G(x)]x, \quad \text{where } \lim_{x \rightarrow 0} G(x) = 0$$

This suggests that in a small neighborhood of the origin we can approximate the nonlinear system $\dot{x} = f(x)$ by its linearization about the origin $\dot{x} = Ax$. Indeed, we can draw conclusions about the stability of the origin as an equilibrium point for the nonlinear system by examining the eigenvalues of A . The origin of Eq. (1) is exponentially stable if and only if A is Hurwitz. It is unstable if $\text{Re}[\lambda_i] > 0$ for one or more of the eigenvalues of A . If $\text{Re}[\lambda_i] \leq 0$ for all i , with $\text{Re}[\lambda_i] = 0$ for

some i , we cannot draw a conclusion about the stability of the origin of Eq. (1).

Lyapunov's Method

Let $V(x)$ be a continuously differentiable scalar function defined in a domain $D \subset R^n$ that contains the origin. The function $V(x)$ is said to be **positive definite** if $V(0) = 0$ and $V(x) > 0$ for $x \neq 0$. It is said to be **positive semidefinite** if $V(x) \geq 0$ for all x . A function $V(x)$ is said to be **negative definite** or **negative semidefinite** if $-V(x)$ is positive definite or positive semidefinite, respectively. The derivative of V along the trajectories of Eq. (1) is given by

$$\dot{V}(x) = \sum_{i=1}^n \frac{\partial V}{\partial x_i} \dot{x}_i = \frac{\partial V}{\partial x} f(x)$$

where $[\partial V / \partial x]$ is a row vector whose i th component is $\partial V / \partial x_i$.

Lyapunov's stability theorem states that *the origin is stable if there is a continuously differentiable positive definite function $V(x)$ so that $\dot{V}(x)$ is negative semidefinite, and it is asymptotically stable if $\dot{V}(x)$ is negative definite*. A function $V(x)$ satisfying the conditions for stability is called a **Lyapunov function**. The surface $V(x) = c$, for some $c > 0$, is called a **Lyapunov surface** or a level surface.

When $\dot{V}(x)$ is only negative semidefinite, we may still conclude asymptotic stability of the origin if we can show that no solution can stay identically in the set $\{\dot{V}(x) = 0\}$, other than the zero solution $x(t) \equiv 0$. Under this condition, $V(x(t))$ must decrease toward 0, and consequently $x(t)$ converges to zero as t tends to infinity. This extension of the basic theorem is known as **the invariance principle**.

Lyapunov functions can be used to estimate the region of attraction of an asymptotically stable origin, that is, to find sets contained in the region of attraction. Let $V(x)$ be a Lyapunov function that satisfies the conditions of asymptotic stability over a domain D . For a positive constant c , let Ω_c be the component of $\{V(x) \leq$

$c\}$ that contains the origin in its interior. The properties of V guarantee that, by choosing c small enough, Ω_c will be bounded and contained in D . Then, every trajectory starting in Ω_c remains in Ω_c and approaches the origin as $t \rightarrow \infty$. Thus, Ω_c is an estimate of the region of attraction. If $D = R^n$ and $V(x)$ is radially unbounded, that is, $\|x\| \rightarrow \infty$ implies that $V(x) \rightarrow \infty$, then any point $x \in R^n$ can be included in a bounded set Ω_c by choosing c large enough. Therefore, *the origin is globally asymptotically stable if there is a continuously differentiable, radially unbounded function $V(x)$ such that for all $x \in R^n$, $V(x)$ is positive definite and $\dot{V}(x)$ is either negative definite or negative semidefinite but no solution can stay identically in the set $\{\dot{V}(x) = 0\}$ other than the zero solution $x(t) \equiv 0$* .

Time-Varying Systems

Equation (1) is time-invariant because f does not depend on t . The more general time-varying system is represented by

$$\dot{x} = f(t, x) \quad (4)$$

In this case, we may allow the Lyapunov function candidate V to depend on t . Let $V(t, x)$ be a continuously differentiable function defined for all $t \geq 0$ and $x \in D$. The derivative of V along the trajectories of Eq. (4) is given by

$$\dot{V}(t, x) = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x)$$

If there are positive definite functions $W_1(x)$, $W_2(x)$, and $W_3(x)$ such that

$$W_1(x) \leq V(t, x) \leq W_2(x) \quad (5)$$

$$\dot{V}(t, x) \leq -W_3(x) \quad (6)$$

for all $t \geq 0$ and all $x \in D$, then the origin is uniformly asymptotically stable, where "uniformly" indicates that the ε - δ definition of stability and the convergence of $x(t)$ to zero are independent of the initial time t_0 . Such uniformity annotation is not needed with time-invariant systems since the solution of a time-invariant state equation starting at time t_0 depends only on the difference

$t - t_0$, which is not the case for time-varying systems. If the inequalities of Eqs. (5) and (6) hold globally and $W_1(x)$ is radially unbounded, then the origin is globally uniformly asymptotically stable. If $W_1(x) = k_1\|x\|^a$, $W_2(x) = k_2\|x\|^a$, and $W_3(x) = k_3\|x\|^a$ for some positive constants k_1, k_2, k_3 , and a , then the origin is exponentially stable.

Perturbed Systems

Consider the system

$$\dot{x} = f(t, x) + g(t, x) \quad (7)$$

where f and g are continuous in t and locally Lipschitz in x , for all $t \geq 0$ and $x \in D$, in which $D \subset \mathbb{R}^n$ is a domain that contains the origin $x = 0$. Suppose $f(t, 0) = 0$ and $g(t, 0) = 0$ so that the origin is an equilibrium point of Eq. (7). We think of the system (7) as a perturbation of the nominal system (4). The perturbation term $g(t, x)$ could result from modeling errors, uncertainties, or disturbances. In a typical situation, we do not know $g(t, x)$, but we know some information about it, like knowing an upper bound on $\|g(t, x)\|$. Suppose the nominal system has an exponentially stable equilibrium point at the origin, what can we say about the stability of the origin as an equilibrium point of the perturbed system? A natural approach to address this question is to use a Lyapunov function for the nominal system as a Lyapunov function candidate for the perturbed system.

Let $V(t, x)$ be a Lyapunov function that satisfies

$$c_1\|x\|^2 \leq V(t, x) \leq c_2\|x\|^2 \quad (8)$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) \leq -c_3\|x\|^2 \quad (9)$$

$$\left\| \frac{\partial V}{\partial x} \right\| \leq c_4\|x\| \quad (10)$$

for all $x \in D$ for some positive constants c_1, c_2, c_3 , and c_4 . Suppose the perturbation term $g(t, x)$ satisfies the linear growth bound

$$\|g(t, x)\| \leq \gamma\|x\|, \quad \forall t \geq 0, \quad \forall x \in D \quad (11)$$

where γ is a nonnegative constant. We use V as a Lyapunov function candidate to investigate the stability of the origin as an equilibrium point for the perturbed system. The derivative of V along the trajectories of Eq. (7) is given by

$$\dot{V}(t, x) = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) + \frac{\partial V}{\partial x} g(t, x)$$

The first two terms on the right-hand side are the derivative of $V(t, x)$ along the trajectories of the nominal system, which is negative definite and satisfies the inequality of Eq. (9). The third term, $[\partial V / \partial x]g$, is the effect of the perturbation. Using Eqs. (9) through (11), we obtain

$$\begin{aligned} \dot{V}(t, x) &\leq -c_3\|x\|^2 + \left\| \frac{\partial V}{\partial x} \right\| \|g(t, x)\| \\ &\leq -c_3\|x\|^2 + c_4\gamma\|x\|^2 \end{aligned}$$

If $\gamma < c_3/c_4$, then

$$\leq -(c_3 - \gamma c_4)\|x\|^2, \quad (c_3 - \gamma c_4) > 0$$

which shows that the origin is an exponentially stable equilibrium point of the perturbed system (7).

Summary

Lyapunov's method is a powerful tool for studying the stability of equilibrium points. However, there are two drawbacks of the method. First, there is no systematic procedure for finding Lyapunov functions. Second, the conditions of the theory are only sufficient; they are not necessary. Failure of a Lyapunov function candidate to satisfy the conditions for stability or asymptotic stability does not mean that the origin is not stable or asymptotically stable. These drawbacks have been mitigated by a long history of using the method in the analysis and design of engineering systems, where various techniques for finding Lyapunov functions for specific systems have been determined.

Cross-References

- [Feedback Stabilization of Nonlinear Systems](#)
- [Input-to-State Stability](#)
- [Regulation and Tracking of Nonlinear Systems](#)

Recommended Reading

For an introduction to Lyapunov's stability theory at the level of first-year graduate students, the textbooks Khalil (2002), Sastry (1999), Slotine and Li (1991), and (Vidyasagar 2002) are recommended. The books by Bacciotti and Rosier (2005) and Haddad and Chellaboina (2008) cover a wider set of topics at the same introductory level. A deeper look into the theory is provided in the monographs Hahn (1967), Krasovskii (1963), Rouche et al. (1977), Yoshizawa (1966), and (Zubov 1964). Lyapunov's theory for discrete-time systems is presented in Haddad and Chellaboina (2008) and Qu (1998). The monograph Michel and Wang (1995) presents Lyapunov's stability theory for general dynamical systems, including functional and partial differential equations.

Bibliography

- Bacciotti A, Rosier L (2005) *Liapunov functions and stability in control theory*, 2nd edn. Springer, Berlin
- Haddad WM, Chellaboina V (2008) *Nonlinear dynamical systems and control*. Princeton University Press, Princeton
- Hahn W (1967) *Stability of motion*. Springer, New York
- Khalil HK (2002) *Nonlinear systems*, 3rd edn. Prentice Hall, Princeton
- Krasovskii NN (1963) *Stability of motion*. Stanford University Press, Stanford
- Michel AN, Wang K (1995) *Qualitative theory of dynamical systems*. Marcel Dekker, New York
- Qu Z (1998) *Robust control of nonlinear uncertain systems*. Wiley-Interscience, New York
- Rouche N, Habets P, Laloy M (1977) *Stability theory by Lyapunov's direct method*. Springer, New York
- Sastry S (1999) *Nonlinear systems: analysis, stability, and control*. Springer, New York
- Slotine J-JE, Li W (1991) *Applied nonlinear control*. Prentice-Hall, Englewood Cliffs
- Vidyasagar M (2002) *Nonlinear systems analysis*, classic edn. SIAM, Philadelphia
- Yoshizawa T (1966) *Stability theory by Liapunov's second method*. The Mathematical Society of Japan, Tokyo
- Zubov VI (1964) *Methods of A. M. Lyapunov and their applications*. P. Noordhoff Ltd., Groningen

Markov Chains and Ranking Problems in Web Search

Hideaki Ishii¹ and Roberto Tempo^{2†}

¹Department of Computer Science, Tokyo Institute of Technology, Yokohama, Japan

²CNR-IEIT, Politecnico di Torino, Torino, Italy

Keywords

Aggregation · Distributed randomized algorithms · Markov chains · Optimization · PageRank · Search engines · World Wide Web

Abstract

Markov chains refer to stochastic processes whose states change according to transition probabilities determined only by the states of the previous time step. They have been crucial for modeling large-scale systems with random behavior in various fields such as control, communications, biology, optimization, and economics. In this entry, we focus on their recent application to the area of search engines, namely, the PageRank algorithm employed at Google, which provides a measure of importance for each page in the web. We present several researches carried out with control theoretic tools such as aggregation, distributed randomized algorithms, and PageRank optimization. Due to the large size of the web, computational issues are the underlying motivation of these studies.

Introduction

For various real-world large-scale dynamical systems, reasonable models describing highly complex behaviors can be expressed as stochastic systems, and one of the most well-studied classes of such systems is that of Markov chains. A characteristic feature of Markov chains is that their behavior does not carry any memory. That is, the current state of a chain is completely determined by the state of the previous time step and not at all on the states prior to that step (Kumar and Varaiya 1986; Norris 1997).

Recently, Markov chains have gained renewed interest due to the extremely successful applications in the area of web search. The search engine of Google has been employing an algorithm known as PageRank to assist the ranking of search results. This algorithm models the network of web pages as a Markov chain whose states represent the pages that web surfers with various interests visit in a random fashion. The objective is to find an order among the pages according to their popularity and importance, and this is done by focusing on the structure of hyperlinks among pages.

[†]Roberto Tempo passed away before publication of this work was completed.

In this entry, we first provide a brief overview on the basics of Markov chains and then introduce the problem of PageRank computation. We proceed to provide further discussions on control theoretic approaches dealing with PageRank problems. The topics covered include aggregated Markov chains, distributed randomized algorithms, and Markov decision problems for link optimization.

Markov Chains

In the simplest form, a Markov chain takes its states in a finite state space with transitions in the discrete-time domain. The transition from one state to another is characterized completely by the underlying probability distribution.

Let $\mathcal{X}$ be a finite set given by $\mathcal{X} := \{1, 2, \dots, n\}$, which is called the state space. Consider a stochastic process $\{X_k\}_{k=0}^{\infty}$ taking values on this set $\mathcal{X}$. Such a process is called a Markov chain if it exhibits the following Markov property:

$$\text{Prob}\{X_{k+1} = j | X_k = i_k, X_{k-1} = i_{k-1}, \dots, X_0 = i_0\} = \text{Prob}\{X_{k+1} = j | X_k = i_k\},$$

where $\text{Prob}\{\cdot | \cdot\}$ denotes the conditional probability and $k \in \mathbb{Z}_+$. That is, the state at the next time step depends only on the current state and not those of previous times.

Here, we consider the homogeneous case where the transition probability is constant over time. Thus, we have for each pair $i, j \in \mathcal{X}$, the probability that the chain goes from state j to state i at time k expressed as

$$p_{ij} := \text{Prob}\{X_k = i | X_{k-1} = j\}, \quad k \in \mathbb{Z}_+.$$

In the matrix form, $P := (p_{ij})$ is called the transition probability matrix of the chain. It is obvious that all entries of P are nonnegative, and for each j , the entries of the j th column of P sum up to 1, i.e., $\sum_{i=1}^n p_{ij} = 1$ for $j \in \mathcal{X}$. In this respect, the matrix P is (column) stochastic (Horn and Johnson 1985).

In this entry, we assume that the Markov chain is ergodic, meaning that for any pair of states, the chain can make a transition from one to the other over time. In this case, the chain and the matrix P are called irreducible. This property is known to imply that P has a simple eigenvalue of 1. Thus, there exists a unique steady-state probability distribution $\pi \in \mathbb{R}^n$ given by

$$\pi = P\pi, \quad \mathbf{1}^T \pi = 1, \quad \pi_i > 0, \quad \forall i \in \mathcal{X},$$

where $\mathbf{1} \in \mathbb{R}^n$ denotes a vector with entries one. Note that in this distribution π , all entries are positive.

Ranking in Search Engines: PageRank Algorithm

At Google, PageRank is used to quantify the importance of each web page based on the hyperlink structure of the web (Brin and Page 1998; Ishii and Tempo 2014; Langville and Meyer 2006). A page is considered important if (i) many pages have links pointing to the page, (ii) such pages having links are important ones, and (iii) the numbers of links that such pages have are limited. Intuitively, these requirements are reasonable. For a web page, its incoming links can be viewed as votes supporting the page, and moreover the quality of the votes count through their importance as well as the number of votes that they make. Even if a minor page (with low PageRank) has many outgoing links, its contribution to the linked pages will not be substantial.

An interesting way to explain the PageRank is through the *random surfer model*: The random surfer starts from a randomly chosen page. Each time visiting a page, he/she follows a hyperlink in that page chosen at random with uniform probability. Hence, if the current page i has n_i outgoing links, then one of them is picked with probability $1/n_i$. If it happens that the current page has no outgoing link (e.g., at PDF documents), the surfer will use the back button. This process will be repeated. The PageRank value for a page

represents the probability of the surfer visiting the page. It is thus higher for pages visited more often by the surfer.

It is now clear that PageRank is obtained by describing the random surfer model as a Markov chain and then finding its stationary distribution. First, we express the network of web pages as the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \dots, n\}$ is the set of nodes corresponding to web page indices, while $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges for links among pages. Node i is connected to node j by an edge, i.e., $(i, j) \in \mathcal{E}$, if page i has an outgoing link to page j .

Let $x_i(k)$ be the distribution of the random surfer visiting page i at time k , and let $x(k)$ be the vector containing all $x_i(k)$. Given the initial distribution $x(0)$, which is a probability vector, i.e., $\sum_{i=1}^n x_i(0) = 1$, the evolution of $x(k)$ can be expressed as

$$x(k+1) = Ax(k). \quad (1)$$

The link matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ is given by $a_{ij} = 1/n_j$ if $(j, i) \in \mathcal{E}$ and 0 otherwise, where n_j is the number of outgoing links of page j . Note that this matrix A is the transition probability matrix of the random surfer. Clearly, it is stochastic, and thus $x(k)$ remains a probability vector so that $\sum_{i=1}^n x_i(k) = 1$ for all k .

As mentioned above, PageRank is the stationary distribution of the process (1) under the assumption that the limit exists. Hence, the PageRank vector is given by $x^* := \lim_{k \rightarrow \infty} x(k)$. In other words, it is the solution of the linear equation

$$x^* = Ax^*, \quad x^* \in [0, 1]^n, \quad \mathbf{1}^T x^* = 1. \quad (2)$$

Notice that the PageRank vector x^* is a non-negative unit eigenvector for the eigenvalue 1 of A . Such a vector exists since the matrix A is stochastic but may not be unique; the reason is that A is a reducible matrix since in the web, not every pair of pages can be connected by simply following links. To resolve this issue, a slight modification is necessary in the random surfer model.

The idea of the *teleportation model* is that the random surfer, after a while, becomes bored and stops following the hyperlinks. At such an instant, the surfer “jumps” to another page not directly connected to the one currently visiting. This page can be in fact completely unrelated in the domains and/or the contents. All n pages in the web have the same probability $1/n$ to be reached by a jump.

The probability to make such a jump is denoted by $m \in (0, 1)$. The original transition probability matrix A is now replaced with the modified one $M \in \mathbb{R}^{n \times n}$ defined by

$$M := (1 - m)A + \frac{m}{n} \mathbf{1}\mathbf{1}^T. \quad (3)$$

For the value of m , we take $m = 0.15$ as reported in the original algorithm in Brin and Page (1998). Notice that M is a positive stochastic matrix. By Perron’s theorem (Horn and Johnson 1985), the eigenvalue 1 is of multiplicity 1 and is the unique eigenvalue with the maximum modulus. Further, the corresponding eigenvector is positive. Hence, we redefine the vector x^* in (2) by using M instead of A as follows:

$$x^* = Mx^*, \quad x^* \in [0, 1]^n, \quad \mathbf{1}^T x^* = 1. \quad (4)$$

Due to the large dimension of the link matrix M , the computation of x^* is difficult. The solution employed in practice is based on the power method given by

$$x(k+1) = Mx(k) = (1 - m)Ax(k) + \frac{m}{n} \mathbf{1}, \quad (5)$$

where the initial vector $x(0) \in \mathbb{R}^n$ is a probability vector. The second equality above follows from the fact $\mathbf{1}^T x(k) = 1$ for $k \in \mathbb{Z}_+$. For implementation, the form on the far right-hand side is important, using only the sparse matrix A and not the dense matrix M . This method asymptotically finds the value vector as $x(k) \rightarrow x^*, k \rightarrow \infty$.

Aggregation Methods for Large-Scale Markov Chains

In dealing with large-scale Markov chains, it is often desirable to predict their dynamic behaviors from reduced-order models that are more computationally tractable. This enables us, for example, to analyze the system performance at a macroscale with some approximation under different operating conditions. Aggregation refers to partitioning or grouping the states so that the states in each group can be treated as a whole. The technique of aggregation is especially effective for Markov chains possessing sparse structures with strong interactions among states in the same group and weak interactions among states in different groups. Such methods have been extensively studied, motivated by applications in queueing networks, power systems, etc. (Meyer 1989).

In the context of the PageRank problem, such sparse interconnection can be expressed in the link matrix A with a block-diagonal structure (after some coordinate change, if necessary). The entries of the matrix A are dense along its diagonal in blocks, and those outside the blocks take small values. More concretely, we write

$$A = I + B + \epsilon C, \quad (6)$$

where B is a block-diagonal matrix given as $B = \text{diag}(B_{11}, B_{22}, \dots, B_{NN})$; B_{ii} is the $\tilde{n}_i \times \tilde{n}_i$ matrix corresponding to the i th group with $\tilde{n}_i$ member pages for $i = 1, 2, \dots, N$; and ϵ is a small positive parameter. Here, the non-diagonal entries of B_{ii} are the same as those in the same diagonal block of A , but the diagonal entries are chosen such that $I + B_{ii}$ becomes stochastic and thus take nonpositive values. Thus, both B and C have column sums equal to zero. The small ϵ suggests us that states can be aggregated into N groups with strong interactions within the groups, but connections among different groups are weak. This class of Markov chains is known as nearly completely decomposable. In general, however, it is difficult to uniquely determine the form (6) for a given chain.

To exploit the sparse structure in the computation of stationary probability distributions, one approach is to carry out decomposition or aggregation of the chains. The basic approach here is (i) to compute the local stationary distributions for $I + B_{ii}$, (ii) to find the global stationary distribution for a chain representing the group interactions, and (iii) to finally use the obtained vectors to compute exact/approximate distribution for the entire chain; for details, see Meyer (1989). By interpreting such methods from the control theoretic viewpoints, in Phillips and Kokotovic (1981) and Aldaheri and Khalil (1991), singular perturbation approaches have been developed. These methods lead us to the two-time scale decomposition of (controlled) Markov chain recursions.

In the case of PageRank computation, sparsity is a relevant property since it is well-known that many links in the web are intra-host ones, connecting pages within the same domains or directories. However, in the real web, it is easy to find pages that have only a few outlinks, but some of them are external ones. Such pages will prevent the link matrix from having small ϵ when decomposed in the form (6). Hence, the general aggregation methods outlined above are not directly applicable.

An aggregation-based method suitable for PageRank computation is proposed in Ishii et al. (2012). There, the sparsity in the web is expressed by the limited number of external links pointing toward pages in other groups. For each page i , the node parameter $\delta_i \in [0, 1]$ is given by

$$\delta_i := \frac{\# \text{ external outgoing links}}{\# \text{ total outgoing links}}.$$

Note that smaller δ_i implies sparser networks. In this approach, for a given bound δ , the condition $\delta_i \leq \delta$ is imposed only in the case page i belongs to a group consisting of multiple members. Thus, a page forming a group by itself is not required to satisfy the condition. This means that we can regroup the pages by first identifying pages that violate this condition in the initial groups and then making them separately as *single* groups. By repeating these steps, it is always possible to

obtain groups for a given web. Once the grouping is settled, an aggregation-based algorithm can be applied, which computes an approximated PageRank vector. A characteristic feature is the trade-off between the accuracy in PageRank computation and the node parameter δ . More accurate computation requires a larger number of groups and thus a smaller δ .

Distributed Randomized Computation

For large-scale computation, distributed algorithms can be effective by employing multiple processors to compute in parallel. There are several methods of constructing algorithms to find stationary distributions of large Markov chains. In this section, motivated by the current literature on multi-agent systems, sequential distributed randomized approaches of gossip type are described for the PageRank problem.

In *gossip-type* distributed algorithms, nodes make decisions and transmit information to their neighbors in a random fashion. That is, at any time instant, each node decides whether to communicate or not depending on a random variable. The random property is important to make the communication asynchronous so that simultaneous transmissions resulting in collisions can be avoided. Moreover, there is no need of any centralized decision-maker or fixed order among pages.

More precisely, each page $i \in \mathcal{V}$ is equipped with a random process $\eta_i(k) \in \{0, 1\}$ for $k \in \mathbb{Z}_+$. If at time k , $\eta_i(k)$ is equal to 1, then page i broadcasts its information to its neighboring pages connected by outgoing links. All pages involved at this time renew their values based on the latest available data. Here, $\eta_i(k)$ is assumed to be an independent and identically distributed (i.i.d.) random process, and its probability distribution is given by $\text{Prob}\{\eta_i(k) = 1\} = \alpha$, $k \in \mathbb{Z}_+$. Hence, all pages are given the same probability α to initiate an update.

One of the proposed randomized approaches is based on the so-called asynchronous iteration algorithms for distributed computation of fixed

points in the field of numerical analysis (Bertsekas and Tsitsiklis 1989). The distributed update recursion is given as

$$\check{x}(k+1) = \check{M}_{\eta_1(k), \dots, \eta_n(k)} \check{x}(k), \quad (7)$$

where the initial state $\check{x}(0)$ is a probability vector, and the distributed link matrices $\check{M}_{p_1, \dots, p_n}$ are given as follows: Its (i, j) th entry is equal to $(1-m)a_{ij} + m/n$ if $p_i = 1$; 1 if $p_i = 0$ and $i = j$; and 0 otherwise. Clearly, these matrices keep the rows of the original link matrix M in (3) for pages initiating updates. Other pages just keep their previous values. Thus, these matrices are not stochastic. From this update recursion, the PageRank x^* is probabilistically obtained (in the mean square sense and in probability one), where the convergence speed is exponential in time k . Note that in this scheme (7), due to the way the distributed link matrices are constructed, each page needs to know which pages have links pointing toward it. This implies that popular pages linked by a number of pages must have extra memory to keep the data of such links.

Another recently developed approach Ishii and Tempo (2010) and Zhao et al. (2013) has several notable differences from the asynchronous iteration approach above. First, the pages need to transmit their states only over their outgoing links; the information of such links are by default available locally, and thus, pages are not required to have the extra memory regarding incoming links. Second, it employs stochastic matrices in the update as in the centralized scheme; this aspect is utilized in the convergence analysis. As a consequence, it is established that the PageRank vector x^* is computed in a probabilistic sense through the time average of the states $x(0), \dots, x(k)$ given by $y(k) := 1/(k+1) \sum_{\ell=0}^k x(\ell)$. The convergence speed in this case is of order $1/k$.

More recent studies have shown that exponential convergence is possible by distributed algorithms utilizing only outgoing links. The approaches in Lagoa et al. (2017) and You et al. (2017) are to view the PageRank problem as a least-square problem and thus employ distributed optimization techniques. On the other hand, in

Ishii and Suzuki (2018), an alternative approach is proposed by reinterpreting the PageRank problem and writing the PageRank vector x^* given in (4) in the form of a Neumann series as

$$x^* = \sum_{t=0}^{\infty} [(1-m)A]^t \frac{m}{n} \mathbf{1}.$$

A distributed algorithm based on this formulation is shown to have fast convergence speed there.

PageRank Optimization via Hyperlink Designs

For owners of websites, it is of particular interest to raise the PageRank values of their web pages. Especially in the area of e-business, this can be critical for increasing the number of visitors to their sites. The values of PageRank can be affected by changing the structure of hyperlinks in the owned pages. Based on the random surfer model, intuitively, it makes sense to arrange the links so that surfers will stay within the domain of the owners as long as possible.

PageRank optimization problems have rigorously been considered in, for example, de Kerchove et al. (2008), Fercoq et al. (2013), and Csáji et al. (2014). In general, these are combinatorial optimization problems since they deal with the issues on where to place hyperlinks, and thus the computation for solving them can be prohibitive especially when the web data is large. However, the work Fercoq et al. (2013) has shown that the problem can be solved in polynomial time. In what follows, we discuss a simplified discrete version of the problem setup of this work.

Consider a subset $\mathcal{V}_0 \subset \mathcal{V}$ of web pages over which a webmaster has control. The objective is to maximize the total PageRank of the pages in this set $\mathcal{V}_0$ by finding the outgoing links from these pages. Each page $i \in \mathcal{V}_0$ may have constraints such as links that must be placed within the page and those that cannot be allowed. All other links, i.e., those that one can decide to have or not, are the design parameters. Hence, the PageRank optimization problem can be stated as

$$\begin{aligned} \max \{ & U(x^*, M) : x^* = Mx^*, x^* \in [0, 1]^n, \\ & \mathbf{1}^T x^* = 1, M \in \mathcal{M} \}, \end{aligned}$$

where U is the utility function $U(x^*, M) := \sum_{i \in \mathcal{V}_0} x_i^*$ and $\mathcal{M}$ represents the set of admissible link matrices in accordance with the constraints introduced above.

In Fercoq et al. (2013), an extended continuous problem is also studied where the set $\mathcal{M}$ of link matrices is a polytope of stochastic matrices and a more general utility function is employed. The motivation for such a problem comes from having weighted links so that webmasters can determine which links should be placed in a more visible location inside their pages to increase clickings on those hyperlinks. Both discrete and continuous problems are shown to be solvable in polynomial time by modeling them as constrained Markov decision processes with ergodic rewards (see, e.g., Puterman 1994).

Summary and Future Directions

Markov chains form one of the simplest classes of stochastic processes but have been found powerful in their capability to model large-scale complex systems. In this entry, we introduced them mainly from the viewpoint of PageRank algorithms in the area of search engines and with a particular emphasis on recent works carried out based on control theoretic tools. Computational issues will remain in this area as major challenges, and further studies will be needed. As we have observed in PageRank-related problems, it is important to pay careful attention to structures of the particular problems.

Cross-References

- [Distributed Optimization](#)
- [Randomized Methods for Control of Uncertain Systems](#)

Bibliography

- Aldaheri R, Khalil H (1991) Aggregation of the policy iteration method for nearly completely decomposable Markov chains. *IEEE Trans Autom Control* 36: 178–187
- Bertsekas D, Tsitsiklis J (1989) *Parallel and distributed computation: numerical methods*. Prentice-Hall, Englewood Cliffs
- Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. *Comput Netw ISDN Syst* 30:107–117
- Csáji B, Jungers R, Blondel V (2014) PageRank optimization by edge selection. *Discrete Appl Math* 169: 73–87
- de Kerchove C, Ninove L, Van Dooren P (2008) Influence of the outlinks of a page on its PageRank. *Linear Algebra Appl* 429:1254–1276
- Fercoq O, Akian M, Bouhtou M, Gaubert S (2013) Ergodic control and polyhedral approaches to PageRank optimization. *IEEE Trans Autom Control* 58: 134–148
- Horn R, Johnson C (1985) *Matrix analysis*. Cambridge University Press, Cambridge
- Ishii H, Suzuki A (2018) Distributed randomized algorithms for PageRank computation: recent advances. In: Basar T (ed) *Uncertainty in complex networked systems*. Birkhäuser, Basel, pp 419–447
- Ishii H, Tempo R (2010) Distributed randomized algorithms for the PageRank computation. *IEEE Trans Autom Control* 55:1987–2002
- Ishii H, Tempo R, Bai EW (2012) A web aggregation approach for distributed randomized PageRank algorithms. *IEEE Trans Autom Control* 57:2703–2717
- Ishii H, Tempo R (2014) The PageRank problem, multi-agent consensus and web aggregation: a systems and control viewpoint. *IEEE Control Syst Mag* 34(3): 34–53
- Kumar P, Varaiya P (1986) *Stochastic systems: estimation, identification, and adaptive control*. Prentice Hall, Englewood Cliffs
- Lagoa C, Zaccarian L, Dabbene F (2017) A distributed algorithm with consistency for PageRank-like linear algebraic systems. In: *Proceedings of 20th IFAC World Congress*, pp 5339–5344
- Langville A, Meyer C (2006) *Google's PageRank and beyond: the science of search engine rankings*. Princeton University Press, Princeton
- Meyer C (1989) Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Rev* 31:240–272
- Norris J (1997) *Markov chains*. Cambridge University Press, Cambridge
- Phillips R, Kokotovic P (1981) A singular perturbation approach to modeling and control of Markov chains. *IEEE Trans Autom Control* 26:1087–1094
- Puterman M (1994) *Markov decision processes: discrete stochastic dynamic programming*. Wiley, New York
- You K, Tempo R, Qiu L (2017) Distributed algorithms for computation of centrality measures in complex networks. *IEEE Trans Autom Control* 62:2080–2094
- Zhao W, Chen H, Fang H (2013) Convergence of distributed randomized PageRank algorithms. *IEEE Trans Autom Control* 58:3255–3259

Mathematical Models of Marine Vehicle-Manipulator Systems

Gianluca Antonelli

University of Cassino and Southern Lazio,
Cassino, Italy

Abstract

Marine intervention requires the use of manipulators mounted on support vehicles. Such systems, defined as vehicle-manipulator systems, exhibit specific mathematical properties and require proper control design methodologies. This article briefly discusses the mathematical model within a control perspective as well as sensing and actuation peculiarities.

Keywords

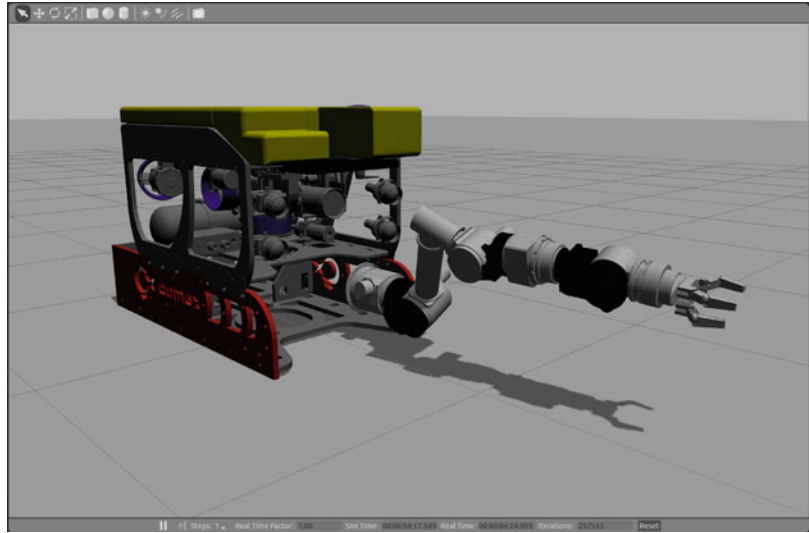
Marine robotics · Underwater robotics · Underwater intervention · Floating-base manipulators

Introduction

In case of marine operations that require interaction with the environment, an underwater vehicle is usually equipped with one or more manipulators; such systems are defined underwater vehicle-manipulator systems (UVMSs). A UVMS holding six degree-of-freedom (DOF) manipulators is illustrated in Fig. 1. The vehicle carrying the manipulator may or may not be connected to the surface; in the first case, we face a so-called remotely operated vehicle (ROV), while in the latter an autonomous underwater vehicle (AUV). ROVs, being physically linked,

Mathematical Models of Marine Vehicle-Manipulator Systems, Fig. 1

Rendering of a ROV equipped with a robotic arm, courtesy of the DexROV consortium (Birk et al. 2018)



via the tether, to an operator that can be on a submarine or on a surface ship, receives power as well as control commands. AUVs, on the other hand, are supposed to be completely autonomous, thus relying to onboard power system and intelligence.

Remotely controlled manipulation represents the state of the art in the underwater environment, while autonomous or semiautonomous operations still are in their embryonic stage. Activities in this direction are ongoing, such as, e.g., the European projects Trident (Sanz et al. 2013) and DexROV (Birk et al. 2018).

Sensory System

Any manipulation task requires that some variables are measured; those may concern the internal state of the system such as the end effector as well as the vehicle position and orientation or the velocities. Some others concern the surrounding environment as it is the case of vision systems or range measurements. The presence of the water, carrying the electromagnetic or acoustic signals, is such that the underwater sensing is characterized by poorer performance with respect to what the users are commonly used to in ground robotics.

One of the major challenges in underwater robotics is the localization due to the absence of a single, proprioceptive sensor that measures the vehicle position and the impossibility to use the Global Navigation Satellite System (GNSS) under the water. The use of redundant multi-sensory systems, thus, is common in order to perform sensor fusion and give fault detection and tolerance capabilities to the vehicle.

Localization

A possible approach for underwater vehicles localization is to rely on Inertial Navigation Systems (INSs); those are algorithms that implement dead reckoning techniques, i.e., the estimation of the position by properly merging and integrating measurements obtained with inertial and velocity sensors. Dead reckoning suffers from numerical drift due to the integration of sensor noise, as well as sensor bias and drift, and may be susceptible to the presence of external currents and model uncertainties. Since the variance of the estimation error grows with the distance traveled, this technique is only used for short dives.

Several algorithms are based on the concept of trilateration. The vehicle measures its distance with respect to known positions and prop-

erly uses this information by applying geometric-based formulas. Under the water, the technology for trilateration is not based on the electromagnetic field, due to the attenuation of its radiations, but on acoustics.

Among the commercially available solutions, long, short, and ultrashort baseline systems have found widespread use. The differences are in the baseline wavelength, the required distance among the transponders, the accuracy, and the installation cost. Acoustic underwater positioning is commercially mature, and several companies offer a variety of products.

In case of intervention, when the UVMS is close to the target, rather than the absolute position with respect to an inertial frame, it is crucial to estimate the relative position with respect to the target itself. In such a case, vision-based systems may be considered.

Actuation

Underwater vehicles are usually controlled by thrusters and/or control surfaces. Control surfaces, such as rudders and sterns, are typically used in vehicles working at cruise speed, i.e., torpedo-shaped vehicles usually used in monitoring or cable/pipeline inspection. In such vehicles a main thruster is used together with at least one rudder and one stern.

This configuration is unsuitable for UVMSs since the force/moment provided by the control surfaces is the function of the velocity, and it is null in hovering, when the manipulation is typically performed.

The relationship between the force/moment acting on the vehicle and the control input of the thrusters is highly nonlinear. It is the function of structural variables such as the density of the water, the tunnel cross-sectional area, the tunnel length, the volumetric flow rate between input and output of the thrusters, and the propeller diameter. The state of the dynamic system describing the thrusters is constituted by the propeller revolution, the speed of the fluid going into the propeller, and the input torque.

Modeling

UVMSs can be modeled as rigid bodies connected to form a serial chain; the vehicle is the floating base, while each link of the manipulator represents an additional rigid body with one DOF, typically the rotation around the corresponding joint's axis. Roughly speaking, modeling of a UVMS is the effort to represent the physical relationships of those bodies in order to measure and control its end effector, typically involved in a manipulation task.

The first step of modeling is the so-called direct kinematics, consisting in computing the position/orientation of the end effector with respect to an inertial, i.e., world fixed, frame. This is done via geometric relationship function of the system kinematic parameters, typically the lengths of the links, and the current system configuration, i.e., the vehicle position/orientation and the joint positions.

Velocities of each rigid body affect the following rigid bodies and thus the end effector. For example, a vehicle roll movement or the joint velocity is projected into a linear and angular end-effector velocity. This velocity transformation is studied by the differential kinematics. Analytic and/or geometric approaches may be used to retrieve those relationships. The study of the velocity-related equations is fundamental to understand how to balance the movement between vehicle and manipulator and, within the manipulator, how to distribute it among the joints. This topic is strictly related to differential, and inverse, kinematics for industrial robots.

The extension of Newton's second law to UVMSs leads to a number of nonlinear differential equations that link together the systems generalized forces and accelerations. With the word generalized forces, it is here intended as the forces and moments acting on the vehicle and the joint torques. Correspondingly, one is interested in the vehicle linear and angular accelerations and joint accelerations. Those equations couple together all the DOFs of the structure, e.g., a force applied to the vehicle causes acceleration also on the joints. Study of the dynamics is crucial to design the controller.

It is not possible to neglect that the bodies are moving in the water, the theory of fluidodynamics is rather complex, and it is difficult to develop a simple model for most of the hydrodynamic effects. A rigorous analysis for incompressible fluids would need to resort to the Navier-Stokes equations (distributed fluid flow). However, most of the hydrodynamic effects have no significant influence in the range of the operative velocities for UVMS intervention tasks with the exception of the added masses, the linear and quadratic damping terms, and the buoyancy.

Control

Not surprisingly, the mathematical model of UVMS shares most of the characteristics of industrial robots as well as space robots modeling. Having taken into account to the physical differences, the control problems are also similar:

- **Kinematic control.** The control problem is given in terms of motion of the end effector and needs to be transformed into the motion of the vehicle and the manipulator. This is often approached by resorting to the inverse differential kinematic algorithms. In particular, algorithms for redundant systems need to be considered since a UVMS always possess at least six DOFs. Moving a UVMS requires to handle additional variables with respect to the end effector such as the vehicle roll and pitch to preserve energy, the robot manipulability, the mechanical joint limits, or eventual directional sensing.
- **Motion control.** Low-level control algorithms are designed to allow the system tracking the desired trajectory. UVMSs are characterized by different dynamics between vehicle and manipulator, uncertainty in the model parameter knowledge, poor sensing performance, and limit cycle in the thruster model. On the other hand, the limited bandwidth of the closed-loop system allows the use of simple control approaches.
- **Interaction control.** Several applications require exchange of forces with the environment. A pure motion control algorithm is not devised for such operation and specific force control algorithms, both direct and indirect, may be necessary. Master/slave systems or haptic devices may be used on the purpose, while autonomous interaction control still is in the research phase for the marine environment.

Summary and Future Directions

This article is aimed at giving a short overview of the main mathematical and technological challenges arising with UVMSs. All the components of an underwater mission, perception, actuation, and communication with the surface, are characterized by poorer performances with respect to the current industrial or advanced robotics applications.

The underwater environment is hostile; as an example the marine current provides disturbances to be counteracted by the dynamic controller, or the sand's whirlwinds obfuscate the vision-based operations close to the sea bottom. Both tele-operated and autonomous underwater missions require a significant human effort in planning, testing, and monitoring all the operations. Fault detection and recovery policies are necessary in each step to avoid loss of expensive hardware.

Future generation of UVMSs needs to be autonomous, to percept and contextualize the environment, to react with respect to unplanned situations, and to safely reschedule the tasks of complex missions. Those characteristics are being shared by all the branches of the service robotics.

Cross-References

- [Advanced Manipulation for Underwater Sampling](#)
- [Mathematical Models of Marine Vehicle-Manipulator Systems](#)
- [Space Robotics](#)
- [Underwater Robots](#)

Recommended Reading

The book of T. Fossen (1994) is one of the first books dedicated to control problems of marine systems, both underwater and surface. The same author presents, in Fossen (2002), an updated and extended version of the topics developed in the first book and in Fossen (2011), a handbook on marine craft hydrodynamics and control. A short introductory chapter to marine robotics may be found in Antonelli et al. (2016). Robotics fundamentals are also useful and can be found in Siciliano et al. (2009). At the best of our knowledge, Antonelli (2018) is the only monograph devoted to addressing the specific problems of underwater vehicle-manipulator systems.

Bibliography

- Antonelli G (2018) Underwater robots. Springer tracts in advanced robotics, 4th edn. Springer, Heidelberg
- Antonelli G, Fossen T, Yoerger D (2016) Underwater robotics. In: Siciliano B, Khatib O (eds) Springer handbook of robotics, 2nd edn. Springer, Heidelberg
- Birk A, Doernbach T, Mueller C, Luczynski T, Gomez Chavez A, Koehnopp D, Kupcsik A, Calinon S, Tanwani AK, Antonelli G, Di Lillo P, Simetti E, Casalino G, Indiveri G, Ostuni L, Turetta A, Caffaz A, Weiss P, Gobert T, Chemisky B, Gancet J, Siedel T, Govindaraj S, Martinez X, Letier P (2018) Dexterous underwater manipulation from onshore locations: streamlining efficiencies for remotely operated underwater vehicles. *IEEE Robot Autom Mag* 25(4): 24–33
- Fossen T (1994) Guidance and control of ocean vehicles. Wiley, Chichester/New York
- Fossen T (2002) Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles. Marine Cybernetics, Trondheim
- Fossen T (2011) Handbook of marine craft hydrodynamics and motion control. Wiley, Chichester
- Sanz P, Ridao P, Oliver G, Casalino G, Petillot Y, Silvestre C, Melchiorri C, Turetta A (2013) TRIDENT an European project targeted to increase the autonomy levels for underwater intervention missions. In: 2013 OCEANS-San Diego, San Diego. IEEE
- Siciliano B, Sciavicco L, Villani L, Oriolo G (2009) Robotics: modelling, planning and control. Springer, London

Mathematical Models of Ships and Underwater Vehicles

Thor I. Fossen

Department of Engineering Cybernetics, Centre for Autonomous Marine Operations and Systems, Norwegian University of Science and Technology, Trondheim, Norway

Abstract

This entry describes the equations of motion of ships and underwater vehicles. Standard hydrodynamic models in the literature are reviewed and presented using the nonlinear robot-like vectorial notation of Fossen (1991, 1994, 2011). The matrix-vector notation is highly advantageous when designing control systems since well-known system properties such as symmetry, skew-symmetry, and positiveness can be exploited in the design.

Keywords

Autonomous underwater vehicle (AUV) · Degrees of freedom · Euler angles · Hydrodynamics · Kinematics · Kinetics · Maneuvering · Remotely operated vehicle (ROV) · Seakeeping · Ship · Underwater vehicles

Introduction

The subject of this entry is mathematical modeling of ships and underwater vehicles. With *ship* we mean “any large floating vessel capable of crossing open waters,” as opposed to a boat, which is generally a smaller craft. An *underwater vehicle* is a “small vehicle that is capable of propelling itself beneath the water surface as well as on the water’s surface.” This includes unmanned underwater vehicles (UUVs), remotely operated vehicles (ROVs), autonomous underwater vehicles (AUVs) and underwater robotic vehicles (URVs).

This entry is based on Fossen (1991, 2011), which contains a large number of standard models for ships, rigs, and underwater vehicles. There exist a large number of textbooks on mathematical modeling of ships; see Rawson and Tupper (1994), Lewandowski (2004), and Perez (2005). For underwater vehicles, see Allmendinger (1990), Sutton and Roberts (2006), Inzartsev (2009), Anotonelli (2010), and Wadoo and Kachroo (2010). Some useful references on ship hydrodynamics are Newman (1977), Faltinsen (1990), and Bertram (2012).

Degrees of Freedom

A mathematical model of marine craft is usually represented by a set of ordinary differential equations (ODEs) describing the motions in six degrees of freedom (DOF): *surge*, *sway*, *heave*, *roll*, *pitch*, and *yaw*.

Hydrodynamics

In hydrodynamics it is common to distinguish between two theories:

- **Seakeeping theory:** The motions of ships at zero or constant speed in waves are analyzed using hydrodynamic coefficients and wave forces, which depends of the wave excitation frequency and thus the hull geometry and mass distribution. For underwater vehicles operating below the wave-affected zone, the wave excitation frequency will not influence the hydrodynamic coefficients.
- **Maneuvering theory:** The ship is moving in restricted calm water – that is, in sheltered waters or in a harbor. Hence, the maneuvering model is derived for a ship moving at positive speed under a zero-frequency wave excitation assumption such that added mass and damping can be represented by constant parameters.

Seakeeping models are typically used for ocean structures and dynamically positioned vessels. Several hundred ODEs are needed to effectively represent a seakeeping model; see Fossen (2011), and Perez and Fossen (2011a, b).

The remainder of this entry assumes *maneuvering theory*, since this gives lower-order models typically suited for controller-observer design. Six ODEs are needed to describe the *kinematics*, that is, the geometrical aspects of motion while Newton-Euler's equations represent additional six ODEs describing the forces and moments causing the motion (*kinetics*).

Notation

The equations of motion are usually represented using generalized position, velocity and forces (Fossen 1991, 1994, 2011) defined by the state vectors:

$$\boldsymbol{\eta} := [x, y, z, \phi, \theta, \psi]^T \quad (1)$$

$$\boldsymbol{v} := [u, v, w, p, q, r]^T \quad (2)$$

$$\boldsymbol{\tau} := [X, Y, Z, K, M, N]^T \quad (3)$$

where $\boldsymbol{\eta}$ is the generalized position expressed in the north-east-down (NED) reference frame $\{n\}$. A body-fixed reference frame $\{b\}$ with axes:

x_b – longitudinal axis (from aft to fore)

y_b – transversal axis (to starboard)

z_b – normal axis (directed downward)

is rotating about the NED reference frame $\{n\}$ with angular velocity $\boldsymbol{\omega} = [p, q, r]^T$. The generalized velocity vector $\boldsymbol{v}$ and forces $\boldsymbol{\tau}$ are both expressed in $\{b\}$, and the 6-DOF states are defined according to SNAME (1950):

- **Surge** position x , linear velocity u , force X
- **Sway** position y , linear velocity v , force Y
- **Heave** position z , linear velocity w , force Z
- **Roll** angle ϕ , angular velocity p , moment K
- **Pitch** angle θ , angular velocity q , moment M
- **Yaw** angle ψ , angular velocity r , moment N

Kinematics

The generalized velocities $\dot{\boldsymbol{\eta}}$ and $\boldsymbol{v}$ in $\{b\}$ and $\{n\}$, respectively satisfy the following kinematic transformation (Fossen 1994, 2011):

$$\dot{\eta} = \mathbf{J}(\eta) \mathbf{v} \quad (4)$$

$$\mathbf{J}(\eta) := \begin{bmatrix} \mathbf{R}(\Theta) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{T}(\Theta) \end{bmatrix} \quad (5)$$

where $\Theta = [\phi, \theta, \psi]^\top$ is the *Euler angles* and

$$\mathbf{R}(\Theta) = \begin{bmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta s\phi & s\psi c\theta + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\theta + s\psi s\theta s\phi & c\psi c\theta + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (6)$$

with $s \cdot = \sin(\cdot)$, $c \cdot = \cos(\cdot)$ and $t \cdot = \tan(\cdot)$.

The matrix $\mathbf{R}$ is recognized as the Euler angle rotation matrix $\mathbf{R} \in SO(3)$ satisfying $\mathbf{R}\mathbf{R}^\top = \mathbf{R}^\top\mathbf{R} = \mathbf{I}$, and $\det(\mathbf{R}) = 1$, which implies that $\mathbf{R}$ is orthogonal. Consequently, the inverse rotation matrix is given by: $\mathbf{R}^{-1} = \mathbf{R}^\top$. The Euler rates $\dot{\Theta} = \mathbf{T}(\Theta)\omega$ are singular for $\theta \neq \pm\pi/2$ since:

$$\mathbf{T}(\Theta) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix}, \quad \theta \neq \pm\frac{\pi}{2} \quad (7)$$

Singularities can be avoided by using *unit quaternions* instead (Fossen 1994, 2011).

Kinetics

The rigid-body kinetics can be derived using the *Newton-Euler formulation*, which is based on *Newton's second law*. Following Fossen (1994, 2011) this gives:

$$\mathbf{M}_{RB}\dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau}_{RB} \quad (8)$$

where $\mathbf{M}_{RB}$ is the rigid-body mass matrix, $\mathbf{C}_{RB}$ is the rigid-body Coriolis and centripetal matrix due to the rotation of $\{b\}$ about the geographical frame $\{n\}$. The generalized force vector $\boldsymbol{\tau}_{RB}$ represents external forces and moments expressed in $\{b\}$. In the nonlinear case:

$$\boldsymbol{\tau}_{RB} = -\mathbf{M}_A\dot{\mathbf{v}} - \mathbf{C}_A(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} - \mathbf{g}(\eta) + \boldsymbol{\tau} \quad (9)$$

where the matrices $\mathbf{M}_A$ and $\mathbf{C}_A(\mathbf{v})$ represent hydrodynamic added mass due to acceleration $\dot{\mathbf{v}}$ and Coriolis acceleration due to the rotation of $\{b\}$ about the geographical frame $\{n\}$. The potential and viscous damping terms are lumped together into a nonlinear matrix $\mathbf{D}(\mathbf{v})$ while $\mathbf{g}(\eta)$ is a vector of generalized restoring forces. The control inputs are generalized forces given by $\boldsymbol{\tau}$.

Formulae (8) and (9) together with (4) are the fundamental equations when deriving the ship and underwater vehicle models. This is the topic for the next sections.

Ship Model

The ship equations of motion are usually represented in three DOFs by neglecting *heave*, *roll* and *pitch*. Combining (4), (8), and (9) we get:

$$\dot{\eta} = \mathbf{R}(\psi)\mathbf{v} \quad (10)$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{wave}} \quad (11)$$

where $\eta := [x, y, \psi]^\top$, $\mathbf{v} := [u, v, r]^\top$ and

$$\mathbf{R}(\psi) = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (12)$$

is the rotation matrix in yaw. It is assumed that wind and wave-induced forces $\boldsymbol{\tau}_{\text{wind}}$ and $\boldsymbol{\tau}_{\text{wave}}$ can be linearly superpositioned. The system matrices $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$ and $\mathbf{C}(\mathbf{v}) = \mathbf{C}_{RB}(\mathbf{v}) + \mathbf{C}_A(\mathbf{v})$ are usually derived under the assumption of port-starboard symmetry and that surge can be decoupled from sway and yaw (Fossen 2011). Moreover,

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix} \quad (13)$$

$$\mathbf{C}_{RB}(\mathbf{v}) = \begin{bmatrix} 0 & -mr & -mx_g r \\ mr & 0 & 0 \\ mx_g r & 0 & 0 \end{bmatrix} \quad (14)$$

$$\mathbf{C}_A(\mathbf{v}) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u \\ -Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u & 0 \end{bmatrix} \quad (15)$$

Hydrodynamic damping will, in its simplest form, be linear:

$$\mathbf{D} = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_r & -N_r \end{bmatrix} \quad (16)$$

while a nonlinear expression based on second-order modulus functions describing quadratic drag and cross-flow drag is:

$$\mathbf{D}(\mathbf{v}) = \begin{bmatrix} -X_{|u|u} |u| & 0 \\ 0 & -Y_{|v|v} |v| - Y_{|r|v} |r| \\ 0 & -N_{|v|v} |v| - N_{|r|v} |r| \\ 0 & -Y_{|v|r} |v| - Y_{|r|r} |r| \\ -N_{|v|r} |v| - N_{|r|r} |r| \end{bmatrix} \quad (17)$$

Other nonlinear representations are found in Fossen (1994, 2011).

In the case of *irrotational ocean currents*, we introduce the relative velocity vector:

$$\mathbf{v}_r = \mathbf{v} - \mathbf{v}_c$$

where $\mathbf{v}_c = [u_c^b, v_c^b, 0]^\top$ is a vector of current velocities in $\{b\}$. Hence, the kinetic model takes the form:

$$\begin{aligned} & \underbrace{\mathbf{M}_{RB} \dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v}) \mathbf{v}}_{\text{rigid-body forces}} \\ & + \underbrace{\mathbf{M}_A \dot{\mathbf{v}}_r + \mathbf{C}_A(\mathbf{v}_r) \mathbf{v}_r + \mathbf{D}(\mathbf{v}_r) \mathbf{v}_r}_{\text{hydrodynamic forces}} \\ & = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{wave}} \end{aligned}$$

This model can be simplified if the *ocean currents* are assumed to be *constant* and *irrotational* in $\{n\}$. According to Fossen (1994, 2011, Property 8.1), $\mathbf{M}_{RB} \dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v}) \mathbf{v} \equiv \mathbf{M}_{RB} \dot{\mathbf{v}}_r + \mathbf{C}_{RB}(\mathbf{v}_r) \mathbf{v}_r$ if the rigid-body Coriolis and centripetal matrix satisfies $\mathbf{C}_{RB}(\mathbf{v}_r) = \mathbf{C}_{RB}(\mathbf{v})$. One parametrization satisfying this is (14). Hence, the Coriolis and centripetal matrix satisfies $\mathbf{C}(\mathbf{v}_r) = \mathbf{C}_{RB}(\mathbf{v}_r) + \mathbf{C}_A(\mathbf{v}_r)$ and it follows that:

$$\mathbf{M} \dot{\mathbf{v}}_r + \mathbf{C}(\mathbf{v}_r) \mathbf{v}_r + \mathbf{D}(\mathbf{v}_r) \mathbf{v}_r = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{wave}} \quad (18)$$

The kinematic equation (10) can be modified to include the relative velocity $\mathbf{v}_r$ according to:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi) \mathbf{v}_r + [u_c^n, v_c^n, 0]^\top \quad (19)$$

where the ocean current velocities $u_c^n = \text{constant}$ and $v_c^n = \text{constant}$ in $\{n\}$. Notice that the body-fixed velocities $\mathbf{v}_c = \mathbf{R}(\psi)^\top [u_c^n, v_c^n, 0]^\top$ will vary with the heading angle ψ .

The maneuvering model presented in this entry is intended for controller-observer design, prediction, and computer simulations, as well as system identification and parameter estimation. A large number of application-specific models for marine craft are found in Fossen (1994, 2011, Chap. 7).

Hydrodynamic programs compute mass, inertia, potential damping and restoring forces while a more detailed treatment of viscous dissipative forces (damping) and sealoads are found in the extensive literature on hydrodynamics – see Faltinsen (1990) and Newman (1977).

Underwater Vehicle Model

The 6-DOF underwater vehicle equations of motion follow from (4), (8), and (9) under the assumption that wave-induced motions can be neglected:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta}) \mathbf{v} \quad (20)$$

$$\mathbf{M} \dot{\mathbf{v}} + \mathbf{C}(\mathbf{v}) \mathbf{v} + \mathbf{D}(\mathbf{v}) \mathbf{v} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \quad (21)$$

with generalized position $\boldsymbol{\eta} := [x, y, z, \phi, \theta, \psi]^\top$ and velocity $\mathbf{v} := [u, v, w, p, q, r]^\top$. Assume

that the gravitational force acts through the center of gravity (CG) defined by the vector $\mathbf{r}_g := [x_g, y_g, z_g]^\top$ with respect to the coordinate origin $\{b\}$. Similarly, the buoyancy force acts through the center of buoyancy (CB) defined by

the vector $\mathbf{r}_b := [x_b, y_b, z_b]^\top$. For most vehicles $y_g = y_b = 0$.

For a port-starboard symmetrical vehicle with homogenous mass distribution, CG satisfying $y_g = 0$ and products of inertia $I_{xy} = I_{yz} = 0$, the system inertia matrix becomes:

$$\mathbf{M} := \begin{bmatrix} m - X_{\dot{u}} & 0 & -X_{\dot{w}} & 0 & mz_g - X_{\dot{q}} & 0 \\ 0 & m - Y_{\dot{v}} & 0 & -mz_g - Y_{\dot{p}} & 0 & mx_g - Y_{\dot{r}} \\ -X_{\dot{w}} & 0 & m - Z_{\dot{w}} & 0 & -mx_g - Z_{\dot{q}} & 0 \\ 0 & -mz_g - Y_{\dot{p}} & 0 & I_x - K_{\dot{p}} & 0 & -I_{zx} - K_{\dot{r}} \\ mz_g - X_{\dot{q}} & 0 & -mx_g - Z_{\dot{q}} & 0 & I_y - M_{\dot{q}} & 0 \\ 0 & mx_g - Y_{\dot{r}} & 0 & -I_{zx} - K_{\dot{r}} & 0 & I_z - N_{\dot{r}} \end{bmatrix} \quad (22)$$

where the hydrodynamic derivatives are defined according to SNAME (1950). The Coriolis and centripetal matrices are:

$$\mathbf{C}_A(\mathbf{v}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & a_3 & 0 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0 & -b_3 & b_2 \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{bmatrix} \quad (23)$$

where

$$\begin{aligned} a_1 &= X_{\dot{u}}u + X_{\dot{w}}w + X_{\dot{q}}q \\ a_2 &= Y_{\dot{v}}v + Y_{\dot{p}}p + Y_{\dot{r}}r \\ a_3 &= Z_{\dot{u}}u + Z_{\dot{w}}w + Z_{\dot{q}}q \\ b_1 &= K_{\dot{v}}v + K_{\dot{p}}p + K_{\dot{r}}r \\ b_2 &= M_{\dot{u}}u + M_{\dot{w}}w + M_{\dot{q}}q \\ b_3 &= N_{\dot{v}}v + N_{\dot{p}}p + N_{\dot{r}}r \end{aligned} \quad (24)$$

and

$$\mathbf{C}_{RB}(\mathbf{v}) = \begin{bmatrix} 0 & -mr & mq & mz_g r & -mx_g q & -mx_g r \\ mr & 0 & -mp & 0 & m(z_g r + x_g p) & 0 \\ -mq & mp & 0 & -mz_g p & -mz_g q & mx_g p \\ -mz_g r & 0 & mz_g p & 0 & -I_{xz} p + I_z r & -I_y q \\ mx_g q & -m(z_g r + x_g p) & mz_g q & I_{xz} p - I_z r & 0 & -I_{xz} r + I_x p \\ mx_g r & 0 & -mx_g p & I_y q & I_{xz} r - I_x p & 0 \end{bmatrix} \quad (25)$$

Notice that this representation of $\mathbf{C}_{RB}(\mathbf{v})$ only depends on the angular velocities p , q , and r , and not the linear velocities u , v , and r . This

property will be exploited when including drift due to ocean currents.

Linear damping for a port-starboard symmetrical vehicle takes the following form:

$$\mathbf{D} = - \begin{bmatrix} X_u & 0 & X_w & 0 & X_q & 0 \\ 0 & Y_v & 0 & Y_p & 0 & Y_r \\ Z_u & 0 & Z_w & 0 & Z_q & 0 \\ 0 & K_v & 0 & K_p & 0 & K_r \\ M_u & 0 & M_w & 0 & M_q & 0 \\ 0 & N_v & 0 & N_p & 0 & N_r \end{bmatrix} \quad (26)$$

Let $W = mg$ and $B = \rho g \nabla$ denote the weight and buoyance where m is the mass of the vehicle including water in free floating space, ∇ the volume of fluid displaced by the vehicle, g the acceleration of gravity (positive downward), and ρ the water density. Hence, the generalized restoring forces for a vehicle satisfying $y_g = y_b = 0$ becomes (Fossen 1994, 2011):

$$\mathbf{g}(\boldsymbol{\eta}) = \begin{bmatrix} (W - B)s\theta \\ -(W - B)c\theta s\phi \\ -(W - B)c\theta c\phi \\ (z_g W - z_b B)c\theta s\phi \\ (z_g W - z_b B)s\theta + (x_g W - x_b B)c\theta c\phi \\ -(x_g W - x_b B)c\theta s\phi \end{bmatrix} \quad (27)$$

The expression for $\mathbf{D}$ can be extended to include nonlinear damping terms if necessary. Quadratic damping is important at higher speeds since the Coriolis and centripetal terms $\mathbf{C}(\mathbf{v})\mathbf{v}$ can destabilize the system if only linear damping is used.

In the presence of irrotational ocean currents, we can rewrite (20) and (21) in terms of relative velocity $\mathbf{v}_r = \mathbf{v} - \mathbf{v}_c$ according to:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\mathbf{v}_r + [u_c^n, v_c^n, w_c^n, 0, 0, 0]^T \quad (28)$$

$$\mathbf{M}\dot{\mathbf{v}}_r + \mathbf{C}(\mathbf{v}_r)\mathbf{v}_r + \mathbf{D}(\mathbf{v}_r)\mathbf{v}_r + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \quad (29)$$

where it is assumed that $\mathbf{C}_{RB}(\mathbf{v}_r) = \mathbf{C}_{RB}(\mathbf{v})$, which clearly is satisfied for (25). In addition, it is assumed that u_c^n , v_c^n , and w_c^n are constant. For more details see Fossen (2011).

Programs and Data

The Marine Systems Simulator (MSS) is a Matlab/Simulink library and simulator for marine craft (<http://www.marinecontrol.org>). It includes

models for ships, underwater vehicles, and floating structures.

Summary and Future Directions

This entry has presented standard models for simulation of ships and underwater vehicles. It is recommended to consult Fossen (1994, 2011) for a more detailed description of marine craft hydrodynamics.

Cross-References

- [Control of Networks of Underwater Vehicles](#)
- [Control of Ship Roll Motion](#)
- [Dynamic Positioning Control Systems for Ships and Underwater Vehicles](#)
- [Underactuated Marine Control Systems](#)

Bibliography

- Allmendinger E (ed) (1990) Submersible vehicle systems design. SNAME, Jersey City
- Anotonelli G (2010) Underwater robots: motion and force control of vehicle-manipulator systems. Springer, Berlin/New York
- Bertram V (2012) Practical ship hydrodynamics. Elsevier, Amsterdam/London
- Faltinsen O (1990) Sea loads on ships and offshore structures. Cambridge
- Fossen TI (1991) Nonlinear modelling and control of underwater vehicles. PhD thesis, Department of Engineering Cybernetic, Norwegian University of Science and Technology
- Fossen TI (1994) Guidance and control of ocean vehicles. Wiley, Chichester/New York
- Fossen TI (2011) Handbook of marine craft hydrodynamics and motion control. Wiley, Chichester/Hoboken
- Inzartsev AV (2009) Intelligent underwater vehicles. I-Tech Education and Publishing. Open access: <http://www.intechweb.org/>
- Lewandowski EM (2004) The dynamics of marine craft. World Scientific, Singapore
- MSS (2010) Marine systems simulator. Open access: <http://www.marinecontrol.org/>
- Newman JN (1977) Marine hydrodynamics. MIT, Cambridge

- Perez T (2005) Ship motion control. Springer, Berlin/London
- Perez T, Fossen TI (2011a) Practical aspects of frequency-domain identification of dynamic models of marine structures from hydrodynamic data. *Ocean Eng* 38:426–435
- Perez T, Fossen TI (2011b) Motion control of marine craft, Ch. 33. In: Levine WS (ed) *The control systems handbook: control system advanced methods*, 2nd edn. CRC, Hoboken
- Rawson KJ, Tupper EC (1994) Basic ship theory. Longman Group, Harlow/New York
- SNAME (1950) Nomenclature for treating the motion of a submerged body through a fluid. SNAME, Technical and Research Bulletin no 1–5, pp 1–15
- Sutton R, Roberts G (2006) Advances in unmanned marine vehicles. IEE control series. The Institution of Engineering and Technology, London
- Wadoo S, Kachroo P (2010) Autonomous underwater vehicles: modeling, control design and simulation. Taylor and Francis, London

Matrix Equations in Control

Vasile Sima

National Institute for Research and Development in Informatics, Bucharest, Romania

Abstract

Linear and quadratic (Riccati) matrix equations are fundamental for systems and control theory and its numerous applications. Generalized or standard Sylvester and Lyapunov equations and generalized Riccati equations for continuous- and discrete-time systems are considered. Essential applications in control are mentioned. The main solvability conditions and properties of these equations, as well as state-of-the-art solution techniques, are summarized. The continuous progress in this area paves the way for further developments and extensions to more complex control problems.

Keywords

Controllability · Gramian · Hankel singular value · Lyapunov equation · Numerical

algorithm · Observability · Performance index · Riccati equation · Schur decomposition · Stability · Stein equation · Sylvester equation

Introduction

Matrix equations are algebraic equations in which the given data and the unknown values are represented by matrices. They appear in many computational techniques in control and systems theory and its applications but also in numerous other engineering and scientific domains. The general definition above includes linear systems of equations, $AX = B$, where $A \in \mathbb{C}^{m \times n}$ and $B \in \mathbb{C}^{m \times p}$ are given complex matrices and $X \in \mathbb{C}^{n \times p}$ is the unknown matrix. Depending on A and B , such a system may have one solution, infinitely many solutions, or no solution. If $m = n$ and A is nonsingular, then $X = A^{-1}B$ is the unique solution, where A^{-1} denotes the inverse of A . If $m \geq n$, but some columns of B are not in the range space of A , there is no (exact) solution X . However, least-squares solutions minimizing $\|AX_i - B_i\|_2^2$, for each $i = 1, 2, \dots, p$, can be computed. Here, $\|\cdot\|_2$ denotes the Euclidean norm of a vector, and M_i is the i -th column of the matrix M . If $\text{rank}(A) = n$, the least-squares solution is unique. If $\text{rank}(A) < n$, the solution is not unique, and a least-squares solution of minimum norm can be defined using the Moore-Penrose generalized inverse of A . For numerical reasons, (generalized) inverses are not used in practice, but suitable decompositions of A are computed, and each X_i is obtained by solving a linear system, possibly in a least-squares sense (Golub and Van Loan 2013). These simple matrix equations will no longer be discussed in this entry, but equations in which the unknown matrix X appears several times will be considered. The most used matrix equations in control are linear matrix equations and quadratic matrix equations in X . Since most control applications use real matrices, the real case only will be dealt with in the remaining sections. Extensions to the complex case are relatively straightforward.

Matrix Equations in Control Theory and Its Applications

Linear Matrix Equations

A general form of the linear matrix equations encountered in control theory and its applications is expressed by the *generalized Sylvester equation*,

$$AXD + EXF = -H, \quad (1)$$

where A , D , E , F , and H are known matrices with compatible dimensions and X is an unknown matrix. If D and E are identity matrices (denoted I , the order being implied), then (1) is the *Sylvester equation*, named after J. J. Sylvester, who defined it in 1884. If, in addition, $F = A^T$ and H is symmetric, that is, $H = H^T$, where the superscript denotes matrix transposition, then (1) is the (*continuous-time*) *Lyapunov equation*, named after A. M. Lyapunov, who published in 1892 an inspiring paper on the stability of motion. If $D = A^T$, $E = I$, $F = -I$, and $H = H^T$, then (1) is a *discrete-time Lyapunov equation*, sometimes called *Stein equation*. By analogy, (1) with $E = I$, $F = I$, and general H may be called *discrete-time Sylvester equation*. Moreover, if $F = A^T$, $D = E^T \neq I$, and $H = H^T$, then (1) is a *generalized continuous-time Lyapunov equation*, and if $D = A^T$, $F = -E^T \neq I$, and $H = H^T$, then (1) is a *generalized discrete-time Lyapunov equation*. For convenience, the generalized Lyapunov equations are explicitly written,

$$AXE^T + EXA^T = -H, \quad (2)$$

$$AXA^T - EXE^T = -H. \quad (3)$$

Standard Lyapunov equations are obtained from (2) and (3) for $E = I$. The symmetry of Lyapunov equations and of H implies that the solution, if it exists, is also symmetric, $X = X^T$.

Equation (1) has a unique solution if and only if the matrix pencils $A - \lambda D$ and $F - \lambda E$ are regular, that is, $\det(A - \lambda E) \neq 0$, $\det(F - \lambda D) \neq 0$, for $\lambda \in \mathbb{C}$, and $\Lambda(A - \lambda E) \cap \Lambda(F + \lambda D) = \emptyset$, that is, the spectra of the pencils $A - \lambda E$ and $F + \lambda D$ are disjoint. Here, $\Lambda(M - \lambda N)$ denotes the set of eigenvalues of the matrix pencil $M - \lambda N$,

and $\emptyset$ is the empty set. Without loss of generality, it can be assumed that matrix E in (2) and (3) is nonsingular. Then, (2) has a solution if $\lambda_i + \lambda_j \neq 0$, for all $\lambda_i, \lambda_j \in \Lambda(A - \lambda E)$. Similarly, (3) has a solution if $\lambda_i \lambda_j \neq 1$. Solvability conditions for standard Lyapunov equations follow simply by setting $E = I$, and so, they are related to the spectrum of A , $\Lambda(A) := \Lambda(A - \lambda I)$.

Sylvester systems (i.e., two coupled linear matrix equations) are employed, for instance, to simultaneously block-diagonalize two square matrices. Solving Sylvester equations is also needed for block-diagonalization of a matrix or for designing Luenberger observers. Lyapunov equations are key mathematical objects in systems theory, in analysis and design of control systems, and in many applications. Solving Lyapunov equations is an essential step in balanced realization algorithms, in procedures for computing reduced order models for systems or controllers, in Newton methods for algebraic Riccati equations (AREs), or in stabilization algorithms; see, for example, Benner et al. (2005), Lancaster and Rodman (1995), Mehrmann (1991), and Sima (1996) and the references therein. Stability analyses for dynamical systems may also resort to Lyapunov equations. A comprehensive recent survey of the theory and applications of linear matrix equations in various domains is Simoncini (2016).

The standard continuous- and discrete-time Lyapunov equations,

$$AX + XA^T = -H, \quad (4)$$

$$AXA^T - X = -H, \quad (5)$$

with $H = H^T$, are associated with an autonomous linear time-invariant system, described by

$$\delta(x(t)) = Ax(t), \quad t \geq 0, \quad x(0) = x_0, \quad (6)$$

where $x(t) \in \mathbb{R}^n$ is the system state and $\delta(x(t))$ is either $dx(t)/dt$, the differential operator, or $x(t+1)$, the advance difference operator, respectively. A necessary and sufficient condition for asymptotic stability of the system (6) is that for any symmetric positive definite matrix H ,

denoted $H > 0$, there is a unique solution $X > 0$ of the Lyapunov equation (4) or (5). Moreover, if H is positive semi-definite, denoted $H \geq 0$, and $X > 0$, then all trajectories of $x(t)$ in (6) are bounded. If, in addition, the pair (A, H) is controllable, that is, $\text{rank}([A - \lambda I \ H]) = n$, for all $\lambda \in \Lambda(A)$, then the system (6) is globally asymptotically stable. Another sufficient condition for global asymptotic stability is that $H > 0$ and $X > 0$. If $H \geq 0$ and $X \not\geq 0$, then A is not stable. If $z(t)$ is a trajectory of (6), and $V(z) := z^T X z$, then $\frac{dV(z)}{dt} = -z^T H z$, in the continuous-time case, and $V(z(t+1)) - V(z(t)) = -z^T H z$, in the discrete-time case. The function $V(z)$ is a quadratic *Lyapunov function*. If $X > 0$, then $V(z) = 0$ implies $z = 0$.

A necessary solvability condition is that both A and E , for (2), or either A or E , for (3), is nonsingular. If $\Lambda(A, E) \subset \mathbb{C}_-$, where $\mathbb{C}_-$ is the open left half of the complex plane $\mathbb{C}$, in the continuous-time case, or the open unit disk centered in the origin of $\mathbb{C}$, in the discrete-time case, then (2) or (3) are *stable* Lyapunov equations. If $H \geq 0$, a stable Lyapunov equation has a unique solution $X \geq 0$ that can be expressed and computed in a factored form, $X = U^T U$, where U is a (lower or upper triangular) Cholesky factor of X (Hammarling 1982).

An important application is the computation of the Hankel singular values of a dynamical system,

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \quad x(0) = x_0, \end{aligned} \quad (7)$$

where $A, E \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and $C \in \mathbb{R}^{p \times n}$ and $u(t)$ and $y(t)$ are the inputs (controls) and measured outputs at time t . Two related Lyapunov equations are defined for this system,

$$\begin{aligned} APE^T + EPA^T &= -BB^T, \\ A^T QE + E^T QA &= -C^T C, \end{aligned} \quad (8)$$

in the continuous-time case, and

$$\begin{aligned} APA^T - EPE^T &= -BB^T, \\ A^T QA - E^T QE &= -C^T C, \end{aligned} \quad (9)$$

in the discrete-time case. The solutions P and Q of these equations are the *controllability* and *observability Gramians*, respectively, of (7). The *Hankel singular values* are the nonnegative square roots of the eigenvalues of the matrix product QP . If the system (7) is stable, then $P \geq 0$ and $Q \geq 0$, and these properties imply that $QP \geq 0$. But these theoretical results may not hold in numerical computations if the symmetry and semi-definiteness are not preserved by a solver. The recommended approach for this application, proposed in Penzl (1998), uses B and C directly, without evaluating BB^T and $C^T C$, and computes the upper triangular Choleky factors R_c and R_o of the Gramians, $P = R_c R_c^T$, $Q = R_o^T R_o$. The Hankel singular values are then obtained as the singular values of the product $R_o R_c$, which are all real nonnegative. The Gramians and the Hankel singular values are used, for instance, for finding balanced realizations and reduced order models.

Many techniques have been proposed for solving linear matrix equations. Some techniques use integrals of products of matrix exponentials or of the resolvents $(A - \lambda E)^{-1}$ and $(F - \lambda D)^{-1}$. For instance, the unique solution of a Sylvester equation is $X = \int_0^\infty e^{At} H e^{Ft} dt$, if this integral exists. However, the most used techniques for small- and medium-size equations (with orders less than a few thousands) are techniques based on *equivalence transformations*. Specifically, the pair(s) (A, E) (and (F, D) , if $F \neq I$ and $D \neq I$) is (are) reduced to generalized real Schur form(s) (Golub and Van Loan 2013) by Schur decomposition(s) with orthogonal transformations, the right-hand side H is updated accordingly, the corresponding *reduced equation* (with matrices in Schur and triangular forms) is solved, and the result is transformed back to the solution of the original equation. For instance, for (2), the computations can be summarized as follows:

1. $\tilde{A} = Q^T A Z$, $\tilde{E} = Q^T E Z$;
2. $\tilde{H} = Q^T H Q$;
3. Solve $\tilde{A} \tilde{X} \tilde{E}^T + \tilde{E} \tilde{X} \tilde{A}^T = -\tilde{H}$;
4. $X = Z \tilde{X} Z^T$.

The pair $(\tilde{A}, \tilde{E})$ is in a *generalized real Schur form* if $\tilde{A}$ is in a real Schur form, i.e., it is block upper triangular with 1×1 and 2×2 diagonal blocks, corresponding to the real and complex conjugate eigenvalues, and $\tilde{E}$ is upper triangular. The reduced Lyapunov equation at step 3 is solved by a special back substitution process. For standard Lyapunov equations, A is reduced to a real Schur form, $\tilde{A} = Q^T A Q$, and the rest of the procedure is similar, but with $Z = Q$. Many algorithmic and computational details for this case are given, for example, in Sima (1996).

If $X \in \mathbb{R}^{m \times n}$ in (1), then the transformation method requires a computational effort of $O(m^3 + n^3)$ for finding the solution. Similarly, solving Lyapunov equations of order n requires a $O(n^3)$ effort. It is worth mentioning that using the equivalence between these equations and linear systems obtained through Kronecker products, their solution would require $O(m^3 n^3)$ or $O(n^6)$ operations, respectively. But even transformation methods are too computationally demanding for large-order equations, with m , and/or n bigger than several thousands (with the current computing machines). For such equations, iterative algorithms are used for obtaining approximate solutions based on exploiting their low-rank structure and/or sparsity. See Simoncini (2016) and the references therein for more details.

Other linear matrix equations are systems of linear matrix equations involving two unknown matrices and referred to as the *generalized Sylvester system of equations* in Varga (2017), $AX + YF = -V$ and $EX + YG = -W$, or equations where the unknown matrix appears more than twice, $\sum_{i=1}^k A_i X D_i = -H$, $k > 2$; see, for instance, Simoncini (2016). Systems of linear matrix equations are encountered when simultaneously block-diagonalizing a pair of matrices.

Quadratic Matrix Equations

The most often encountered quadratic matrix equations in the control domain are the algebraic Riccati equations (AREs). This name is given after the Italian mathematician Count J. F. Riccati (1676–1754), who investigated a scalar version

of these equations. The numerical solution of AREs is an essential step in many computational methods for linear-quadratic and robust control, filtering, controller order reduction, inner-outer factorization, spectral factorization, and other applications. Let $A, E, Q \in \mathbb{R}^{n \times n}$, $B, L \in \mathbb{R}^{n \times m}$, $Q = Q^T$, and $R = R^T \in \mathbb{R}^{m \times m}$, with E assumed nonsingular. In a compact notation, the *generalized continuous-time ARE* (GCARE), with unknown $X = X^T \in \mathbb{R}^{n \times n}$, and R nonsingular, is defined by

$$A^T X E + E^T X A - \mathcal{L}(X) R^{-1} \mathcal{L}(X)^T + Q = 0, \\ \mathcal{L}(X) := L + E^T X B. \quad (10)$$

Similarly, the *generalized discrete-time ARE* (GDARE) is defined by

$$A^T X A - E^T X E - \hat{\mathcal{L}}(X) \hat{R}(X)^{-1} \hat{\mathcal{L}}(X)^T + Q = 0, \\ \hat{\mathcal{L}}(X) := L + A^T X B, \quad (11)$$

where $\hat{R}(X) := R + B^T X B$ is nonsingular.

Riccati equations can have infinitely many solutions, but in certain conditions, there is a unique solution, X_s , with special properties. Equation (10) is associated with an *optimal regulator problem* that requires to minimize, with respect to $u(t)$, $t \geq 0$, the following quadratic *performance index* in terms of the system state and control input,

$$\int_0^\infty \left(x(t)^T Q x(t) + 2x(t)^T L u(t) + u(t)^T R u(t) \right) dt, \quad (12)$$

subject to the dynamic constraint in (7), with initial condition $x(0) = x_0$. For (11), the integral is replaced by an infinite sum for $t = 0, 1, \dots$. Discrete-time AREs are important since many dynamic systems are modeled by difference equations. It is assumed that the given *weighting matrices* Q and R satisfy $Q \geq 0$, and $R > 0$, for (10), while $R \geq 0$, for (11). In practice, often Q and L are expressed as $C^T \hat{Q} C$ and $L = C^T \hat{L}$, respectively, where $C \in \mathbb{R}^{p \times n}$ is the output matrix of the system and C , $\hat{Q}$, and $\hat{L}$ are given.

By a proper selection of Q , L , and R , the closed-loop dynamics can be modified to achieve, for instance, a faster transient response.

The solution of the minimization problem for (12) is a *state-feedback control law*, $u(t) = -Kx(t)$, $t \geq 0$, where

$$K := R^{-1} \mathcal{L}(X_s)^T, \quad (13)$$

and $X_s \geq 0$ is a *stabilizing solution* of (10), that is, $\Lambda(A - BK, E) \subset \mathbb{C}_-$. The matrix K is the *optimal regulator gain*. The same results hold for the discrete-time optimal control, in which case K is given by the formula

$$K := \hat{R}(X_s)^{-1} \hat{\mathcal{L}}(X_s)^T. \quad (14)$$

Briefly speaking, these control laws achieve system stabilization and a trade-off between the regulation error and control effort.

The optimal estimation or filtering problem, for systems with Gaussian noise disturbances, can be solved as a *dual* of an optimal control problem, and its solution gives the minimum variance state estimate, based on the system input and output. Specifically, an optimal estimation problem involves the solution of a similar ARE with A and E replaced by A^T and E^T , respectively, and the *input matrix* B replaced by the transpose of the *output matrix* $C \in \mathbb{R}^{p \times n}$. The results of an optimal design are often better suited in practice than those found by other approaches. For instance, pole assignment may deliver too large gain matrices, producing high-magnitude inputs, which might not be acceptable. The H_∞ theory and robust control applications also resort to AREs, but with different coefficient matrices and optimization objective (Francis 1987).

There is a vast literature concerning the theory and numerical solution of AREs and their use for solving optimal control and estimation problems and other practical applications. Several monographs, for instance Bini et al. (2012), Lancaster and Rodman (1995), Mehrmann (1991), and Sima (1996), address various theoretical and practical results. Numerous numerical methods, either “direct” or iterative, have been proposed for solving AREs; see, for instance, Mehrmann

(1991), Sima (1996), and Van Dooren (1981). The first class includes the (generalized) Schur techniques or structure-exploiting (QR-like) methods, for instance, Benner et al. (2016). These techniques use the connection between an ARE and a structured – Hamiltonian, for GCARE, or symplectic, for GDARE – matrix or matrix pencil, of order $2n$ or $2n + m$. An orthonormal basis of the stable invariant (or deflating subspace) of this matrix (pencil) is computed using the iterative QR (QZ) algorithm (Golub and Van Loan 2013) or a structured variant. Under standard assumptions, such as stabilizability of the pair $(A - \lambda E, B)$ and detectability of the pair $(Q - LR^{-1}L^T, A - \lambda E)$, the stable subspace has dimension n , and there is a stabilizing solution X_s , $X_s \geq 0$, that can be easily computed using the subspace basis. The second class of methods has several categories, including matrix sign function techniques, Newton techniques, doubling algorithms, or recursive algorithms. There are also several software implementations, for example, in MATLAB® or in the SLICOT Library (Benner et al. 1999; Van Huffel et al. 2004).

Newton’s method is best used for iterative improvement of a solution, delivering the maximal possible accuracy when starting from a good approximate solution. Moreover, it may be preferred in implementing certain fault-tolerant or slowly varying systems, which require online controller updating; then, the previously computed ARE solution can be used for initialization. It is worth mentioning that Newton’s method has been applied for solving special classes of large-order AREs, using low-rank Cholesky factors of the solutions of the Lyapunov equations built during the iterative process. However, the specialized solvers require additional assumptions, namely, A is structured or sparse; the solution X has a small rank compared with n .

Summary and Future Directions

The main results concerning linear and quadratic matrix equations often encountered in control

theory and its applications have been briefly reviewed. General or special equations related to both continuous- and discrete-time linear time-invariant systems have been considered. The essential assumptions involved and the properties of the solutions and of the related results have been addressed. Various solution methods have been mentioned, and recommended methods have been highlighted.

Reliability, accuracy, and efficiency are essential practical issues that should be achieved by the solvers. The advances in numerical linear algebra and scientific computations together with the spectacular progress of the computing systems performance and of the associated software supported the developments of basic numerical tools for control-related applications.

Both linear and quadratic matrix equation solvers may benefit of an iterative improvement of a computed solution. This is best performed by solving a similar equation that has the residual matrix at each iteration in the right-hand side. The solution of this equation is added as a correction to the currently computed solution. Usually, two-three iterations are sufficient to achieve the limiting accuracy.

The order of the problems tractable with the current solvers for dense matrices is limited to a few thousands. Large-order problems are now, and will be, the focus of the actual and future research interest. Iterative projection methods and ADI methods continue to be investigated, based on the fact that the solution has a low rank in many applications. Such methods are used by Newton-based Riccati solvers that solve Lyapunov equations at each iteration. New developments for more general matrix equations, containing terms with a nonlinear function of X , or further improvements in solving matrix equations for periodic systems are also expected. More applications will benefit of these advances.

Cross-References

- [Basic Numerical Methods and Software for Computer Aided Control Systems Design](#)

- [Descriptor System Techniques and Software Tools](#)
- [Model Building for Control System Synthesis](#)
- [Model Order Reduction: Techniques and Tools](#)
- [Optimization-Based Control Design Techniques and Tools](#)
- [Robust Synthesis and Robustness Analysis Techniques and Tools](#)

Recommended Reading

For more details, a list of monographs and basic articles is included in the next section.

Bibliography

- Benner P, Mehrmann V, Sima V, Van Huffel S, Varga A (1999) SLICOT – A subroutine library in systems and control theory. In: Datta BN (ed) *Applied and computational control, signals, and circuits*, vol 1, chapter 10. Birkhäuser, Boston, pp 499–539
- Benner P, Mehrmann V, Sorensen D (eds) (2005) *Dimension reduction of large-scale systems. Lecture notes in computational science and engineering*, vol 45. Springer, Berlin/Heidelberg
- Benner P, Sima V, Voigt M (2016) Algorithm 961: Fortran 77 subroutines for the solution of skew-Hamiltonian/Hamiltonian eigenproblems. *ACM Trans Math Softw (TOMS)* 42(3):1–26
- Bini DA, Iannazzo B, Meini B (2012) Numerical solution of algebraic Riccati equations. SIAM, Philadelphia
- Francis BA (1987) *A course in H_∞ control theory. Lecture notes in control and information sciences*, vol 88. Springer, New York
- Golub GH, Van Loan CF (2013) *Matrix computations*, 4th edn. The Johns Hopkins University Press, Baltimore
- Hammarling SJ (1982) Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA J Numer Anal* 2(3):303–323
- Lancaster P, Rodman L (1995) *The algebraic Riccati equation*. Oxford University Press, Oxford
- Mehrmann V (1991) *The autonomous linear quadratic control problem: theory and numerical solution. Lecture notes in control and information sciences*, vol 163. Springer, Berlin
- Penzl T (1998) Numerical solution of generalized Lyapunov equations. *Adv Comput Math* 8:33–48
- Sima V (1996) *Algorithms for linear-quadratic optimization. Pure and applied mathematics: a series of monographs and textbooks*, vol 200. Marcel Dekker, Inc., New York
- Simoncini V (2016) Computational methods for linear matrix equations. *SIAM Rev* 58(3):377–441

- Van Dooren P (1981) A generalized eigenvalue approach for solving Riccati equations. *SIAM J Sci Stat Comput* 2(2):121–135
- Van Huffel S, Sima V, Varga A, Hammarling S, Delebecque F (2004) High-performance numerical software for control. *IEEE Control Syst Mag* 24(1):60–76
- Varga A (2017) Solving fault diagnosis problems: linear synthesis techniques. Springer, Berlin

of the infinite population problem is given by the fundamental MFG Hamilton-Jacobi-Bellman (HJB) and Fokker-Planck-Kolmogorov (FPK) equations which are linked by the state distribution of a generic agent, otherwise known as the system's mean field.

Mean Field Games

Peter E. Caines

Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada

Abstract

The notion of the infinite population limit of large population games where agents are realized by controlled stochastic dynamical systems is introduced. The theory of infinite population mean field games (MFGs) is then presented including the fundamental MFG equations. Proofs of the existence and uniqueness of Nash equilibrium solutions to the MFG equations are discussed, and systems possessing major agents along with the standard asymptotically negligible agents are introduced. A short bibliography is included.

Keywords

Game theory · Mean field games · Stochastic control · Stochastic systems

Definition

Mean field game (MFG) theory studies the existence of Nash equilibria, together with the individual strategies which generate them, in games involving a large number of agents modelled by controlled stochastic dynamical systems. This is achieved by exploiting the relationship between the finite and corresponding infinite limit population problems. The solution

Introduction

Large population dynamical multi-agent competitive and cooperative phenomena occur in a wide range of designed and natural settings such as communication, environmental, epidemiological, transportation, and energy systems, and they underlie much economic and financial behavior. Analysis of such systems is intractable using the finite population game theoretic methods which have been developed for multi-agent control systems (see, e.g., Basar and Ho 1974; Ho 1980; Basar and Olsder 1995; Bensoussan and Frehse 1984). The continuum population game theoretic models of economics (Aumann and Shapley 1974; Neyman 2002) are static, as, in general, are the large population models employed in network games (Altman et al. 2002) and transportation analysis (Wardrop 1952; Haurie and Marcotte 1985; Correa and Stier-Moses 2010). However, dynamical (or sequential) stochastic games were analyzed in the continuum limit in the work of Jovanovic and Rosenthal (1988) and Bergin and Bernhardt (1992), where the fundamental mean field equilibrium appears in the form of a discrete time dynamic programming equation and an evolution equation for the population state distribution.

The mean field equations for the infinite limits of dynamical games with large but finite populations of asymptotically negligible agents originated in the work of Huang et al. (2003, 2006, 2007) (where the framework was called the Nash certainty equivalence principle) and independently in that of Lasry and Lions (2006a,b, 2007), where the now standard terminology of mean field games (MFG) was introduced. Independent of both of these, the closely related notion of oblivious equilibria for large population dynamic games was introduced

by Weintraub et al. (2005) in the framework of Markov decision processes (MDP).

One of the main results of MFG theory is that in large population stochastic dynamic games individual feedback strategies exist for which any given agent will be in a Nash equilibrium with respect to the pre-computable behavior of the mass of the other agents; this holds exactly in the asymptotic limit of an infinite population and with increasing accuracy for a finite population of agents using the infinite population feedback laws as the finite population size tends to infinity, a situation which is termed an ε -Nash equilibrium. This behavior is described by the solution to the infinite population MFG equations which are fundamental to the theory; they consist of (i) a parameterized family of HJB equations (in the nonuniform parameterized agent case) and (ii) a corresponding family of pairs of McKean-Vlasov (MV) FPK PDEs, where (i) and (ii) are linked by the probability distribution of the state of a generic agent, that is to say, the mean field. For each agent these yield (i) a Nash value of the game, (ii) the best response strategy for the agent, (iii) the agent's stochastic differential equation (SDE) (i.e., the MV SDE pathwise description), and (iv) the state distribution of such an agent (i.e., the MV FPK SDE for the parameterized individual).

Dynamical Agents

In the diffusion-based models of large population games, the state evolution of a collection of N agents A_i , $1 \leq i \leq N < \infty$, is specified by a set N of controlled stochastic differential equations (SDEs) which in the important linear case take the form

$$dx_i(t) = [F_i x_i(t) + G_i u_i]dt + D_i dw_i(t),$$

$$1 \leq i \leq N,$$

where $x_i \in \mathbb{R}^n$ is the state, $u_i \in \mathbb{R}^m$ the control input, and w_i the state Wiener process of the i th agent A_i , where $\{w_i, 1 \leq i \leq N\}$ is a collection of N independent standard Wiener processes

in $\mathbb{R}^r$ independent of all mutually independent initial conditions. For simplicity, throughout this entry, all collections of system initial conditions are taken to be independent with zero mean and to have uniformly bounded second moment.

A simplified form of the general case treated in Huang et al. (2007) and Nourian and Caines (2013) is given by the following set of controlled SDEs which for each agent A_i includes state coupling with all other agents.

$$dx_i(t) = \frac{1}{N} \sum_{j=1}^N f(t, x_i(t), u_i(t), x_j(t))dt$$

$$+ \sigma dw_i(t), \quad 1 \leq i \leq N,$$

where here, for the sake of simplicity, only the uniform (non-parameterized) generic agent case is presented. The dynamics of a generic agent in the infinite population limit of this system is then described by the following controlled MV stochastic differential equation

$$dx(t) = f[x(t), u(t), \mu_t]dt + \sigma dw(t),$$

where $f[x, u, \mu_t] = \int_{\mathbb{R}^n} f(x, u, y) \mu_t(dy)$, with the initial condition measure μ_0 specified, where $\mu_t(\cdot)$ denotes the state distribution of the population at $t \in [0, T]$. The dynamics used in the analysis in Lasry and Lions (2006a, b, 2007) and Cardaliaguet (2012) are of the form $dx_i(t) = u_i(t)dt + \sigma dw_i(t)$.

The dynamical evolution of the state x_i of the i -th agent A_i in the discrete time Markov decision process (MDP)-based formulation of the so-called anonymous sequential games (Jovanovic and Rosenthal 1988; Bergin and Bernhardt 1992; Weintraub et al. 2005) is described by a Markov state transition function, or kernel, of the form $P_{t+1} := P(x_i(t+1)|x_i(t), x_{-i}(t), u_i(t), P_t)$.

Agent Performance Functions

In the basic finite population linear-quadratic diffusion case, the agent A_i , $1 \leq i \leq N$, possesses a performance, or loss, function of the form

$$J_i^N(u_i, u_{-i}) \\ = E \int_0^T \{\|x_i(t) - m_N(t)\|_Q^2 + \|u_i(t)\|_R^2\} dt,$$

where we assume the cost coupling to be of the form $m_N(t) := (\overline{x_N(t)} + \eta)$, $\eta \in \mathbb{R}^n$, where u_{-i} denotes all agents' control laws except for that of the i th agent, $\overline{x_N}$ denotes the population average state $(1/N) \sum_{i=1}^N x_i$, and here and below the expectation is taken over an underlying sample space which carries all initial conditions and Wiener processes.

For the general nonlinear case introduced in the previous section, a corresponding finite population mean field loss function is

$$J_i^N(u_i; u_{-i}) := E \int_0^T \left((1/N) \sum_{j=1}^N l(t, x_i(t), u_i(t), x_j(t)) \right) dt, \quad 1 \leq i \leq N,$$

where L is the nonlinear state cost-coupling function. Setting $l[x, u, \mu_t] = \int_{\mathbb{R}} l(x, u, y) \mu_t(dy)$, the infinite population limit of this cost function for a generic individual agent A is given by

$$J(u, \mu) := E \int_0^T l[x(t), u(t), \mu_t] dt,$$

which is the general expression for the infinite population individual performance functions appearing in Huang et al. (2006) and Nourian and Caines (2013) and which includes those of Lasry and Lions (2006a, b, 2007) and Cardaliaguet (2012). $e^{-\rho t}$ discounted costs are employed for infinite time horizon performance functions (Huang et al. 2003, 2007), while the sample path limit of the long-range average is used for ergodic MFG problems (Lasry and Lions 2006a, 2007) and in the analysis of adaptive MFG systems (Kizilkale and Caines 2013).

The Existence of Equilibria

The objective of each agent is to find strategies (i.e., control laws) which are admissible with

respect to information and other constraints and which minimize its performance function. The resulting problem is necessarily game theoretic, and consequently central results of the topic concern the existence of Nash equilibria and their properties.

The basic linear-quadratic mean field problem has an explicit solution characterizing a Nash equilibrium (see Huang et al. 2003, 2007). Consider the scalar infinite time horizon discounted case, with nonuniform parameterized agents A_θ with parameter distribution $F(\theta)$, $\theta \in \mathcal{A}$, and dynamical parameters identified as $a_\theta := F_\theta$, $b_\theta := G_\theta$, $Q := 1$, $r := R$; then the equation scheme generating the MFG equilibrium solution takes the form:

$$\rho s_\theta = \frac{ds_\theta}{dt} + a_\theta s_\theta - \frac{b_\theta^2}{r} \Pi_\theta s_\theta - x^*,$$

$$\frac{d\bar{x}_\theta}{dt} = (a_\theta - \frac{b_\theta^2}{r} \Pi_\theta) \bar{x}_\theta - \frac{b_\theta^2}{r} s_\theta, \quad 0 \leq t < \infty,$$

$$\bar{x}(t) = \int_{\mathcal{A}} \bar{x}_\theta(t) dF(\theta),$$

$$x^*(t) = \gamma(\bar{x}(t) + \eta),$$

$$\rho \Pi_\theta = 2a_\theta \Pi_\theta - \frac{b_\theta^2}{r} \Pi_\theta + 1, \quad \Pi_\theta > 0,$$

Riccati Equation

where the control action of the generic parameterized agent A_θ is given by $u_\theta^0(t) = -\frac{b_\theta}{r} (\Pi_\theta x_a(t) + s_\theta(t))$, $0 \leq t < \infty$. u_θ^0 is the optimal tracking feedback law with respect to $x^*(t)$ which is an affine function of the mean field term $\bar{x}(t)$, and the mean with respect to the parameter distribution F of the $\theta \in \mathcal{A}$ parameterized state means of the agents. Subject to conditions for the MFG scheme to have a solution, each agent is necessarily in a Nash equilibrium, within all full information causal (i.e., non-anticipative) feedback laws, with respect to the remainder of the agents when these are employing the law u^0 .

It is an important feature of the best response control law u_a^0 that its form depends only on the parametric data of the entire set of agents and, at

any instant, it is a feedback function of only the state of the agent A_a itself and the deterministic mean field dependent offset s .

For the general nonlinear case, the MFG equations on $[0, T]$ are given by the linked equations for (i) the performance function V for each agent in the continuum, (ii) the FPK for the MV-SDE

for that agent, and (iii) the specification of the best response feedback law depending on the mean field measure μ_t , with density p_t , and the agent's state $x(t)$. In the uniform agent case, these take the following form:

The Mean Field Game (MV) HJB – (MV) FPK Equations

$$\begin{aligned}
 \text{[MV-HJB]} \quad & -\frac{\partial V(t, x)}{\partial t} = \inf_{u \in U} \left\{ f[x(t), u(t), \mu_t] \frac{\partial V(t, x)}{\partial x} + l[x(t), u(t), \mu_t] \right\} + \frac{\sigma^2}{2} \frac{\partial^2 V(t, x)}{\partial x^2} \\
 & V(T, x) = 0, \quad (t, x) \in [0, T] \times \mathbb{R} \\
 \text{[MV-FPK]} \quad & \frac{\partial p_t(t, x)}{\partial t} = -\frac{\partial \{f[x(t), u(t), \mu_t] p_t(t, x)\}}{\partial x} + \frac{\sigma^2}{2} \frac{\partial^2 p_t(t, x)}{\partial x^2} \\
 \text{[MV-BR]} \quad & u(t) = \varphi(t, x(t) | \mu_t), \quad (t, x) \in [0, T] \times \mathbb{R}
 \end{aligned}$$

The general nonlinear MFG problem is approached by different routes in, for instance, (Huang et al. 2006; Nourian and Caines 2013) and in, for instance, (Lasry and Lions 2006a, b, 2007; Cardaliaguet 2012), respectively. In the former, subject to technical conditions, an iterated contraction argument in Banach space establishes the existence and uniqueness of a solution to the HJB-(MV) FPK equations, while in the latter a Schauder fixed point argument gives the existence of a solution and a dissipativity condition on the Hamiltonian yields uniqueness. The solutions to the MFG equations give the individual agents' best response control laws which necessarily yield Nash equilibria within all causal feedback laws for the infinite population problem. In Lasry and Lions (2006a, 2007), the MFG equations on the infinite time interval (i.e., the ergodic case) are obtained, while in the expository notes (Cardaliaguet 2012) the analytic properties of solutions to the HJB-FPK equations on the finite interval are analyzed using PDE methods including the theory of viscosity solutions. A third method to establish the existence and uniqueness of solutions is the probabilistic method (Carmona and Delarue 2018) by which the so-called master equation which unifies the MFGHJB and MFGFPK equations is analyzed directly.

An important result in MFG theory is that the conditions which yield the existence and uniqueness of solutions to the MFG equations together with the corresponding agents' best response (BR) strategies are sufficient for those strategies to possess the ε -Nash property (see, e.g., Huang et al. 2003, 2006, 2007; Nourian and Caines 2013; Cardaliaguet 2012). Specifically, it is shown that for any $\varepsilon > 0$, there exists a population size N_ε such that for all larger populations the use of the infinite population BR feedback laws gives each agent a value to its performance function within ε of its infinite population Nash value.

A counterintuitive feature of these results is that, asymptotically in population size, observations of the states of rival agents in MFG systems are of vanishing value to any given agent; this is in contrast to the situation in single-agent optimal control theory where the value of observations on an agent's environment is in general positive. However, when there is a major agent present in the population (Huang 2010; Nguyen and Huang 2012), state estimation plays a role, and a sequence of papers beginning with (Caines and Kizilkale 2016; Sen and Caines 2016) establishes, in the linear and nonlinear cases, respectively, the existence of equilibria when the minor agents extend their states with their individual estimates of the major agent's state.

Current Developments

Mean field game research has been rapidly expanding in terms of its exposition and in terms of both the fundamental theory and its applications. In particular the monographs by Bensoussan et al. (2013), Gomes and Saude (2014), and Kolokolstov and Malafeyav, and the two-volume monograph by Carmona and Delarue (2018), have now appeared summarizing, systematizing, and providing foundations and examples for large parts of the field. Examples of recent work include that of Bardi and Feleqi (2016) on nonlinear elliptic PDE systems and mean field games; the monograph (Gomes et al. 2016) treating the regularity of solutions; robust MFG theory with applications in (Bauso 2016); applications in algorithmic trading in finance in (Casgrain and Jaimungal 2018); applications to demand management in electrical power distribution in (Kizilkale and Malhamé 2016); and the economics of energy production (Ludkovski and Sircar 2015).

Cross-References

- [Game Theory: A General Introduction and a Historical Overview](#)

Bibliography

- Altman E, Basar T, Srikant R (2002) Nash equilibria for combined flow control and routing in networks: asymptotic behavior for a large number of users. *IEEE Trans Autom Control Spec Issue Control Issues Telecommun Netw* 47(6):917–930, June
- Aumann RJ, Shapley LS (1974) *Values of non-atomic games*. Princeton University Press, Princeton
- Bardi M, Feleqi E (2016) Nonlinear elliptic systems and mean-field games. *Nonlinear Differ Equ Appl NoDEA* 23(4):44
- Basar T, Ho YC (1974) Informational properties of the nash solutions of two stochastic nonzero-sum games. *J Econ Theory* 7:370–387
- Basar T, Olsder GJ (1995) *Dynamic noncooperative game theory*. SIAM, Philadelphia
- Bauso D (2016) Robust Mean Field Games. *Dyn Games Appl* 6:277–303
- Bensoussan A, Frehse J (1984) Nonlinear elliptic systems in stochastic game theory. *J Reine Angew Math* 350:23–67
- Bensoussan A, Frehse J, Yam P et al (2013) *Mean field games and mean field type control theory*, vol 101. Springer, New York
- Bergin J, Bernhardt D (1992) Anonymous sequential games with aggregate uncertainty. *J Math Econ* 21:543–562. North-Holland
- Caines PE, Kizilkale AC (2016) e-nash equilibria for partially observed lqg mean field games with a major player. *IEEE Trans Autom Control* 62(7):3225–3234
- Cardaliaguet P (2012) Notes on mean field games. Collège de France
- Carmona R, Delarue F (2018) *Probabilistic theory of mean field games with applications*, vols I and II. Springer International Publishing, Cham, pp 83–84
- Casgrain P, Jaimungal S (2018) Algorithmic trading with partial information: a mean field game approach. *arXiv preprint arXiv:1803.04094*
- Correa JR, Stier-Moses NE (2010) Wardrop equilibria. In: Cochran JJ (ed) *Wiley encyclopedia of operations research and management science*. Wiley, Hoboken
- Gomes DA, Saude J (2014) Mean field games models – a brief survey. *Dyn Games Appl* 4(2):110–154
- Gomes DA, Pimentel EA, Voskanyan V (2016) *Regularity Theory for Mean-Field Game Systems*. Springer, New York
- Haurie A, Marcotte P (1985) On the relationship between nash-cournot and wardrop equilibria. *Networks* 15(3):295–308
- Ho YC (1980) Team decision theory and information structures. *Proc IEEE* 68(6):15–22
- Huang MY (2010) Large-population LQG games involving a major player: the nash certainty equivalence principle. *SIAM J Control Optim* 48(5):3318–3353
- Huang MY, Caines PE, Malhamé RP (2003) Individual and mass behaviour in large population stochastic wireless power control problems: centralized and nash equilibrium solutions. In: *IEEE conference on decision and control*, HI, USA, December, pp 98–103
- Huang MY, Malhamé RP, Caines PE (2006) Large population stochastic dynamic games: closed loop Kean-Vlasov systems and the nash certainty equivalence principle. *Commun Inf Syst* 6(3):221–252
- Huang MY, Caines PE, Malhamé RP (2007) Large population cost-coupled LQG problems with non-uniform agents: individual-mass behaviour and decentralized ε – nash equilibria. *IEEE Tans Autom Control* 52(9):1560–1571. September
- Jovanovic B, Rosenthal RW (1988) Anonymous sequential games. *J Math Econ* 17(1):77–87. Elsevier, February
- Kizilkale AC, Caines PE (2013) Mean field stochastic adaptive control. *IEEE Trans Autom Control* 58(4):905–920. April
- Kizilkale AC, Malhamé RP (2016) Collective target tracking mean field control for Markovian jump-driven models of electric water heating loads. In Vamvoudakis K, Sarangapani J (eds) *Recent advances in adaptation*

- and control. Lecture notes in control and information sciences. Springer, pp 559–589
- Kolokoltsov VN, Malafeyav OA (2019) Many agent games in socio-economic systems: corruption, inspection, coalition building, network growth, security. Springer, Cham
- Lasry JM, Lions PL (2006a) Jeux à champ moyen. I – Le cas stationnaire. *Comptes Rendus Mathématique* 343(9):619–625
- Lasry JM, Lions PL (2006b) Jeux à champ moyen. II – Horizon fini et contrôle optimal. *Comptes Rendus Mathématique* 343(10):679–684
- Lasry JM, Lions PL (2007) Mean field games. *Jpn J Math* 2:229–260
- Ludkovski M, Sircar R (2015) Game theoretic models for energy production. In: Aïd R, Ludkovski M, Sircar R (eds) *Commodities, energy and environmental finance*. Springer, New York, pp 317–333
- Neyman A (2002) Values of games with infinitely many players. In: Aumann RJ, Hart S (eds) *Handbook of game theory*, vol 3. North-Holland, Amsterdam, pp 2121–2167.
- Nguyen SL, Huang M (2012) Linear-quadratic-Gaussian mixed games with continuum-parametrized minor players. *SIAM J Control Optim* 50(5):2907–2937
- Nourian M, Caines PE (2013) ε -nash mean field games theory for nonlinear stochastic dynamical systems with major and minor agents. *SIAM J Control Optim* 50(5):2907–2937
- Sen N, Caines PE (2016) Mean field game theory with a partially observed major agent. *SIAM J Control Optim* 54(6):3174–3224
- Tembine H, Zhu Q, Basar T (2012) Risk-sensitive mean field games. *arXiv preprint arXiv:1210.2806*
- Wardrop JG (1952) Some theoretical aspects of road traffic research. *Proc Inst Civ Eng II* 1:325–378
- Weintraub GY, Benkard C, Van Roy B (2005) Oblivious equilibrium: a mean field approx. for large-scale dynamic games. In: *Advances in neural information processing systems*. MIT Press

Mechanical Design and Control for Speed and Precision

William S. Nagel and Kam K. Leang
Department of Mechanical Engineering,
University of Utah, Salt Lake City, UT, USA

Abstract

This chapter discusses mechanical design techniques and control approaches that can be used to achieve high speed and fine precision

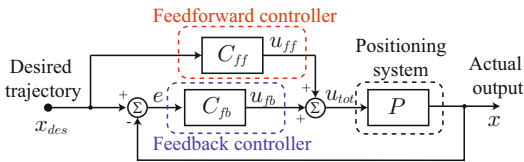
control of micro- and nanoscale positioning systems. By carefully using these design considerations, undesirable effects such as thermal drift, mechanical resonances, and off-axis motion can be minimized. Additionally, control systems can be designed to improve the positioning capabilities in open-loop and closed-loop configurations, allowing for high-bandwidth motion control while minimizing nonlinearities, such as hysteresis, and detrimental external disturbances.

Keywords

Flexure mechanism design ·
Nanopositioning · High-precision
mechatronic systems · Piezoelectric actuators

Introduction

Positioning systems have been of interest to engineers for several decades, with advances for achieving micro- and nanoscale motion occurring primarily with the advent of scanning probe microscopy (SPM) techniques in the 1980s (Binnig et al. 1982). Such improvements have allowed for the imaging of biological processes at the micron scale, precisely orienting of optic lenses, increasing information density in data storage systems, and reducing the scale of reliable manufacturing, among many other applications (Devasia et al. 2007). Positioning of tools at this scale is commonly achieved with piezoelectric ceramic actuators, but other methods including voice coil motors and custom electromagnetic actuators have been found effective as well. Measurements of output displacements can be made using strain sensors, capacitive sensors, or laser interferometers. A natural consequence of these potentially different application, actuator, and sensor combinations is to take efforts towards designing custom platforms which integrates all necessary components for particular instruments. A commonality across almost all of these applications is a desire to increase operation speed and the capable



Mechanical Design and Control for Speed and Precision, Fig. 1 General block diagram structure for precision positioning system, P , designed to track a desired trajectory, x_{des} . The total control effort, u_{tot} , is composed of the summation of feedforward, u_{ff} , and feedback, u_{fb} , inputs. The actual system output is represented by x

precision of the positioning system to expand its utility.

In practice, the speed and precision capabilities of a positioning system depend on several components, including the power electronics and data acquisition hardware. In the following, emphasis is placed on mechanical design techniques and control systems which allow for increased bandwidth and precision. A typical control system block diagram for a positioning system is illustrated in Fig. 1, where mechanical design affects the plant dynamics (P) and controller design impacts feedforward (C_{ff}) and feedback control (C_{fb}). Additionally, emphasis is being placed on piezoelectric actuators due to their ubiquity in precision positioners for the sake of brevity, but the techniques discussed herein still are practical for devices with alternative actuation methods.

Mechanical Design

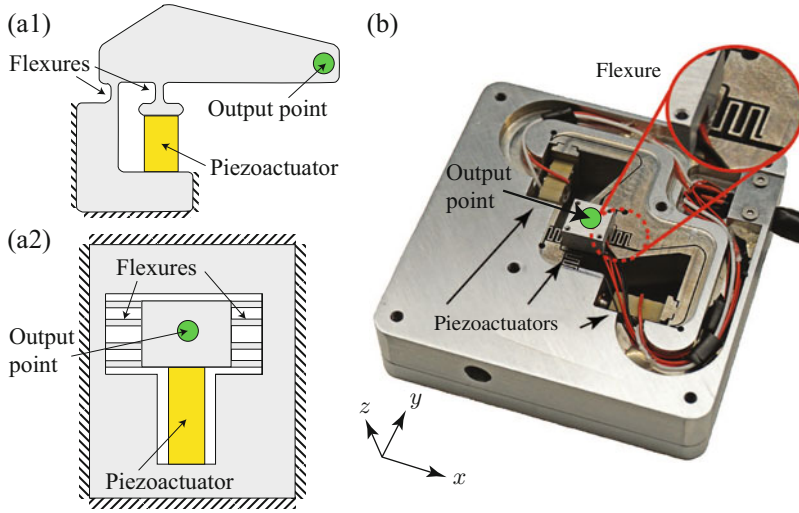
The mechanical design necessary for precision positioning consists of incorporating the selected actuators and sensors into a single mechatronic system (Fantner et al. 2006). This process can also be utilized to shape the displacement of an uncontrolled positioner such that it better matches a desired behavior. For example, monolithic flexure mechanisms, shown in Fig. 2a1–a2, can be designed to amplify the stroke of piezoactuators or guide their direction of motion to minimize off-axis effects (Fleming and Leang 2014). Even though motion can be improved with

such mechanisms, other detrimental behaviors can be present if left unaccounted for. Some examples of these undesired effects include thermal drift in actuators and mechanical components, inertial forces and interferences from resonances, and off-axis motion of the positioner's displacement (referred to as runout). Efforts can be taken to minimize or eliminate these effects through careful mechanical design.

Thermal Drift

Materials expand and contract when they experience changes in temperature. While this may be inconsequential for some macroscale applications, positioning systems that operate with nanometer precision can find such behavior significantly impactful. For example, if a piece of 6021 aluminum that is only 4.5 cm in length is used as the housing material of a positioner, it will expand slightly more than one micron for each degree Celsius increase in temperature it experiences. A potential source of temperature change is through the repeated excitation of the positioner's actuators. For example, if an actuator is cycled with a repeating trajectory (such as a sawtooth signal), the actuator and material coincident to it may begin to heat and expand. The environment in which the positioning system is contained may also introduce temperature fluctuations for the entire positioning system.

Design approaches can be taken to mitigate the thermal drift effects on a positioning platform. Materials with very low thermal deformation properties, such as titanium or Invar, can be utilized in the composition of a positioner. From the first example above, simply changing the material to Super Invar reduces the expansion to 13.5 nm for every degree Celsius increase in temperature, keeping all other dimensions the same. Operating environments can be carefully controlled or selected such that thermal fluctuations are minimized, for example, through isolation chambers which can be purchased for SPM-type instruments. The thermal expansion experienced by a positioner can also be directed based on the positioner design itself. For example, a platform with three contact points of



Mechanical Design and Control for Speed and Precision, Fig. 2 Examples of flexure mechanisms: (a1) motion-amplifying concept, (a2) motion-guiding concept,

and (b) an example positioning system with both amplifying and guiding flexure mechanisms (Nagel and Leang 2017)

constraint will experience out-of-plane bowing if all three points are fixed. By relaxing two of the constraints to be rolling contacts (supported by ball-type feet), then the device's expansion can be confined to planar motion, and the vertical drift will be substantially reduced.

Mechanical Resonances

The mechanical bandwidth of positioning systems is quantified by the first mechanical resonance in the direction of interest demonstrated by the device. However, resonant modes may be present in the housing or supporting structure of the positioner. For example, fastening hardware used to mount actuators can resonate at low and high frequencies and interfere with the output displacement. Furthermore, guiding mechanisms may introduce additional mechanical modes of oscillation before or near the primary resonance frequency of the overall positioning system. Another problem that arises from high-speed applications is due to inertial forces encountered with positioning actuators. The acceleration experienced by a sample platform displaced by an actuator increases proportionally with the frequency squared, resulting in quick increases in inertial forces that lead to excessive oscillations. The inertial force experienced on a piezoactuator

exhibiting sinusoidal motion at a frequency f is determined to be

$$F = 4\pi^2 m_{\text{eff}} \left(\frac{\Delta x}{2} \right) f^2, \quad (1)$$

where m_{eff} is the effective mass and Δx is the stroke length (Fleming and Leang 2014). These forces must be carefully managed to minimize damage to piezoelectric actuators and undesirable mechanical modes of vibration.

Flexure-guiding mechanisms are a standard design technique to mitigate the effects of mechanical resonances, as well as to increase the dominant mode of vibration for high-speed positioning. Additionally, they are employed to minimize runout. These compliant mechanisms can be approximated with spring-mass-damper models which consist of the effective inertia, stiffness, and damping of the device. For example, piezoelectric stack actuators are typically modeled as lightly damped second-order systems, and the first resonance (the natural frequency, ω_n) is defined from the equivalent mass and stiffness m_p and k_p ,

$$\omega_n = \sqrt{\frac{k_p}{m_p}}. \quad (2)$$

By introducing a guiding flexure mechanism to counteract the piezoactuator displacement, the nominal output of the actuator δ_0 is reduced through the addition of the flexure stiffness k_f added in parallel. The resultant output δ_f is then calculated to be

$$\delta_f = \delta_0 \frac{k_p}{k_p + k_f}. \quad (3)$$

Clearly, an infinitely stiff guiding flexure will result in no displacement. Although the displacement is reduced, the new equivalent stiffness can be increased greatly while the equivalent mass grows only slightly, thus increasing the first mechanical resonance frequency through Eq. (2). Applying this concept to off-axis directions, the stiffness in the off-axis directions can be increased such that their modes far exceed that of the primary actuation direction, eliminating their effects in the workable frequency range of the device.

Destructive forces experienced through high-frequency accelerations are typically addressed via preloading mechanisms. A compressive force is included on actuators in flexure-based mechanisms either through an additional loading mechanism (such as a finely threaded ball-ended screw) or by fitting the actuator into the flexure mechanism such that it naturally compresses the actuator when dormant (Yong et al. 2013). A maximum force can be approximated through Eq. (1) with known values for the actuator and target masses, as well as the maximum frequency that the device is expected to be operated at. By preloading the actuator such that it counteracts the expected inertial forces caused by the output accelerations, the forces experienced by the device's resonances can be effectively opposed.

Runout (Off-Axis Motion)

A majority of precision positioning systems will associate an input signal (e.g., voltage or current) with a desired direction of motion. Even multidirectional positioners typically have actuators oriented such that their displacements are independent of each other. However, positioning actuators will often have runout that degrade

performance. Even for the unidirectional positioner, this behavior can limit the speed capabilities of a positioning system and affect the precision of the device.

A simple solution to off-axis motion in positioners is to utilize flexure beams to guide displacements of an actuator in the direction of desired motion. Practitioners are usually satisfied if a design's dominant resonance frequency is in the desired direction of motion. In multi-axis designs, nesting actuators such that each layer is more compliant in their primary actuation direction (called a serial kinematic arrangement) allows for stiffness to be increased in the undesired directions (Kenton and Leang 2012). A consequence of such a configuration is the subsystems decreasing in speed for the outer positioners due to more effective mass being displaced. Decoupling mechanisms can be integrated into the body of a positioning system to mitigate parasitic motion for parallel-kinematic designs where actuators share the task of moving a target sample mass.

Motion Control

Once the mechanical design for a precision positioning system is established, a controller should be developed to further improve performance (Devasia et al. 2007). Choosing an adequate control architecture is essential for reducing tracking errors at higher frequencies, rejecting environmental disturbances, and mitigating nonlinear effects which may not be eliminated through mechanical design. Some examples of these nonlinearities include hysteresis (substantial in piezoelectric actuators), vibrational effects from high-order dynamics, creep, and off-axis interference for multidirectional positioners. Accounting for these issues is essential in achieving high-speed actuation while retaining high precision.

Control techniques that have been used for high-speed precision positioners can typically be categorized into two approaches, feedback and feedforward, as shown above in Fig. 1. Feedback techniques which rely on real-time sensor

information are preferable for removing errors and adding robustness to the overall system performance for precise and repeatable operation, while feedforward designs, typically based on system models, have been essential in pushing the speed capabilities of positioning platforms near or beyond the dominant mechanical resonance (Clayton et al. 2009).

High-Speed Trajectory Tracking

One main difficulty in achieving high-speed positioning is due to the lightly damped dynamics of the positioning system which are commonly present. High-frequency input signals (and even electrical noise) can excite these dominant resonance modes and cause undesired oscillations in the output position. Furthermore, because the dominant poles of the open-loop system are so close to the imaginary axis, closed-loop control can easily become unstable with higher control gains.

Despite the challenges with closed-loop stability, feedback control is the standard approach to controlling precision positioners. The standard controller is proportional-integral (PI) control, where the control input is given by

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau, \quad (4)$$

for constant proportional control gain K_p and integral control gain K_i . The tracking error is denoted by $e(t)$.

However, for lightly damped systems, this controller can be limited in bandwidth. A common approach that can improve PI control is to include a notch filter (or other inverse model-based filters) to attenuate the effects of the open-loop system resonances. By doing so, not only are these dynamics less susceptible to excitation, but the potential closed-loop bandwidth can be improved through an increased gain margin. However, small changes in the system dynamics (which can occur with changes in payload) reduce this techniques effectiveness and can lead to an unstable system. Other feedback methods can reduce the vibrational effects, including positive position feedback (PPF) and internal

resonance control (IRC). Robust controllers and techniques including sliding mode control (SMC), H_∞ , and loop-shaping have been shown to achieve high-speed closed-loop control, while repetitive errors can be reduced through learning techniques such as iterative learning control (ILC) and repetitive control (RC). In particular, the RC shown in Fig. 3 is designed to generate a discrete control effort based on a periodic error which repeats every N samples. This is done by tuning the low-pass filter

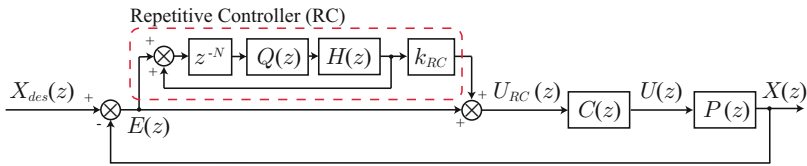
$$Q(z) = \frac{a}{z + b}, \quad (5)$$

where $|a| + |b| = 1$, along with phase-lead compensator $H(z) = z^m$, where m is a positive integer, and the RC gain, k_{RC} (Shan and Leang 2012). Such a controller is beneficial because it avoids the necessity to reset initial conditions for every cycle of the reference, which is the case for ILC.

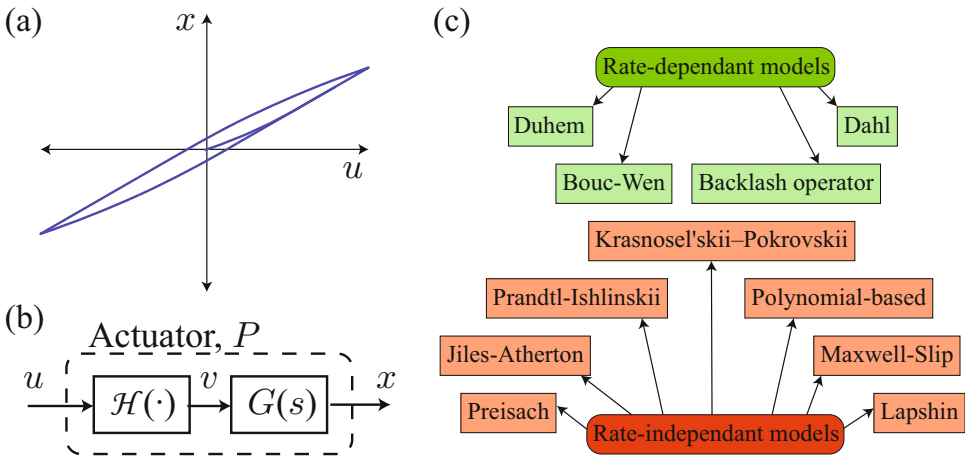
A simple feedforward method for high-speed positioning is to actuate at the same frequency as the dominant resonance. High-speed positioning is achievable through this technique, but is practically limited to sinusoidal trajectories, and any excitation frequency higher or lower will experience poor performance, consequently. More recently, techniques to compute the control input ideal for a given system based on more general desired trajectories have been developed. One such method is based on optimal inversion (Croft et al. 1999), where a feedforward control input for a desired position trajectory of an actuator can be computed off-line based on the optimal inversion of the system's dynamics. Input shaping and trajectory redesign approaches have also been demonstrated to help achieve fast positioning (Boeren et al. 2014).

Dealing with Hysteresis

Hysteresis is a common issue present in piezoelectric materials and thus is frequently encountered in precision positioning. Cycling the input voltage of a piezoelectric device does not result in a linear relationship between voltage and position, but rather all previous input information,



Mechanical Design and Control for Speed and Precision, Fig. 3 Conventional repetitive control structure, designed to compensate for periodic trajectories



Mechanical Design and Control for Speed and Precision, Fig. 4 Hysteresis modeling: (a) hysteresis loop observed from cyclic input, (b) Hammerstein-Wiener structure cascading nonlinear hysteresis effects with linear

dynamics, and (c) variety of rate-dependent and rate-independent hysteresis models used in literature, based on Gan and Zhang (2019)

or its history of inputs, dictates how much the actuator will displace. Periodic trajectories result in hysteresis loops, an example of which is shown in Fig. 4a. This effect can lead to closed-loop instability for controllers and reduce the repeatability of a positioner if left unaccounted for. It should be noted that controlling the charge of a piezoactuator circumvents hysteretic behavior, but this is less commonly implemented, and controller designs should still be considered.

A large amount of research has been dedicated to removing hysteresis from positioning devices, with both feedback and feedforward approaches being successfully implemented. Feedback techniques include proportional-integral-derivative (PID) control, the more robust control methods mentioned above, and iterative methods like ILC and RC to correct for repeating errors in periodic reference trajectories. Robust methods such as H_∞ controllers also reduce the effects of hysteresis.

Feedforward methods typically exploit some model of the hysteresis behavior, which can be modeled in a Hammerstein-Wiener model structure: a cascade combination of a nonlinear operator with the system's linear dynamics, as shown in Fig. 4b. Modeling of this hysteretic effect can be achieved either through physics-based or phenomenological models, detailed in Fig. 4c, such as polynomial-based, Preisach, Prandtl-Ishlinskii, Bouc-Wen, and Jiles-Atherton models (Yong et al. 2013; Gan and Zhang 2019). Furthermore, these models vary in their handling of the hysteresis phenomena as either a rate-dependent operator or as a nonlinear effect that is independent of the input frequency. Once the observed hysteresis is modeled, steps can then be taken to determine an effective compensation method, usually through some form of model inversion, to effectively linearize the positioning output.

Disturbance Rejection

As with any general control problem, interferences can appear in the output of a positioner, either through environmental effects or interactions with other subsystems (e.g., multi-axis coupling). Acoustic signals, structural vibrations from buildings, and electrical noise from power-line sources can all affect positioners and degrade their performance. While some of these disturbances can be mitigated through mechanical designs and deployment considerations, for example, with vibration isolation tables and acoustic isolation chambers, it can be difficult to completely eliminate their effects. Therefore controllers that compensate for known and unknown disturbances are quintessential to high-speed and precise positioning in practical environments.

Robust feedback control has been essential in removing disturbance effects, starting with standard PI controllers. Compensators that are designed using the H_∞ approach have been very successful in countering a wide variety of interferences. Sliding mode controllers have also been used for general disturbance rejection (Wang and Xu 2018). Feedforward techniques are less common because such interferences are usually unmodeled or independent of normal operation. However, in the cases where knowledge of the interference origin can be determined, feedforward methods can be used to counteract their effects. Inversion methods can be used on models of the disturbance phenomena, such as coupling in multi-axis applications (Tien et al. 2005) or acoustic sources (Yi et al. 2018). The feedforward effort can be precomputed for repetitive errors, such as through ILC (Wu et al. 2009), or implemented in real time with finite impulse response representations.

Summary and Future Directions

Flexure-guided positioning stages have been heavily explored for precision positioning applications for over several decades and have been instrumental in achieving high-speed capabilities in a variety of fields, including nanotechnology. Through the careful design of such systems,

hindering effects can be reduced or eliminated. Control techniques can be deployed to achieve fast positioning, where a variety of feedback and feedforward methods have been shown to improve the system's speed and robustness.

New positioning system designs are actively being explored to expand on their effectiveness and utility. Integrating actuation and sensing components into a single monolithic design, such as microelectromechanical systems (MEMS) devices, and composing the entire positioning device from a material serving as the actuation method have shown potential as a means for improved precision, bandwidth, and versatility in positioners (Maroufi et al. 2018). Even more, additive manufacturing with metallic materials has expanded the achievable structures which can be developed as well as reduced manufacturing time. Multiple actuators of varying ranges and speeds are also being integrated into designs so as to allow for improved bandwidth and precision without sacrificing range capabilities (Bozchalooi et al. 2016). The increase in data storage and acquisition hardware speed has allowed for more information intensive techniques for control as well, including the use of machine learning to mitigate nonlinearities (Cheng et al. 2015). These emerging techniques may prove to be beneficial to industrial implementation of high-speed and precise products.

Cross-References

- [Control for Precision Mechatronics](#)
- [Control Systems for Nanopositioning](#)
- [Experiment Design and Identification for Control](#)

Bibliography

- Binnig G, Rohrer H, Gerber C, Weibel E (1982) Surface studies by scanning tunneling microscopy. *Phys Rev Lett* 49(1):57–61

- Boeren F, Bruijnen D, van Dijk N, Oomen T (2014) Joint input shaping and feedforward for point-to-point motion: automated tuning for an industrial nanopositioning system. *Mechatronics* 24(6):572–581
- Bozchalooi IS, Houck AC, AlGhamdi J, Youcef-Toumi K (2016) Design and control of multi-actuated atomic force microscope for large-range and high-speed imaging. *Ultramicroscopy* 160:213–224
- Cheng L, Liu W, Hou Z-G, Yu J, Tan M (2015) Neural-network-based nonlinear model predictive control for piezoelectric actuators. *IEEE Trans Ind Electron* 62(12):7717–7727
- Clayton GM, Tien S, Leang KK, Zou Q, Devasia S (2009) A review of feedforward control approaches in nanopositioning for high speed SPM. *ASME J Dyn Syst Meas Control* 131(6):061101 (19 pages)
- Croft D, Stilson S, Devasia S (1999) Optimal tracking of piezo-based nanopositioners. *Nanotechnology* 10(2):201
- Devasia S, Eleftheriou E, Moheimani SOR (2007) A survey of control issues in nanopositioning. *IEEE Trans Control Syst Technol* 15(5):802–823
- Fantner GE, Schitter G, Kindt JH, Ivanov T, Ivanova K, Patel R, Holten-Anderson H, Adams J, Thurner PJ, Rangelow IW, Hansma PK (2006) Components for high speed atomic force microscopy. *Ultramicroscopy* 106:881–887
- Fleming AJ, Leang KK (2014) Design, modeling and control of nanopositioning systems. Springer, Cham
- Gan J, Zhang X (2019) A review of nonlinear hysteresis modeling and control of piezoelectric actuators. *AIP Adv* 9(4):040702
- Kenton BJ, Leang KK (2012) Design and control of a three-axis serial-kinematic high-bandwidth nanopositioner. *IEEE/ASME Trans Mechatron* 17(2):356–369
- Maroufi M, Alemansour H, Coskun MB, Moheimani SR (2018) An adjustable-stiffness MEMS force sensor: design, characterization, and control. *Mechatronics* 56:198–210
- Nagel WS, Leang KK (2017) Design of a dual-stage, three-axis hybrid parallel-serial-kinematic nanopositioner with mechanically mitigated cross-coupling. In: 2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), pp 706–711. IEEE.
- Shan Y, Leang KK (2012) Dual-stage repetitive control with Prandtl-Ishlinskii hysteresis inversion for piezo-based nanopositioning. *Mechatronics* 22:271–281
- Tien S, Zou Q, Devasia S (2005) Iterative control of dynamics-coupling-caused errors in piezoscanners during high-speed AFM operation. *IEEE Trans Control Syst Technol* 13(6):921–931
- Wang G, Xu Q (2018) Sliding mode control with disturbance rejection for piezoelectric nanopositioning control. In: 2018 Annual American Control Conference (ACC), pp 6144–6149. IEEE.
- Wu Y, Shi J, Su C, Zou Q (2009) A control approach to cross-coupling compensation of piezotube scanners in tapping-mode atomic force microscope imaging. *Rev Sci Instrum* 80(4):0433709

- Yi S, Li T, Zou Q (2018) Active control of acoustics-caused nano-vibration in atomic force microscope imaging. *Ultramicroscopy* 195:101–110
- Yong Y, Moheimani SOR, Kenton BJ, Leang KK (2013) Invited review: high-speed flexure-guided nanopositioning: mechanical design and control issues. *Rev Sci Instrum* 83:121101

Mechanism Design

Ramesh Johari

Stanford University, Stanford, CA, USA

Abstract

Mechanism design is concerned with the design of strategic environments to achieve desired outcomes at equilibria of the resulting game. We briefly overview central ideas in mechanism design. We survey both objectives the mechanism designer may seek to achieve and equilibrium concepts the designer may use to model agents. We conclude by discussing a seminal example of mechanism design at work: the Vickrey-Clarke-Groves (VCG) mechanisms.

Keywords

Game theory · Incentive compatibility · Mechanism design

Introduction

Informally, *mechanism design* might be considered “inverse game theory.” In mechanism design, a principal (the “designer”) creates a system (the “mechanism”) in which strategic agents interact with each other. Typically, the goal of the mechanism designer is to ensure that at an “equilibrium” of the resulting strategic interaction, a “desirable” outcome is achieved. Examples of mechanism design at work include the following:

1. The FCC chooses to auction spectrum among multiple competing, strategic bidders to maximize the revenue generated. How should the FCC design the auction?
2. A search engine decides to run a market for sponsored search advertising. How should the market be designed?
3. The local highway authority decides to charge tolls for certain roads to reduce congestion. How should the tolls be chosen?

In each case, the mechanism designer is shaping the incentives of participants in the system. The mechanism designer must first define the desired objective and then choose a mechanism that optimizes that objective given a prediction of how strategic agents will respond. The theory of mechanism design provides guidance in solving such optimization problems.

We provide a brief overview of some central concepts in mechanism design. In the first section, we delve into more detail on the structure of the optimization problem that a mechanism designer solves. In particular, we discuss two central features of this problem: (1) What is the objective that the mechanism designer seeks to achieve or optimize? (2) How does the mechanism designer model the agents, i.e., what equilibrium concept describes their strategic interactions? In the second section, we study a specific celebrated class of mechanisms, the Vickrey-Clarke-Groves mechanisms.

Objectives and Equilibria

A mechanism design problem requires two essential inputs, as described in the introduction. First, what is the objective the mechanism designer is trying to achieve or optimize? And second, what are the constraints within which the mechanism designer operates? On the latter question, perhaps the biggest “constraint” in mechanism design is that the agents are assumed to act rationally in response to whatever mechanism is imposed on them. In other words, the mechanism designer needs to model *how* the agents will interact with each other. Mathematically, this is modeled by

a choice of equilibrium concept. For simplicity, we focus only on *static* mechanism design, i.e., mechanism design for settings where all agents act simultaneously.

Objectives

In this section we briefly discuss three objectives the mechanism designer may choose to optimize for: *efficiency*, *revenue*, and a *fairness* criterion.

1. **Efficiency** When the mechanism designer focuses on “efficiency,” they are interested in ensuring that the equilibrium outcome of the game they create is a Pareto efficient outcome. In other words, at an equilibrium of the game, there should be no individual that can be made strictly better off while leaving all others at least as well off as they were before. The most important instantiation of the efficiency criterion arises in *quasilinear* settings, i.e., settings where the utility of all agents is measured in a common, transferable monetary unit. In this case, it can be shown that achieving efficient outcomes is equivalent to maximizing the aggregate utility of all agents in the system. See Chap. 23 in Mas-Colell et al. (1995) for more details on mechanism design for efficient outcomes.
2. **Revenue** Efficiency may be a reasonable goal for an impartial social planner; on the other hand, in many applied settings, the mechanism designer is often herself a profit-maximizing party. In these cases, it is commonly the goal of the mechanism designer to maximize her own payoff from the mechanism itself.

A common example of this scenario is in the design of *optimal auctions*. An auction is a mechanism for the sale of a good (or multiple goods) among many competing buyers. When the principal is self-interested, she may wish to choose the auction design that maximizes her revenue from sale; the celebrated paper of Myerson (1981) studies this problem in detail.
3. **Fairness** Finally, in many settings, the mechanism designer may be interested more in achieving a “fair” outcome – even if such

an outcome is potentially not Pareto efficient. Fairness is subjective, and therefore, there are many potential objectives that might be viewed as fair by the mechanism designer. One common setting where the mechanism design strives for fair outcomes is in *cost sharing*: in a canonical example, the cost of a project must be shared “fairly” among many participants. See Chap. 15 of Nisan et al. (2007) for more discussion of such mechanisms.

Equilibrium Concepts

In this section we briefly discuss a range of equilibrium concepts the mechanism designer might use to model the behavior of players. From an optimization viewpoint, mechanism design should be viewed as maximization of the designer’s objective, subject to an equilibrium constraint. The equilibrium concept used captures the mechanism designer’s judgment about how the agents can be expected to interact with each other, once the mechanism designer has fixed the mechanism. Here we briefly discuss three possible equilibrium concepts that might be used by the mechanism designer.

1. **Dominant strategies** In dominant strategy implementation, the mechanism designer assumes that agents will play a (weak or strict) dominant strategy against their competitors. This equilibrium concept is obviously quite strong, as dominant strategies may not exist in general. However, the advantage is that when the mechanism possesses dominant strategies for each player, the prediction of play is quite strong. The Vickrey-Clarke-Groves mechanisms described below are central in the theory of mechanism design with dominant strategies.
2. **Bayesian equilibrium** In a Bayesian equilibrium, agents optimize given a common prior distribution about the other agents’ preferences. In Bayesian mechanism design, the mechanism designer chooses a mechanism taking into account that the agents will play according to a Bayesian equilibrium of the resulting game. This solution concept allows

the designer to capture a lack of complete information among players but typically allows for a richer family of mechanisms than mechanism design with dominant strategies. Myerson’s work on optimal auction design is carried out in a Bayesian framework (Myerson 1981).

3. **Nash equilibrium** Finally, in a setting where the mechanism designer believes the agents will be quite knowledgeable about each other’s preferences, it may be reasonable to assume they will play a Nash equilibrium of the resulting game. Note that in this case, it is typically assumed the designer does not know the utilities of agents at the time the mechanism is chosen – even though agents *do* know their own utilities at the time the resulting game is actually played. See, e.g., Moore (1992) for an overview of mechanism design with Nash equilibrium as the solution concept.

The Vickrey-Clarke-Groves Mechanisms

In this section, we describe a seminal example of mechanism design at work: the *Vickrey-Clarke-Groves* (VCG) class of mechanisms. The key insight behind VCG mechanisms is that by structuring payment rules correctly, individuals can be incentivized to truthfully declare their utility functions to the market and in turn achieve an efficient allocation. VCG mechanisms are an example of mechanism design with dominant strategies and with the goal of welfare maximization, i.e., efficiency. The presentation here is based on the material in Chap. 5 of Berry and Johari (2011), and the reader is referred there for further discussion. See also Vickrey (1961), Clarke (1971), and Groves (1973) for the original papers discussing this class of mechanisms.

To illustrate the principle behind VCG mechanisms, consider a simple example where we allocate a single resource of unit capacity among R competing users. Each user’s utility is measured in terms of a common currency unit; in particular, if the allocated amount is x_r and the payment to user r is t_r , then her utility is $U_r(x_r) + t_r$; we

refer to U_r as the *valuation* function, and let the space of valuation functions be denoted by $\mathcal{U}$. For simplicity we assume the valuation functions are continuous. In line with our discussion of efficiency above, it can be shown that the Pareto efficient allocation is obtained by solving the following:

$$\text{maximize } \sum_r U_r(x_r) \quad (1)$$

$$\text{subject to } \sum_r x_r \leq 1; \quad (2)$$

$$\mathbf{x} \geq 0. \quad (3)$$

However, achieving the efficient allocation requires knowledge of the utility functions; what can we do if these are unknown? The key insight is to make each user act *as if* they are optimizing the aggregate utility, by structuring payments appropriately. The basic approach in a VCG mechanism is to let the strategy space of each user r be the set $\mathcal{U}$ of possible valuation functions and make a payment t_r to user r so that her net payoff has the same form as the social objective (1). In particular, note that if user r receives an allocation x_r and a payment t_r , the payoff to user r is

$$U_r(x_r) + t_r.$$

On the other hand, the social objective (1) can be written as

$$U_r(x_r) + \sum_{s \neq r} U_s(x_s).$$

Comparing the preceding two expressions, the most natural means to align user objectives with the social planner's objectives is to *define the payment t_r as the sum of the valuations of all users other than r .*

A VCG mechanism first asks each user to declare a valuation function. For each r , we use $\hat{U}_r$ to denote the declared valuation function of user r and use $\hat{\mathbf{U}} = (\hat{U}_1, \dots, \hat{U}_R)$ to denote the vector of declared valuations. Formally, given a vector of declared valuation functions $\hat{\mathbf{U}}$, a VCG mechanism chooses the allocation $\mathbf{x}(\hat{\mathbf{U}})$ as an optimal solution to (1), (2), and (3) given $\hat{\mathbf{U}}$, i.e.,

$$\mathbf{x}(\hat{\mathbf{U}}) \in \arg \max_{\mathbf{x} \geq 0: \sum_r x_r \leq 1} \sum_r \hat{U}_r(x_r). \quad (4)$$

The payments are then structured so that

$$t_r(\hat{\mathbf{U}}) = \sum_{s \neq r} \hat{U}_s(x_s(\hat{\mathbf{U}})) + h_r(\hat{\mathbf{U}}_{-r}). \quad (5)$$

Here h_r is an arbitrary function of the declared valuation functions of users other than r , and various definitions of h_r give rise to variants of the VCG mechanisms. Since user r cannot affect this term through the choice of $\hat{U}_r$, she chooses $\hat{U}_r$ to maximize

$$U_r(x_r(\hat{\mathbf{U}})) + \sum_{s \neq r} \hat{U}_s(x_s(\hat{\mathbf{U}})).$$

Now note that given $\hat{\mathbf{U}}_{-r}$, the above expression is bounded above by

$$\max_{\mathbf{x} \geq 0: \sum_r x_r \leq 1} \left[U_r(x_r) + \sum_{s \neq r} \hat{U}_s(x_s) \right].$$

But since $\mathbf{x}(\hat{\mathbf{U}})$ satisfies (4), user r can achieve the preceding maximum by truthfully declaring $\hat{U}_r = U_r$. Since this optimal strategy does not depend on the valuation functions ($\hat{U}_s, s \neq r$) declared by the other users, we recover the important fact that in a VCG mechanism, *truthful declaration is a weak dominant strategy for user r .*

For our purposes, the interesting feature of the VCG mechanism is that it elicits the true utilities from the users and in turn (because of the definition of $\mathbf{x}(\hat{\mathbf{U}})$) chooses an efficient allocation. The feature that truthfulness is a dominant strategy is known as *incentive compatibility*: the individual incentives of users are aligned, or “compatible,” with overall efficiency of the system. The VCG mechanism achieves this by effectively paying each agent to tell the truth. The significance of the approach is that this payment can be properly structured even if the resource manager does not have prior knowledge of the true valuation functions.

Summary and Future Directions

Mechanism design provides an overarching framework for the “engineering” of economic systems. However, many significant challenges remain. First, VCG mechanisms are not computationally tractable in complex settings, e.g., combinatorial auctions; finding computationally tractable yet efficient mechanisms is a very active area of current research. Second, VCG mechanisms optimize for overall welfare, rather than revenue, and finding simple mechanisms that maximize revenue also presents new challenges. Finally, we have only considered implementation in static environments. Most practical mechanism design settings are dynamic. Dynamic mechanism design remains an active area of fruitful research.

Cross-References

- [Auctions](#)
- [Dynamic Noncooperative Games](#)
- [Game Theory: A General Introduction and a Historical Overview](#)

Bibliography

- Berry RA, Johari R (2011) Economic modeling in networking: a primer. *Found Trends Netw* 6(3): 165–286
- Clarke EH (1971) Multipart pricing of public goods. *Public Choice* 11(1):17–33
- Groves T (1973) Incentives in teams. *Econometrica* 41(4):617–631
- Mas-Colell A, Whinston M, Green J (1995) *Microeconomic theory*. Oxford University Press, New York
- Moore J (1992) Implementation, contracts, and renegotiation in environments with complete information. *Adv Econ Theory* 1:182–281
- Myerson RB (1981) Optimal auction design. *Math Oper Res* 6(1):58–73
- Nisan N, Roughgarden T, Tardos E, Vazirani V (eds) (2007) *Algorithmic game theory*. Cambridge University Press, Cambridge/New York
- Vickrey W (1961) Counterspeculation, auctions, and competitive sealed tenders. *J Financ* 16(1):8–37

Medical Cyber-Physical Systems: Challenges and Future Directions

Insup Lee, Oleg Sokolsky, and James Weimer
University of Pennsylvania, Philadelphia, PA,
USA

Abstract

This entry considers the area of medical cyber-physical systems, which recently has enjoyed much research and development efforts. We present a brief summary of recent developments, followed by a discussion of remaining challenges and directions for further development.

Keywords

Medical cyber-physical systems · Physiology modeling · Closed-loop treatments · Safety assurance

Introduction

One of the main characteristics of clinical practice is that it traditionally relies primarily on clinician’s expertise and direct interaction with the patient. Clinicians interpret observations from multiple sensors to evaluate the state of patient’s health and then decide on adjustments in treatment. This mode of operation has not fundamentally changed in recent times. What has changed is the workload of a typical clinician. Doctors and nurses are overworked. Moreover, recent improvement in technologies has given us new sensors that provide additional sources of information on the patient, all of which needs to be taken into account, further adding to the workload. This has raised concerns that clinician’s overload and “alarm fatigue” may result in reduced personal attention to each patient, potentially impacting the quality of decisions and treatment outcomes.

The key idea behind medical cyber-physical systems (MCPS) is that they can address these

concerns by providing computer-controlled automation. Such automation can monitor patient state by fusing data from multiple sensors and use this information in two complementary ways, as illustrated in Fig. 1. On the one hand, the system can leverage a model of patient's physiology to provide automatic closed-loop control of the treatment process, which does not require clinician's intervention most of the time. On the other hand, data-driven and machine learning techniques, needed for personalization of MCPS, require new approaches to safety and fairness verification. Note that the clinician normally has additional sources of information about the patient and can intervene with the treatment directly.

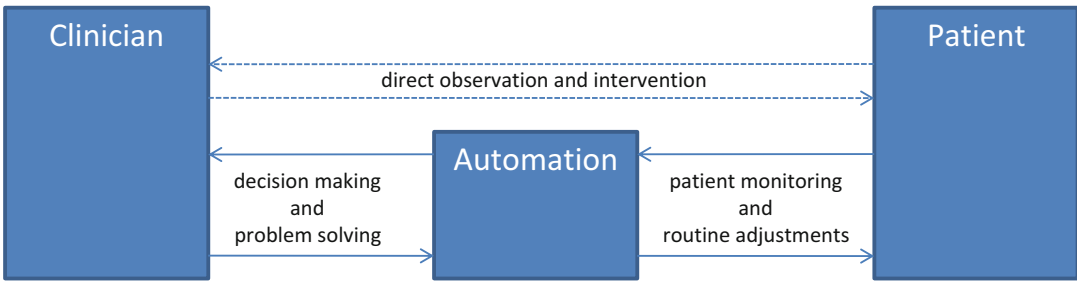
For MCPS to have a significant positive impact on clinician's workload and treatment outcomes, MCPS developers must overcome a variety of challenges. These challenges related to constructing models of patient physiology that are suitable for both closed-loop control and decision support; anomaly detection techniques that alert clinicians when human intervention is necessary; and safety assurance for the automation in order to prevent potential harm to the patient. In addition, modern MCPS are complex networked systems that can be affected by malicious external actors, both at the cyber and physical levels. Therefore, ensuring security and privacy is of paramount importance.

For the rest of the paper, we give a brief account of the state of the art in MCPS, followed by a discussion of the above challenges and future research directions.

State of the Art

Much of the work on closed-loop MCPS has concentrated on several conditions where patient's physiology is well understood and treatment effects are relatively isolated. Examples of closed-loop medical devices include automated drug infusion systems for analgesia (Derlien 1986; Liu and Northrop 1990; Cortes et al. 2008); control of blood pressure (Mason et al. 1985; Ruiz et al. 1993); control of blood

oxygenation (Iobbi et al. 2007; Morozoff and Smyth 2009) (and the related problem of control of sleep apnea (Yamashiro 2007)); control of epilepsy (Iasemidis et al. 2009; Salam et al. 2012) (and the related problem of anticipating and detecting seizures (Waterhouse 2003; Verma et al. 2011)); control of Parkinson's disease (Lawo and Herzog 2011); decision support for inpatient management of hyperglycemia (Davidson et al. 2005; Cochran et al. 2006; Juneja et al. 2007; Saager et al. 2008; Chase et al. 2011); telemedicine and decision support for treatment of insulin-dependent diabetes (Albisser 1989; Bott et al. 2007; Garcia-Saez et al. 2009); automated control of insulin-dependent diabetes (Patek et al. 2012; Cobelli et al. 2012; Breton et al. 2012; Percival et al. 2011; Steil et al. 2006; Weinzimer et al. 2008; Bruttomesso et al. 2009; Kovatchev et al. 2010; Hovorka et al. 2010; Dassau et al. 2010; Palerm 2011; Zisser et al. 2011; Atlas et al. 2010); pacemakers and other cardiac medical devices (Aseltine et al. 1979; Lai et al. 2009; Arzbacher et al. 2010; Jiang et al. 2011a,b; Shoaib et al. 2011; Jiang and Pajic 2012; Nirjon et al. 2012; Slepian et al. 2013); control of intracranial pressure (Rekate 1980); rehabilitation (Petrofsky et al. 1984; Lee et al. 1995; Stauffer et al. 2008; Pennycott et al. 2010; Nikitzuk et al. 2010); neural interfaces and prosthetics (Wu et al. 2004; Hatsopoulos et al. 2005; Weber et al. 2006; Marzullo et al. 2010; Emborg et al. 2011; Frankel et al. 2011; Simeral et al. 2011; Stanslaski et al. 2012; Witcher and Ellis 2012); force feedback in exoskeletal prostheses (Worsnopp et al. 2007; Wang et al. 2011); medical image acquisition (Harders and Szekely 2003); mechanical ventilation (Favre et al. 2003); and robotic surgical tools (Smith et al. 1991; Wyman et al. 1992; Sung and Gill 2001; Guodong et al. 2007; Penning et al. 2011; Turchetti et al. 2012; Hansen et al. 2012). From this long list of applications, it is clear that the medical establishment perceives a clear need for advanced treatment options. While the above MCPS address specific clinical needs, there remain several challenges to the design and analysis of safe, effective, and usable closed-loop MCPS.



Medical Cyber-Physical Systems: Challenges and Future Directions, Fig. 1 A typical MCPS involves two nested control loops. The automation may handle routine

treatment. The clinician would provide adjustments in response to anomalies in the treatment process, based on decision support provided by the automation

Challenges

Modeling patient physiology Design of closed-loop control typically requires a plant model. In the MCPS case, the “plant” is the patient being treated. Majority of today’s patient models are built in one of the two ways. Physiology-based models are credible, accurate, and generalizable, but too complex for real-time computation or for efficient design of closed-loop controllers. Specifically, they are often not amenable to existing techniques for model identification and parameter estimation. By contrast, data-based models are phenomenological black-box objects. They are simple and amenable to real-time computations and closed-loop control design, but cannot be used to explain what is happening with the patient. While simplicity and fidelity are important for control design, physiological transparency is critical for maintaining situational awareness by the caregiver, who should be able to tell when the treatment is not progressing as planned and intervene in an emergency. Therefore, finding the right balance between simplicity, fidelity, and physiological transparency is critical for rigorous design of closed-loop MCPS.

Understanding human-automation interactions Evaluation of safety and effectiveness of a closed-loop MCPS is impossible without a thorough understanding of how it will be used and how the operator – clinician or patient – will interact with the automation. Such interaction

takes the form of supervisory control, e.g., when a clinician changes the infusion pump rate, or the form of additional inputs to the controller that cannot be supplied by sensors, e.g., type 1 diabetes patients are typically expected to enter information about their meals. However, automation that requests frequent interactions (via alarms) with human operators, over time, can erode trust of operators in the system’s decision support and cause them to ignore recommendations. Many existing threshold-based bedside alarm systems are known to generate the large numbers of false positives, which has created the “alarm fatigue” problem, with clinicians ignoring alarms or completely shutting them off (Clark et al. 2007; Koppel et al. 2005; Weingart et al. 2003). Thus, automation needs to consider the quality of information provided by clinicians to produce feedback (e.g., alarms) that are highly specific and actionable. In doing so, automation can supply information to the operator to help make supervisory decisions, closing the outer control loop of the MCPS that directly includes the operator. User inputs such as meal intake records are often inaccurate and incomplete, and operators may misinterpret decision support information provided by the controller. Accurately capturing these human-automation interactions is key to the design and analysis of closed-loop MCPS.

Formal design of closed-loop systems with humans-in-the-loop Existing control design techniques are difficult to apply in the setting where both patient and operator exhibit complex

behaviors. There are several interrelated challenges that require us to develop new control design methods. First, limited observability of patient's physiology requires controllers to incorporate reliable detection of relevant physiological events based on available indirect observations. Second, control algorithms should allow personalization, which accounts for individual variation of physiology in different patients, as well as for different preferences for the levels of automation in operators. Third, the closed-loop system must be fair, in the sense that control performance minimizes nonclinical bias (e.g., race, gender, socioeconomic status). Finally, the physiological processes being controlled by MCPS are often distributed, taking the form of multiple hierarchical control loops.

Safety and fairness assurance Patient safety and fairness is one of the primary concerns in regulatory approval of MCPS. While the current generation MCPS primarily target a single physiological process, closed-loop MCPS targeting multiple physiological processes will need radical advances in safety analysis techniques due to inter/intra-patient variability and diversity of physiological responses. On the one hand, complexity of patient's physiological models that include complex parameter spaces will make most existing verification techniques impractical. On the other hand, as data-driven and machine learning techniques, that are needed for personalization of MCPS, require new approaches to safety and fairness verification. Furthermore, data-driven techniques need performance guarantees in diverse patient populations.

Security and privacy The safety of MCPS limits the ability to frequently update and patch MCPS. At the same time, many existing techniques for access control are impractical in the clinical setting and lead to work-arounds. Serious vulnerabilities have been found in many existing devices such as ICDs (Halperin et al. 2008) and insulin pumps (Radcliffe 2011; Advisory 2016; Raghunathan and Jha 2011; Paul et al. 2011). O'Keeffe et al. discuss vulnerabilities in AP systems in clinical trials (O'Keeffe et al. 2015). These vulnerabilities arise from failure to

implement well-known cryptographic solutions that leave the devices open to remote "hijacks," spoofing, or replay attacks. Recent attacks on insulin pumps suggest that manufacturers have been relatively slow in securing their systems (Advisory 2016). Solutions for low power encryption have been proposed (Beck et al. 2011), including for glucose sensors (Guan et al. 2011). Additionally, detailed threat assessments are needed to ensure that the system as a whole is secure and private (Kotz 2011).

Future Research Directions

The MCPS research community has been actively studying ways to address the challenges described above. Below, we discuss some of the open problems for MCPS.

Physiological modeling Common approaches to physiological modeling are based either on *maximal models*, first-principle models that accurately represent the underlying physiology (Cobelli and Carson 2008), or *minimal models*, simplified, phenomenological black-box models often based on compartmentalized techniques (Carson and Cobelli 2013). Maximal models are often unsuitable for data-driven modeling needed for personalization, or for existing control design approaches, due to their extreme complexity (Jacquez 1996; Godfrey 1983; Cobelli et al. 2009). By contrast, minimal models tend to be simpler and are suitable for data-driven modeling and control design. However, brute-force data-driven modeling techniques are often insensitive to interactions with the operator (e.g., manually delivered medications, equipment adjustments) and patient changes that are not recorded by sensors (e.g., getting out of bed, eating, exercising, etc.). Furthermore, data-driven minimal models tend to lack physiological transparency and are hard to be used for feedback to the operator. It may be possible to combine the benefits of the two modeling approaches. For example, in (Bighamian et al. 2018), the system model combines a control-theoretic model with a simple physics-based model and a phenomenological

model, balancing the overall simplicity of the model with physiological transparency.

Autonomous medical systems Most control algorithms for closed-loop MCPS referenced above involve single-input, single-output control loops and are designed using black-box data-driven models or models with locally linearized nonlinearities. Similar to autonomous vehicles, autonomous MCPS require human supervision. Existing work does not systematically address the hard problems of extreme variability of patient physiology and clinical interpretability, where, due to the severe observability constraints inherent in many MCPS, accurate adaptation to the complete patient physiology is impractical. In these scenarios, constraining adaptation by identifying and utilizing physiological invariants can enable safe and effective adaptation (Roederer et al. 2015; Weimer et al. 2016, 2017; Ivanov et al. 2015). While there has been recent work on incorporating physical invariants in vehicular systems, these technologies have not yet been applied to the medical domain.

Safety assurance and certification Formal methods have been used to verify MCPS in specific applications in limited scope (Jiang et al. 2012; Pajic et al. 2014; Abbas et al. 2016). Beyond these examples, there are emerging techniques for addressing complex physiological models (Chen et al. 2017) and modeling behavior of patients and clinicians (Chen et al. 2015). However, a methodology for safety assurance and certification in MCPS that integrates physiology and patient/operator behavior remains an active area of research.

Human factors in MCPS Given the central roles that humans play in closed-loop MCPS, interaction between humans and automation are central to the success of MCPS. While there is much work on human-automation interaction in other domains (e.g., aviation), in the health-care domain, this work has been mostly limited to the sociological context. A conceptual framework for clinicians interacting with AI-based automation has been laid out in Dermody and Fritz (2019). At the same time, security

and privacy of patient data remains an important topic to address in such systems. It is well-known that clinicians resort to a variety of work-arounds when they feel that automation prevents them from doing their work (Blythe et al. 2013; Koppel et al. 2008). Work-arounds can interfere with safety protections built into the automation system. For example, medication bar codes can be scanned after being detached from the medication container, allowing wrong medication to be administered (Koppel et al. 2008). Understanding the effects of human-automation interactions and corresponding work-arounds is essential for assuring safety and security of a system. However, such unanticipated interactions between operators and automation in clinical scenarios have not so far been considered in a rigorous modeling context.

Summary

MCPS is an actively evolving area of CPS research and development. It has a potential to enormously improve treatment procedures and health outcomes for many patients. However, complexity and diversity of human physiology, inherent roles of caregivers, technological resource constraints, as well as paramount safety, security, and privacy concerns make MCPS very challenging to build. In this entry, we discussed some of the more important challenges and promising research directions.

Cross-References

- [Cyber-Physical-Human Systems](#)
- [Cyber-Physical Security](#)
- [Industrial Cyber-Physical Systems](#)

Acknowledgments This work was supported in part by NSF-1915398, NIH R01EB029363, and NIH R01EB029767.

Bibliography

- Abbas H, Jiang KJ, Jiang Z, Mangharam R (2016) In: Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control

- Advisory R (2016) R7-2016-07: Multiple vulnerabilities in animas onetouch ping insulin pump. Cf. <https://community.rapid7.com/community/infosec/blog/2016/10/04/r7-2016-07-multiple-vulnerabilities-in-animas-onetouch-ping-insulin-pump>
- Albisser AM (1989) CRC Crit Rev Biomed Eng 17(1):1
- Arzbaeher R, Hampton DR, Burke MC, Garrett MC (2010) J Electrocardiol 43(6):601. <https://doi.org/10.1016/j.jelectrocard.2010.05.017>
- Aseltine RG, Feldman CL, Paraskos JA, Moruzzi RL (1979) IEEE Trans Biomed Eng BME-26(8):456
- Atlas E, Nimri R, Miller S, Grunberg EA, Phillip M (2010) Diabetes Care 33:1072
- Beck C, Masny D, Geiselmann W, Bretthauer G (2011) In: Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies, ISABEL '11. ACM, New York, pp 62:1–62:5. <https://doi.org/10.1145/2093698.2093760>
- Bighamian R, Parvinian B, Scully CG, Kramer G, Hahn JO (2018) Control Eng Pract 73:149
- Blythe J, Koppel R, Smith SW (2013) IEEE Secur Priv 11(5):80
- Bott OJ, Hoffmann I, Bergmann J, Gusew N, Schnell O, Gomez EJ, Hernando ME, Kosche P, von Ahn C, Mattfeld DC, Pretschner DP (2007) Int J Med Inform 76(3):447–55
- Breton M, Farret A, Bruttomesso D, Anderson S, Magni L, Patek S, Dalla Man C, Place J, Demartini S, Del Favero S, Toffanin C, Hughes-Karvetski C, Dassau E, Zisser H, Doyle 3rd F, Nicolao GD, Avogaro A, Cobelli C, Renard E, Kovatchev B (2012) Diabetes 61(9):2230
- Bruttomesso D, Farret A, Costa S, Marescoti MC, Vettore M, Avogaro A, Tiengo A, Dalla Man C, Place J, Facchinetti A, Guerra S, Magni L, De Nicolao G, Cobelli C, Renard E, Maran A (2009) J Diab Sci and Tech 3:1014
- Carson E, Cobelli C (2013) Modeling methodology for physiology and medicine. Newnes
- Chase J, Le Compte A, Preiser J, Shaw G, Penning S, Desai T (2011) Ann Intensive Care 1(1):1
- Chen S, Feng L, Rickels MR, Peleckis A, Sokolsky O, Lee I (2015) In: 2015 International Conference on Healthcare Informatics. IEEE, pp 213–222
- Chen X, Dutta S, Sankaranarayanan S (2017) In: Workshop on Applied Verification of Hybrid Systems (ARCH), p 16
- Clark T, David Y, Baretich M, Bauld T (2007) J Clin Eng 32(1):22
- Cobelli C, Carson E (2008) Introduction to modeling in physiology and medicine. Academic Press
- Cobelli C, Dalla Man C, Sparacino G, Magni L, De Nicolao G, Kovatchev BP (2009) IEEE Rev Biomed Eng 2:54
- Cobelli C, Renard E, Kovatchev BP, Keith-Hynes P, Ben Brahim N, Place J, Favero SD, Breton M, Farret A, Bruttomesso D, Dassau E, Doyle III FJ, Patek SD, Avogaro A (2012) Diabetes Care 35(9):e65
- Cochran S, Miller E, Dunn K, Burgess W, Miles W, Lobdell K (2006) Crit Care Med 34(12):A68
- Cortes PA, Krishnan SM, Lee I, Goldman JM (2008) In: 2007 Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability – HCMDSS-MD PnP '07, pp 149–150
- Dassau E, Zisser H, Percival M, Grosman B, Jovanovic L, Doyle III FJ (2010) Diabetes 59(Supl 1):A94. Proc Amer Dia Assoc Meeting
- Davidson P, Steed R, Bode B (2005) Diabetes Care 28(10):2418
- Derlien ML (1986) Eng Med 15(4):191
- Dermody G, Fritz R (2019) Nurs Inq 26(1)
- Emborg J, Matjacic Z, Bendtsen JD, Spaich EG, Cikajlo I, Goljar N, Andersen OK (2011) IEEE Trans Biomed Eng 58(4):960. <https://doi.org/10.1109/TBME.2010.2096507>
- Favre AS, Jandre FC, Giannella-Neto A (2003) In: Proceedings of the 25th annual international conference of the IEEE engineering in medicine and biology society, Piscataway, vol 1, pp 419–422
- Frankel MA, Dowden BR, Mathews VJ, Normann RA, Clark GA, Meek SG (2011) IEEE Trans Neural Syst Rehabil Eng 19(3):325. <https://doi.org/10.1109/TNSRE.2011.2123920>
- Garcia-Saez G, Hernando ME, Martinez-Sarriegui I, Rigla M, Torralba V, Bragues E, de Leiva A, Gomez EJ (2009) Int J Med Inform 78(6):391
- Godfrey K (1983) In: Compartmental models and their application. Academic Press, London
- Guan S, Gu J, Shen Z, Wang J, Huang Y, Mason A (2011) In: 2011 IEEE Biomedical Circuits and Systems Conference (BioCAS), pp 193–196
- Guodong C, Peifa J, Yan L (2007) J Tsinghua Univ Sci Technol 47(1):131
- Halperin D, Heydt-Benjamin TS, Ransford B, Clark SS, Defend B, Morgan W, Fu K, Kohno T, Maisel WH (2008) In: 2008 IEEE Symposium on Security and Privacy (SP 2008), pp 129–142
- Hansen T, Henningsen C, Nielsen J, Pedersen R, Schwensen J, Sivabalan S, Larsen J, Leth J (2012) In: 2012 IEEE International Conference on Control Applications (CCA). IEEE, pp 859–864
- Harders M, Szekely G (2003) Proc IEEE 91(9):1430
- Hatsopoulos N, Mukand J, Polykoff G, Friehs G, Donoghue J (2005) In: 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Piscataway, pp 4
- Hovorka R, Allen JM, Eleri D, Chassin LJ, Harris J, Xing D, Kollman C, Hovorka T, Larsen AM, Nodale M, De Palma A, Willinska ME, Acerini CL, Dunger DB (2010) Lancet 375:743
- Iasemidis LD, Sabesan S, Good LB, Tsakalis K, Treiman DM (2009) In: 11th International Congress of the IUPESM. Medical Physics and Biomedical Engineering. World Congress 2009. Neuroengineering, Neural Systems, Rehabilitation and Prosthetics, Berlin, pp 588–591
- Iobbi MG, Simonds AK, Dickinson RJ (2007) J Clin Monit Comput 21(2):115. <https://doi.org/10.1007/s10877-006-9064-6>

- Ivanov R, Weimer J, Simpao A, Rehman M, Lee I (2015) In: Proceedings of the 6th International Conference on Cyber-Physical Systems, pp 110–119
- Jacquez J (1996) BioMedware
- Jiang Z, Pajic M, Mangharam R (2012) *Proc IEEE* 100(1):122
- Jiang Z, Pajic M, Mangharam R (2011a) In: Proceedings 10th International Conference on Information Processing in Sensor Networks (IPSN 2010), pp 119–120
- Jiang Z, Pajic M, Mangharam R (2011b) In: Proceedings 2011 IEEE/ACM International Conference on Cyber-Physical Systems (ICCPs), pp 131–140
- Jiang Z, Pajic M, Mangharam R (2012) In: *Proc IEEE*, pp 122–137
- Juneja R, Roudebush C, Kumar N, Macy A, Golas A, Wall D, Wolverton C, Nelson D, Carroll J, Flanders S (2007) *Diabetes Technol Ther* 9(3):232
- Koppel R, Cohen A, Abaluck B, Localio AR, Kimmel SE, Strom BL (2005) *J Am Med Assoc* 293(10):1197
- Koppel R, Wetterneck TB, Telles JL, Karsh BT (2008) *J Am Med Inform Assoc* 15(4):408
- Kotz D (2011) In: 2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011), pp 1–6
- Kovatchev B, Cobelli C, Renard E, Anderson S, Breton M, Patek S, Clarke W, Bruttomesso D, Maran A, Costa S, Avogaro A, Man CD, Facchinetti A, Magni L, De Nicolao G, Place J, Farret A (2010) *J Diab Sci and Tech* 4:1374
- Lai CC, Lee RG, Hsiao CC, Liu HS, Chen CC (2009) *J Netw Comput Appl* 32(6):1229. <https://doi.org/10.1016/j.jnca.2009.05.007>
- Lawo M, Herzog O (2011) In: 8th International Conference and Expo on Emerging Technologies for a Smarter World, CEWIT 2011, Hauppauge. <https://doi.org/10.1109/CEWIT.2011.6135882>
- Lee MY, Wong MK, Chang HS, Chen KY (1995) *Biomed Eng Appl Basis Commun* 7(2):153
- Liu FY, Northrop RB (1990) *IEEE Trans Biomed Eng* 37(12):1147. <https://doi.org/10.1109/10.64459>
- Marzullo TC, Lehmkuhle MJ, Gage GJ, Kipke DR (2010) *IEEE Trans Neural Syst Rehabil Eng* 18(2):117
- Mason DG, Packer JS, Cade JF, McDonald RD (1985) *Australas Phys Eng Sci Med* 8(4):164
- Morozoff EP, Smyth JA (2009) In: 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society. EMBC 2009, Piscataway, pp 3079–3082. <https://doi.org/10.1109/IEMBS.2009.5332532>
- Nikitzczuk J, Weinberg B, Canavan PK, Mavroidis C (2010) *IEEE/ASME Trans Mechatronics* 15(6):952
- Nirjon S, Dicerson RF, Li Q, Asare P, Stankovic JA, Hong D, Zhang B, Jiang X, Shen G, Zhao F (2012) In: Proceedings of SenSys
- O’Keeffe DT, Maraka S, Basu A, Keith-Hynes P, Kudva YC (2015) *Diabetes Technol Ther* 17(9):664
- Pajic M, Mangharam R, Sokolsky O, Arney D, Goldman J, Lee I (2014) *IEEE Trans Ind Inform* 10(1):3
- Palerm CC (2011) *Comput Methods Program Biomed* 102(2):130. <https://doi.org/10.1016/j.cmpb.2010.06.007>
- Patek SD, Magni L, Dassau E, Hughes-Karvetski C, Toffanin C, De Nicolao G, Del Favero S, Breton M, Dalla Man C, Renard E, Zisser H, Doyle III FJ, Cobelli C, Kovatchev BP (2012) *IEEE Trans Biomed Eng* 59(11):2986
- Paul N, Kohno T, Klonoff DC (2011) *J Diabetes Sci Technol* 5(6):1557
- Penning RS, Jung J, Borgstadt JA, Ferrier NJ, Zinn MR (2011) In: IEEE International Conference on Robotics and Automation (ICRA 2011), pp 4822–4827
- Pennycott A, Hunt KJ, Coupaud S, Allan DB, Kakebeeke TH (2010) *IEEE Trans Control Syst Technol* 18(1):136
- Percival MW, Wang Y, Grosman B, Dassau E, Zisser H, Jovanovic L, Doyle III FJ (2011) *J Process Control* 21(3):391
- Petrofsky JS, Heaton HH III, Phillips CA (1984) *Med Biol Eng Comput* 22(4):298
- Radcliffe J (2011) Hacking medical devices for fun and insulin: Breaking the human SCADA system. Black Hat 2011, Cf. https://media.blackhat.com/bh-us-11/Radcliffe/BH_US_11_Radcliffe_Hacking_Medical_Devices_WP.pdf
- Li C, Raghunathan A, Jha NK (2011) In: International Conference on e-Health Networking, Applications and Security, pp 151–156
- Rekate HL (1980) *Ann Biomed Eng* 8(4–6):515
- Roederer A, Weimer J, DiMartino J, Gutsche J, Lee I (2015) In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, pp 1504–1507
- Ruiz R, Borches D, Gonzalez A, Corral J (1993) *Biomed Instrum Technol* 27(3):244
- Saager L, Collins G, Burnside B, Tymkew H, Zhang L, Jacobsohn E, Avidan M (2008) *J Cardiothorac Vasc Anesth* 22(3):377
- Salam MT, Mirzaei M, Ly MS, Nguyen DK, Sawan M (2012) *IEEE Trans. Neural Syst Rehabil Eng* 20(4):432
- Shoaib M, Jha N, Verma N (2011) In: Proceedings 2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC 2011), Piscataway, NJ, USA, pp 591–596
- Simeral JD, Kim SP, Black MJ, Donoghue JP, Hochberg LR (2011) *J Neural Eng* 8(2):025027 (24 pp). <https://doi.org/10.1088/1741-2560/8/2/025027>
- Slepian MJ, Alemu Y, Soares JS, Smith RG, Einav S, Bluestein D (2013) *J Biomech* 46(2):266. <https://doi.org/10.1016/j.jbiomech.2012.11.032>
- Smith B, Barnea O, Moore TW, Jaron D (1991) *Med Biol Eng Comput* 29(2):180
- Stanislaski S, Afshar P, Cong P, Giftakis J, Stypulkowski P, Carlson D, Linde D, Ullestad D, Avestruz A, Denison T (2012) *IEEE Trans Neural Syst Rehabil Eng* 20(4):410. <https://doi.org/10.1109/TNSRE.2012.2183617>
- Stauffer Y, Allemand Y, Bourri M, Fournier J, Clavel R, Metrailler P, Brodard R (2008) *J Biomech* 41:S136
- Steil GM, Rebrin K, Darwin C, Hariri F, Saad MF (2006) *Diabetes* 55:3344
- Sung GT, Gill IS (2001) *Urology* 58(6):893

- Turchetti G, Palla I, Pierotti F, Cuschieri A (2012) *Surg Endosc* 1–9
- Verma N, Kyong HL, Shoeb A (2011) *J Low Power Electron Appl* 1(1):150. <https://doi.org/10.3390/jlpea1010150>
- Wang F, Shastri M, Jones CL, Gupta V, Osswald C, Kang X, Kamper DG, Sarkar N (2011) In: 2011 IEEE International Conference on Robotics and Automation (ICRA 2011), Piscataway, 2011, pp 3688–3693. <https://doi.org/10.1109/ICRA.2011.5980099>
- Waterhouse E (2003) *IEEE Eng Med Biol Mag* 22(3):74
- Weber DJ, Stein RB, Everaert DG, Prochazka A (2006) *IEEE Trans Neural Syst Rehabil Eng* 14(2):240. <https://doi.org/10.1109/TNSRE.2006.875575>
- Weimer J, Chen S, Peleckis A, Rickels MR, Lee I (2016) *Diabetes Technol Ther* 18(10):616
- Weimer J, Ivanov R, Chen S, Roederer A, Sokolsky O, Lee I (2017) *Proc IEEE* 106(1):71
- Weingart SN, Toth M, Sands DZ, Aronson MD, Davis RB, Phillips RS (2003) *Arch Internal Med* 163(21):2625
- Weinzimer S, Steil G, Swan K, Dziura J, Kurtz N, Tamborlane W (2008) *Diabetes Care* 31:934
- Witcher MR, Ellis TL (2012) *Front Comput Neurosci* 6. <https://doi.org/10.3389/fncom.2012.00061>
- Worsnopp TT, Peshkin MA, Colgate JE, Kamper DG (2007) In: IEEE 10th International Conference on Rehabilitation Robotics, Piscataway, pp 896–901
- Wu W, Shaikhouni A, Donoghue JR, Black MJ (2004) In: 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Piscataway, vol 6, pp 4126–4129
- Wyman DR, Wilson BC, Malone DE (1992) *Proc IEEE* 80(6):890. <https://doi.org/10.1109/5.149451>
- Yamashiro SM (2007) *Med Biolog Eng Comput* 45(4):345. <https://doi.org/10.1007/s11517-006-0153-y>
- Zisser H, Dassau E, Bevier W, Harvey R, Percival MW, Grosman B, Seborg D, Jovanovic L, Doyle III FJ (2011) *Diabetes* 60(Supl 1), A41. *Proc. Amer. Dia. Assoc. Meeting*

Model Building for Control System Synthesis

Marco Lovera and Francesco Casella
Politecnico di Milano, Milano, Italy

Abstract

The process of developing control-oriented mathematical models of physical systems is a complex task, which in general implies a careful combination of prior knowledge about the physics of the system under study with information coming from experimental data.

In this article the role of mathematical models in control system design and the problem of developing compact control-oriented models are discussed.

Keywords

Control-oriented modelling · Simulation · Continuous-time systems · Discrete-time systems · Time-invariant systems · Time-varying systems · Parameter-varying systems · Analytical models · Computational modelling · System identification · Uncertainty

Introduction

The design of automatic control systems requires the availability of some knowledge of the dynamics of the process to be controlled. In this respect, current methods for control system synthesis can be classified in two broad categories: model-free and model-based ones.

The former aim at designing (or tuning) controllers solely on the basis of experimental data collected directly on the plant, without resorting to mathematical models.

The latter, on the contrary, assume that suitable models of the plant to be controlled are available and rely on this information to work out control laws capable of meeting the design requirements.

While the research on model-free design methods is a very active field, the vast majority of control synthesis methods and tools fall in the model-based category and therefore assume that knowledge about the plant to be controlled is encoded in the form of dynamic models of the plant itself. Furthermore, in an increasing number of application areas, control system performance is becoming a key competitive factor for the success of innovative, high-tech systems. Consider, for example, high-performance mechatronic systems (such as robots), vehicles enhanced by active integrated stability, suspension and braking control, aerospace systems, and advanced energy conversion systems. All the

abovementioned applications possess at least one of the following features, which in turn call for accurate mathematical modelling for the design of the control system: closed-loop performance critically depends on the dynamic behavior of the plant; the system is made of many closely interacting subsystems; advanced control systems are required to obtain competitive performance, and these in turn depend on explicit mathematical models for their design; the system is safety critical and requires extensive validation of closed-loop stability and performance by simulation.

Therefore, building control-oriented mathematical models of physical systems is a crucial prerequisite to the design process itself (see, e.g., Lovera 2014 for a more detailed treatment of this topic).

In the following, two aspects related to modelling for control system synthesis will be discussed, namely, the role of models for control system synthesis and the actual process of model building itself.

The Role of Models for Control System Synthesis

Mathematical models play a number of different roles in the design of control systems. In particular, different classes of mathematical models are usually employed: *detailed*, high-fidelity, models for system simulation and *compact* models for control design. In this section the two model classes are presented, and their respective roles in the design of control systems are described. Note, in passing, that although hybrid system control is an interesting and emerging field, this paper will focus on purely continuous-time physical models, with application to the design of continuous-time or sampled-data control systems.

Detailed Models for System Simulation

Detailed models play a double role in the control design process. On one hand they allow checking how good (or crude) the compact model is, compared to a more detailed description, thus helping to develop good compact models. On the other

hand, they allow closed-loop performance verification of the controlled system, once a controller design is available (Indeed, verifying closed-loop performance using the same simplified model that was used for control system design is not a sound practice; conversely, verification performed with a more detailed model is usually deemed a good indicator of the control system performance, whenever experimental validation is not possible for some reason.).

Object-oriented modelling (OOM) methodologies and equation-based, object-oriented languages (EOOLs) provide very good support for the development of such high-fidelity models, thanks to equation-based modelling, acausal physical ports, hierarchical system composition, and inheritance. In particular, the Modelica language embeds these concepts and methodologies in an open standard, supported by a growing number of commercial and open-source tools; see Tiller (2001) and Fritzson (2014) for a comprehensive overview, and (Tiller 2019) for a quick introduction to the language. Any continuous-time EOOL model is equivalent to the set of differential-algebraic equations (DAEs) expressed in terms of the vector function F given by

$$F(x(t), \dot{x}(t), u(t), y(t), p, t) = 0, \quad (1)$$

where x is the vector of dynamic variables, u is the vector of input variables, y is the vector of algebraic variables, p is the vector of parameters, and t is the time.

It is interesting to highlight that the object-oriented approach (in particular, the use of replaceable components) allows defining and managing families of models of the same plant with different levels of complexity, by providing more or less detailed implementations of the same abstract interfaces. This feature of OOM allows the development of simulation models for different purposes and with different degrees of detail throughout the entire life of an engineering project, from preliminary design down to commissioning and personnel training, all within a coherent framework.

When focusing on control systems verification (and regardless of the actual control design methodology) once the controller has been set up, an OOM tool can be used to run closed-loop simulations, including both the plant and the controller model.

It is also possible to run such simulations within a computer-aided control system design (CACSD) environment that only supports causal plant models described by state-space equations (2). To this aim, one can use the model export facilities of OOM tools that automatically transform a detailed EOOL plant model featuring only causal top-level connectors (actuator input and sensor output signals) into a causal model. This is accomplished by reformulating (1) in state-space form (2), by means of symbolic and numerical transformations and then by generating executable code to compute the right-hand side of the state equations. The Functional Mock-up Interface (FMI) is a widely popular open standard enabling this kind of model exchange, see FMI Working Group (2014). As of the time of this writing, FMI is supported by over a hundred simulation and CACSD tools.

Finally, it is important to point out that physical model descriptions based on partial-differential equations (PDEs) can be handled in the OOM framework by means of discretization using finite volume, finite elements, or finite differences methods.

Compact Models for Control Design

The requirements for a control-oriented model can vary significantly from application to application. Design models can be tentatively classified in terms of two key features: complexity and accuracy. For a dynamic model, complexity can be measured in terms of its order; accuracy, on the other hand, can be measured using many different metrics (e.g., time-domain simulation or prediction error, frequency domain matching with the real plant, etc.) related to the capability of the model to reproduce the behavior of the true system in the operating conditions of interest.

Broadly speaking, it can be safely stated that the required level of closed-loop performance drives the requirements on the accuracy and

complexity of the design model. Similarly, it is intuitive that more complex models have the potential for being more accurate. So, one might be tempted to resort to very detailed mathematical representations of the plant to be controlled in order to maximize closed-loop performance. This consideration however is moderated by a number of additional requirements, which actually end up driving the control-oriented modelling process. First of all, present-day controller synthesis methods and tools have computational limitations in terms of the complexity of the mathematical models they can handle, so compact models representative of the dominant dynamics of the system under study are what is really needed. Furthermore, for many synthesis methods (such as, e.g., LQG or H_∞ synthesis), the complexity of the design model has an impact on the complexity of the controller, which in turn is constrained by implementation issues. Last but not least, in engineering projects the budget of the control-oriented modelling activity is usually quite limited, so the achievable level of accuracy is affected by this limitation.

It is clear from the above discussion that developing mathematical models suitable for control system synthesis is a nontrivial task but rather corresponds to the pursuit of a careful trade-off between complexity and accuracy. Furthermore, throughout the model development, one should keep in mind the eventual control application of the model, so its mathematical structure has to be compatible with currently available methods and tools for control system analysis and design.

Control-oriented models are usually formulated in state-space form:

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t), p, t) \\ y(t) &= g(x(t), u(t), p, t)\end{aligned}\quad (2)$$

where x is the vector of state variables, u is the vector of system inputs (control variables and disturbances), y is the vector of system outputs, p is the vector of parameters, and t is the continuous time. In the following, however, the focus will be on linear models, which constitute the starting point for most control law design methods and tools. In this respect, the main categories of

models used in control system synthesis can be defined as follows.

Linear Time-Invariant Models

Linear, time-invariant (LTI) models can be described in state-space form as

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}\quad (3)$$

or, equivalently, using an input–output model given by the (rational) transfer function matrix

$$G(s) = C(sI - A)^{-1}B + D, \quad (4)$$

where s denotes the Laplace variable. In many cases, the dynamics of systems in the form (2) in the neighborhood of an equilibrium (trim) point is approximated by (3) via analytical or numerical linearization; in this case, typically, u , x , and y represent the deviations of the variables with respect to the corresponding equilibrium values.

If, on the contrary, the control-oriented model is obtained by linearization of the DAE system (1), then a generalized LTI (or descriptor) model in the form

$$\begin{aligned}E\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}\quad (5)$$

is obtained. Clearly, a generalized LTI model is equivalent to a conventional one as long as E is non-singular. The generalized form, however, encompasses the wider class of linearized plants with a singular E .

Linear Time-Varying Models

In some engineering applications, the need may arise to linearize the detailed model in the neighborhood of a trajectory rather than around an equilibrium point. Whenever this is the case, a linear, time-varying (LTV) model is obtained, in the form

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t),\end{aligned}\quad (6)$$

where $A(t)$, $B(t)$, $C(t)$, and $D(t)$ are matrices and the entries of which are functions of time. An important subclass is the one of time-periodic behavior of the entries of the state-space matrices of the model, which corresponds to a linear time-periodic (LTP) model. LTP models arise when considering the linearization along periodic trajectories or, as it occurs in a number of engineering problems, whenever rotating systems are considered (e.g., spacecraft, rotorcraft, wind turbines). Finally, it is interesting to recall that (discrete-time) LTP models are needed to model multi-rate sampled-data systems.

Linear Parameter-Varying Models

The control-oriented modelling problem can be also formulated as the one of *simultaneously* representing all the linearizations of interest for control purposes of a given nonlinear plant. Indeed, in many control engineering applications, a single control system must be designed to guarantee the satisfactory closed-loop operation of a given plant in many different operating conditions (either in equilibria or along trajectories). Many design techniques are now available for this problem (see, e.g., Mohammadpour and Scherer 2012), provided that a suitable model in parameter-dependent form has been derived for the system to be controlled. Linear, parameter-varying (LPV) models described in state-space form as

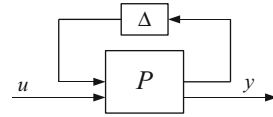
$$\begin{aligned}\dot{x}(t) &= A(p(t))x(t) + B(p(t))u(t) \\ y(t) &= C(p(t))x(t) + D(p(t))u(t)\end{aligned}\quad (7)$$

are linear models the state-space matrices of which depend on a parameter vector p that can be time dependent. The elements of vector p may or may not be measurable, depending on the specific problem formulation. The present state-of-the-art of LPV modelling can be briefly summarized by defining two classes of approaches (see dos Santos et al. 2011 for details). *Analytical* methods are based on the availability of reliable nonlinear equations for the dynamics of the plant, from which suitable control-oriented representations can be derived (by resorting to, broadly speaking, suitable extensions of the familiar notion

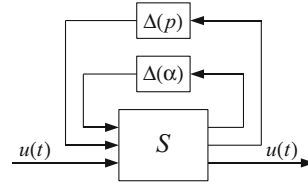
of linearization, developed in order to take into account off-equilibrium operation of the system). *Experimental* methods are based entirely on identification, i.e., aimed at deriving LPV models for the plant directly from input–output data. In particular, some LPV identification techniques assume that one *global* identification experiment in which both the control input and the parameter vector are (persistently) excited in a simultaneous way, while others aim at deriving a parameter-dependent model on the basis of *local* experiments only, i.e., experiments in which the parameter vector is held constant and only the control input is excited.

Modern control theory provides methods and tools to deal with design problems in which stability and performance have to be guaranteed also in the presence of model uncertainty, both for regulation around a specified operating point and for gain-scheduled control system design. Therefore, modelling for control system synthesis should also provide methods to account for model uncertainty (both parametric and nonparametric) in the considered model class.

Most of the existing control design literature assumes that the plant model is given in the form of a linear fractional transformation (LFT) (see, e.g., Skogestad and Postlethwaite 2007 for an introduction to LFT modelling of uncertainty and Hecker et al. 2005 for a discussion of algorithms and software tools). LFT models consist of a feedback interconnection between a *nominal* LTI plant and a (usually norm-bounded) operator which represents model uncertainties, e.g., poorly known or time-varying parameters, nonlinearities, etc. A typical LFT interconnection is depicted in Fig. 1, where the nominal plant is denoted with P and the uncertainty block is denoted with Δ . The LFT formalism can be also used to provide a structured representation for the state-space form of LPV models, as depicted in Fig. 2, where the block $\Delta(\alpha)$ takes into account the presence of the uncertain parameter vector α , while the block $\Delta(p)$ models the effect of the varying operating point, parameterized by the vector of time-varying parameters p . Therefore, LFT models can be used for the design of robust and gain-scheduling controllers; in addition they



Model Building for Control System Synthesis, Fig. 1 Block diagram of the typical LFT interconnection adopted in the robust control framework



Model Building for Control System Synthesis, Fig. 2 Block diagram of the typical LFT interconnection adopted in the robust LPV control framework

can also serve as a basis for structured model identification techniques, where the uncertain parameters that appear in the feedback blocks are estimated based on input–output data sequences. The process of extracting uncertain/scheduling parameters from the design model of the system to be controlled is a highly complex one, in which symbolic techniques play a very important role. Tools already exist to perform this task (see, e.g., Hecker et al. 2005), while a recent overview of the state-of-the-art in this research area can be found in Hecker and Varga (2006).

Finally, it is important to point out that there is a vast body of advanced control techniques which are based on discrete-time models:

$$\begin{aligned} x(k+1) &= f(x(k), u(k), p, k) \\ y(k) &= g(x(k), u(k), p, k) \end{aligned} \quad (8)$$

where the integer time step k usually corresponds to multiples of a sampling period T_s . Many techniques are available to transform (2) into (8). Furthermore, LTI, LTV, and LPV models can be formulated in discrete time rather than in continuous time (see, e.g., Landau and Zito 2006 and Toth et al. 2012).

Building Models for Control System Synthesis

The development of control-oriented models of physical systems is a complex task, which in general implies a careful combination of prior knowledge about the physics of the system under study with information coming from experimental data. In particular, this process can follow very different paths depending on the type of information which is available on the plant to be controlled. Such paths are typically classified in the literature as follows (see, e.g., Ljung 2008 for a more detailed discussion).

White box modelling refers to the development of control-oriented models on the basis of first principles only. In this framework, one uses the available information on the plant to develop a detailed model using OOM or EOOL tools and subsequently works out a compact control-oriented model from it. If the adopted tool only supports simulation, then one can run simulations of the plant model, subject to suitably chosen excitation inputs (ranging from steps to persistently exciting input sequences such as, e.g., pseudorandom binary sequences and sine sweeps), and then reconstruct the dynamics by means of system identification methods. Note that in this way, the structure/order selection stage of the system identification process provides effective means to manage the complexity versus accuracy trade-off in the derivation of the compact model. A more direct approach, presently supported by many tools, is to directly compute the A, B, C, D matrices of the linearized system around specified equilibrium (trim) points, using symbolic or numerical linearization techniques. The result is usually a high-order linear system, which then can (sometimes must) be reduced to a low-order system by using model order reduction techniques (such as, e.g., balanced truncation). Model reduction techniques (see Antoulas 2009 for an in-depth treatment of this topic) allow to automatically obtain approximated compact models such as (3), starting from much more detailed simulation models, by formulating specific approximation bounds in

control-relevant terms (e.g., percentage errors of steady-state output values, norm-bounded additive or multiplicative errors of weighted transfer functions, or L_2 -norm errors of output transients in response to specified input signals).

Black box modelling, on the other hand, corresponds to situations in which the modelling activity is entirely based on input–output data collected on the plant (which therefore must be already available), possibly in dedicated, suitably designed, experiments (see Ljung 1999). Regardless of the type of model to be built (i.e., linear or nonlinear, time-invariant or time-varying, discrete-time or continuous time), the black box approach consists of a number of well-defined steps. First of all the structure of the model to be identified must be defined: in the linear time-invariant case, this corresponds to the choice of the number of poles and zeros for an input–output model or to the choice of model order for a state-space representation; in the nonlinear case, structure selection is a much more involved process in view of the much larger number of degrees of freedom which are potentially involved. Once a model structure has been defined, a suitable cost function to measure the model performance must be selected (e.g., time-domain simulation or prediction error, frequency domain model fitting, etc.), and the experiments to collect identification and validation data must be designed. Finally, the uncertain model parameters must be estimated from the available identification dataset, and the model must be validated on the validation dataset.

Gray box modelling (in various shades) corresponds to the many possible intermediate cases which can occur in practice, ranging from the white box approach to the black box one. As discussed in Ljung (2008), the critical issue in the development of an effective approach to control-oriented gray box modelling lies in the integration of existing methods and tools for physical systems modelling and simulation with methods and tools for parameter estimation. Such integration can take place in a number of different ways depending on the relative role of data and priors on the physics of the system in the specific application. A typical situation which occurs frequently in applications is when a white box model

(developed by means of OOM or EOOL tools) contains parameters having unknown or uncertain numerical values (such as, e.g., damping factors in structural models, aerodynamic coefficients in aircraft models and so on). Then, one may rely on input–output data collected in dedicated experiments on the real system to refine the white box model by estimating the parameters using the information provided by the data. This process is typically dependent on the specific application domain as the type of experiment, the number of measurements, and the estimation technique must meet application-specific constraints (as an example, see, e.g., Klein and Morelli 2006 for an overview of gray box modelling in aerospace applications and Tischler and Tobias 2016 for an overview of the “stitching” approach to LPV modelling using LTI identified models).

Summary and Future Directions

In this article the problem of model building for control system synthesis has been considered. An overview of the different uses of mathematical models in control system design has been provided, and the process of building compact control-oriented models starting from prior knowledge about the system or experimental data has been discussed. Present-day modelling and simulation tools support advanced control system design in a much more direct way. In particular, while methods and tools for the individual steps in the modelling process (such as OOM, linearization and model reduction, parameter estimation) are available, an integrated environment enabling the pursuit of all the abovementioned paths to the development of compact control-oriented models is still a subject for future development. The availability of such a tool might further promote the application of advanced, model-based techniques that are currently limited by the model development process.

Cross-References

- [Descriptor System Techniques and Software Tools](#)

- [Model Order Reduction: Techniques and Tools](#)
- [Modeling of Dynamic Systems from First Principles](#)
- [Multi-domain Modeling and Simulation](#)
- [System Identification: An Overview](#)

Bibliography

- Antoulas A (2009) Approximation of large-scale dynamical systems. SIAM, Philadelphia
- Lopes dos Santos P, Azevedo Perdicoulis TP, Novara C, Ramos JA, Rivera DE (eds) (2011) Linear parameter-varying system identification: new developments and trends. World Scientific Publishing Company, Singapore
- FMI Working Group (2014) Functional mock-up interface for model exchange and co-simulation. <https://fmi-standard.org/downloads/>
- Fritzson P (2014) Principles of object-oriented modeling and simulation with modelica 3.3: a cyber-physical approach. Wiley, Hoboken
- Hecker S, Varga A (2006) Symbolic manipulation techniques for low order LFT-based parametric uncertainty modelling. *Int J Control* 79(11):1485–1494
- Hecker S, Varga A, Magni J-F (2005) Enhanced LFR-toolbox for MATLAB. *Aerosol Sci Technol* 9(2): 173–180
- Klein V, Morelli EA (2006) Aircraft system identification: theory and practice. AIAA, Reston
- Landau ID, Zito G (2006) Digital control systems. Springer, New York
- Ljung L (1999) System identification: theory for the user. Prentice-Hall, Upper Saddle River
- Ljung L (2008) Perspectives on system identification. In: 2008 IFAC world congress, Seoul
- Lovera M (ed) (2014) Control-oriented modelling and identification: theory and practice. IET, Stevenage
- Mohammadpour J, Scherer C (eds) (2012) Control of linear parameter varying systems with applications. Springer, Boston
- Skogestad S, Postlethwaite I (2007) Multivariable feedback control analysis and design. Wiley, Chichester
- Tiller M (2001) Introduction to physical modelling with Modelica. Kluwer, Boston
- Tiller M (2019) Modelica by example. <https://mbe.modelica.university/>. [Online]
- Tischler MB, Tobias EL (2016) A model stitching architecture for continuous full flight-envelope simulation of fixed-wing aircraft and rotorcraft from discrete point linear models. Technical Report AD1008448, Aviation and Missile Research, Development and Engineering Center Redstone Arsenal United States
- Toth R, Lovera M, Heuberger PSC, Corno M, Van den Hof PMJ (2012) On the discretization of linear fractional representations of LPV systems. *IEEE Trans Control Syst Technol* 20(6):1473–1489

Model Order Reduction: Techniques and Tools

Peter Benner¹ and Heike Faßbender²

¹Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany

²Institut Computational Mathematics, Technische Universität Braunschweig, Braunschweig, Germany

Abstract

Model order reduction is here understood as a computational technique to reduce the order of a dynamical system described by a set of ordinary or differential-algebraic equations to facilitate or enable its simulation, the design of a controller, or optimization and design of the physical system modeled. It focuses on representing the map from inputs into the system to its outputs, while its dynamics are treated as a black box so that the large-scale set of describing equations can be replaced by a much smaller set of analogous equations without sacrificing the accuracy of the input-output behavior.

Keywords

Balanced truncation · Interpolation-based methods · Reduced-order models · SLICOT · Truncation-based methods

Problem Description

This survey is concerned with linear time-invariant (LTI) systems in state-space form:

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \quad (1)$$

where $E, A \in \mathbb{R}^{n \times n}$ are the system matrices, $B \in \mathbb{R}^{n \times m}$ is the input matrix, $C \in \mathbb{R}^{p \times n}$ is the output matrix, and $D \in \mathbb{R}^{p \times m}$ is the feedthrough (or input-output) matrix. The size n of the matrix A is often referred to as the order of the LTI

system. It mainly determines the amount of time needed to simulate the LTI system.

Such LTI systems often arise from a finite element modeling using commercial software such as ANSYS or NASTRAN or from a space discretization of nonlinear partial differential equations followed by a suitable linearization. The number n of equations involved may range from a few hundred to a few million. Thus, solving (1) may become infeasible due to limitations on computational resources as well as growing inaccuracy due to numerical ill-conditioning. Model order reduction (MOR) seeks to replace (1) with a (much) smaller set of such equations so that the input-output behavior of both systems is similar in an appropriately defined sense.

The matrix E may be singular. In that case the first equation in (1) defines a system of differential-algebraic equations (DAEs); otherwise it is a system of ordinary differential equations (ODEs). For example, let $E = \begin{bmatrix} J & 0 \\ 0 & 0 \end{bmatrix}$ with a $j \times j$ nonsingular matrix J , $A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$ with the $j \times j$ matrix A_{11} and a nonsingular matrix A_{22} and $B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$ with the $j \times m$ matrix B_1 . Partitioning the state vector $x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$ with $x_1(t)$ of length j , the DAE $E\dot{x}(t) = Ax(t) + Bu(t)$ splits into the set of algebraic equations $0 = A_{22}x_2(t) + B_2u(t)$ and the ODE

$$J\dot{x}_1(t) = A_{11}x_1(t) + (B_1 - A_{12}A_{22}^{-1}B_2)u(t).$$

For clarity of presentation, we focus in the following on systems of ODEs only. All techniques discussed in this survey can be generalized for systems of DAEs even if their structure is more complicated than in the case considered above; see Benner and Stykel (2017) for details. Moreover, to simplify the description, only continuous-time systems are considered here. The discrete-time case, i.e., when the first equation in (1) is a system of difference equations, can be treated mostly analogously; see, e.g., Antoulas (2005).

An alternative way to represent LTI systems is provided by the transfer function matrix (TFM), a matrix-valued function whose elements are rational functions. Assuming $x(0) = 0$ and tak-

ing Laplace transforms in (1) yield $sX(s) = AX(s) + BU(s)$, $Y(s) = CX(s) + DU(s)$, where $X(s)$, $Y(s)$, and $U(s)$ are the Laplace transforms of the time signals $x(t)$, $y(t)$, and $u(t)$, respectively. The map from inputs U to outputs Y is then described by $Y(s) = G(s)U(s)$ with the TFM

$$G(s) = C(sE - A)^{-1}B + D, \quad s \in \mathbb{C}. \quad (2)$$

The aim of MOR is to find an LTI system

$$\tilde{E}\dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \tilde{B}u(t), \quad \tilde{y}(t) = \tilde{C}\tilde{x}(t) + \tilde{D}u(t) \quad (3)$$

of reduced order $r \ll n$ such that the corresponding TFM

$$\tilde{G}(s) = \tilde{C}(s\tilde{E} - \tilde{A})^{-1}\tilde{B} + \tilde{D} \quad (4)$$

approximates the original TFM (2). That is, using the same input $u(t)$ in (1) and (3), we want that the output $\tilde{y}(t)$ of the reduced-order model (ROM) (3) approximates the output $y(t)$ of (1) well enough for the application considered (e.g., controller design). In general, one requires $\|y(t) - \tilde{y}(t)\| \leq \varepsilon$ for all feasible inputs $u(t)$, for (almost) all t in the time domain of interest, and for a suitable norm $\|\cdot\|$. In control theory one often employs the $\mathcal{L}_2$ - or $\mathcal{L}_\infty$ -norms on $\mathbb{R}$ or $[0, \infty]$, respectively, to measure time signals or their Laplace transforms. In the situations considered here, the $\mathcal{L}_2$ -norms employed in frequency and time domain coincide due to the Paley-Wiener theorem (or Parseval's equation or the Plancherel theorem, respectively); see Antoulas (2005) and Zhou et al. (1996) for details. As $Y(s) - \tilde{Y}(s) = (G(s) - \tilde{G}(s))U(s)$, one can therefore consider the approximation error of the TFM $\|G(\cdot) - \tilde{G}(\cdot)\|$ measured in an induced norm instead of the error in the output $\|y(\cdot) - \tilde{y}(\cdot)\|$.

Depending on the choice of the norm, different MOR goals can be formulated. Typical choices are (see, e.g., Antoulas 2005 for a more thorough discussion)

- $\|G(\cdot) - \tilde{G}(\cdot)\|_{\mathcal{H}_\infty}$, where

$$\|F(\cdot)\|_{\mathcal{H}_\infty} = \sup_{s \in \mathbb{C}_+} \sigma_{\max}(F(s)).$$

Here, $\sigma_{\max}$ is the largest singular value of the matrix $F(s)$, and $\mathbb{C}_+$ denotes the open right half complex plane. This minimizes the maximal magnitude of the frequency response of the error system and by the Paley-Wiener theorem bounds the $\mathcal{L}_2$ -norm of the output error;

- $\|G(\cdot) - \tilde{G}(\cdot)\|_{\mathcal{H}_2}$, where (with $\iota = \sqrt{-1}$)

$$\|F(\cdot)\|_{\mathcal{H}_2}^2 = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \text{tr}(F(\iota\omega)^* F(\iota\omega)) d\omega.$$

This ensures a small error

$$\|y(\cdot) - \tilde{y}(\cdot)\|_{\mathcal{L}_\infty(0, \infty)} = \sup_{t > 0} \|y(t) - \tilde{y}(t)\|_\infty$$

(with $\|\cdot\|_\infty$ denoting the maximum norm of a vector) uniformly over all inputs $u(t)$ having bounded $\mathcal{L}_2$ -energy, that is, $\int_0^\infty u(t)^T u(t) dt \leq 1$; see Gugercin et al. (2008).

Besides a small approximation error, one may impose additional constraints for the ROM. One might require certain properties (such as stability and passivity) of the original system to be preserved. Rather than considering the full nonnegative real line in time domain or the full imaginary axis in frequency domain, one can also consider bounded intervals in both domains. For these variants, see, e.g., Antoulas (2005) and Obinata and Anderson (2001).

Methods

There are a number of different methods to construct ROMs; see, e.g., Antoulas (2005), Benner et al. (2005, 2017), Obinata and Anderson (2001), and Schilders et al. (2008). In this survey, we concentrate on projection-based methods which restrict the full state $x(t)$ to an r -dimensional subspace by choosing $\tilde{x}(t) = W^* x(t)$, where W is an $n \times r$ matrix. Here the conjugate transpose of a complex-valued matrix Z is denoted by Z^* , while the transpose of a matrix Y will be denoted by Y^T .

Choosing $V \in \mathbb{C}^{n \times r}$ such that $W^*V = I \in \mathbb{R}^{r \times r}$ yields an $n \times n$ projection matrix $\Pi = VW^*$ which projects onto the r -dimensional subspace spanned by the columns of V along the kernel of W^* . Applying this projection to (1), one obtains the reduced-order LTI system (3) with

$$\tilde{E} = W^*EV, \quad \tilde{A} = W^*AV, \quad \tilde{B} = W^*B, \quad \tilde{C} = CV \quad (5)$$

and an unchanged $\tilde{D} = D$. If $V = W$, Π is an orthogonal projector and is called a Galerkin projection. If $V \neq W$, Π is an oblique projector, sometimes called a Petrov-Galerkin projection.

In the following, we will briefly discuss the main classes of methods to construct suitable matrices V and W : truncation-based methods and interpolation-based methods. Other methods, in particular combinations of the two classes discussed here, can be found in the literature. In case the original LTI system is real, it is often desirable to construct a real ROM. All of the methods discussed in the following either do construct a real ROM or there is a variant of the method which does. In order to keep this exposition at a reasonable length, the reader is referred to the cited literature.

Truncation-Based Methods

The general idea of truncation is most easily explained by *modal truncation*: For simplicity, assume that $E = I$ and that A is diagonalizable, $T^{-1}AT = D_A = \text{diag}(\lambda_1, \dots, \lambda_n)$. Further we assume that the eigenvalues $\lambda_\ell \in \mathbb{C}$ of A can be ordered such that

$$\text{Re}(\lambda_n) \leq \text{Re}(\lambda_{n-1}) \leq \dots \leq \text{Re}(\lambda_1) < 0, \quad (6)$$

(i.e., all eigenvalues lie in the open left half complex plane). This implies that the system is stable. Let V be the $n \times r$ matrix consisting of the first r columns of T , and let W^* be the first r rows of T^{-1} , that is, $W = V(V^*V)^{-1}$. Applying the transformation T to the LTI system (1) yields

$$T^{-1}\dot{x}(t) = (T^{-1}AT)T^{-1}x(t) + (T^{-1}B)u(t) \quad (7)$$

$$y(t) = (CT)T^{-1}x(t) + Du(t) \quad (8)$$

with

$$T^{-1}AT = \begin{bmatrix} W^*AV & \\ & A_2 \end{bmatrix}, \quad T^{-1}B = \begin{bmatrix} W^*B \\ B_2 \end{bmatrix},$$

and $CT = [CV \ C_2]$, where

$$W^*AV = \text{diag}(\lambda_1, \dots, \lambda_r) \text{ and}$$

$A_2 = \text{diag}(\lambda_{r+1}, \dots, \lambda_n)$. Preserving the r dominant poles (eigenvalues with largest real parts) by truncating the rest (i.e., discarding A_2 , B_2 , and C_2 from (7)) yields the ROM as in (5). It can be shown that the error bound

$$\|G(\cdot) - \tilde{G}(\cdot)\|_{\mathcal{H}_\infty} \leq \|C_2\| \|B_2\| \frac{1}{|\text{Re}(\lambda_{r+1})|}$$

holds, e.g. (Benner 2006). As eigenvalues contain only limited information about the system, this is not necessarily a meaningful ROM. In particular, the dependence of the input-output relation on B and C is completely ignored. This can be enhanced by more refined dominance measures taking B and C into account; see, e.g., Varga (1995) and Benner et al. (2011).

More suitable ROMs can be obtained by *balanced truncation*. To introduce this concept, we no longer need to assume A to be diagonalizable, but we require the stability of A in the sense of (6). For simplicity, we assume $E = I$. Loosely speaking, a balanced representation of an LTI system is obtained by a change of coordinates such that the states which are hard to reach are at the same time those which are difficult to observe. This change of coordinates amounts to an equivalence transformation of the realization (A, B, C, D) of (1) called state-space transformation as in (7), where T now is the matrix representing the change of coordinates. The new system matrices $(T^{-1}AT, T^{-1}B, CT, D)$ form a balanced realization of (1). Truncating in this balanced realization the states that are hard to reach and difficult to observe results in a ROM.

Consider the Lyapunov equations

$$AP + PA^T + BB^T = 0, \quad A^T Q + QA + C^T C = 0. \quad (9)$$

The solution matrices P and Q are called controllability and observability Gramians, respectively. If both Gramians are positive definite, the LTI system is minimal. This will be assumed from hereon in this section.

In balanced coordinates the Gramians P and Q of a stable minimal LTI system satisfy $P = Q = \text{diag}(\sigma_1, \dots, \sigma_n)$ with the Hankel singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$. The Hankel singular values are the positive square roots of the eigenvalues of the product of the Gramians PQ , $\sigma_k = \sqrt{\lambda_k(PQ)}$. They are system invariants, i.e., they are independent of the chosen realization of (1) as they are preserved under state-space transformations.

Given the minimal LTI system (1) in a non-balanced coordinate form and the Gramians P and Q satisfying (9), the transformation matrix T which yields an LTI system in balanced coordinates can be computed via the so-called square root algorithm as follows:

- Compute the Cholesky factors S and R of the Gramians such that $P = S^T S$, $Q = R^T R$.
- Compute the singular value decomposition of $SR^T = \Phi \Sigma \Gamma^T$, where Φ and Γ are orthogonal matrices and $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ is a diagonal matrix with the Hankel singular values σ_i , $i = 1, \dots, n$, ordered decreasingly on its diagonal. $T = S^T \Phi \Sigma^{-\frac{1}{2}}$ yields the balancing transformation (note that $T^{-1} = \Sigma^{\frac{1}{2}} \Phi^T S^{-T} = \Sigma^{-\frac{1}{2}} \Gamma^T R$).
- Partition Φ , Σ , Γ , T , T^{-1} into blocks of corresponding sizes

$$\Sigma = \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix}, \Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}, \Gamma^T = \begin{bmatrix} \Gamma_1^T \\ \Gamma_2^T \end{bmatrix},$$

$$T = [V \ V_2], T^{-1} = \begin{bmatrix} W^T \\ W_2^T \end{bmatrix},$$

with $\Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_r)$ and apply T to (1) to obtain (7) with

$$T^{-1}AT = \begin{bmatrix} W^T AV & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

$$T^{-1}B = \begin{bmatrix} W^T B \\ B_2 \end{bmatrix}, CT = [CV \ C_2]. \quad (10)$$

Note that $W = R^T \Gamma_1 \Sigma_1^{-\frac{1}{2}}$ and $V = S^T \Phi_1 \Sigma_1^{-\frac{1}{2}}$. Preserving the r dominant Hankel singular values by truncating the rest yields the ROM as in (5).

As $W^T V = I$, balanced truncation is an oblique projection method. The ROM is stable with the Hankel singular values $\sigma_1, \dots, \sigma_r$. It can be shown that if $\sigma_r > \sigma_{r+1}$, the error bound

$$\|G(\cdot) - \tilde{G}(\cdot)\|_{\mathcal{H}_\infty} \leq 2 \sum_{k=r+1}^n \sigma_k \quad (11)$$

holds. Given an error tolerance, this allows to choose the appropriate order r of the ROM in the course of the computations.

As the explicit computation of the balancing transformation T is numerically hazardous, one usually uses the equivalent balancing-free square root algorithm (Varga 1991) in which orthogonal bases for the column spaces of V and W are computed. The so obtained ROM is no longer balanced but preserves all other properties (error bound, stability). Furthermore, it is shown in Benner et al. (2000) how to implement the balancing-free square root algorithm using low-rank approximations to S and R without ever having to resort to the square solution matrices P and Q of the Lyapunov equations (9). This yields an efficient algorithm for balanced truncation for LTI systems with large dense matrices. For systems with large-scale sparse A , efficient algorithms based on sparse solvers for (9) exist; see Benner (2006).

By replacing the solution matrices P and Q of (9) by other pairs of positive (semi-)definite matrices characterizing alternative controllability and observability related system information, one obtains a family of model reduction methods including stochastic/bounded-real/positive-real balanced truncation. These can be used if further properties like minimum phase, passivity, etc. are to be preserved in the ROM; for further details,

see Antoulas (2005) and Obinata and Anderson (2001).

The balanced truncation yields good approximation at high frequencies as $\tilde{G}(i\omega) \rightarrow G(i\omega)$ for $\omega \rightarrow \infty$ (as $\tilde{D} = D$), while the maximum error is often attained for $\omega = 0$. For a perfect match at zero and a good approximation for low frequencies, one may employ the *singular perturbation approximation* (SPA, also called balanced residualization). In view of (7) and (10), balanced truncation can be seen as partitioning $T^{-1}x$ according to (10) into $[x_1^T, x_2^T]^T$ and setting $x_2 \equiv 0$ implying $\dot{x}_2 \equiv 0$ as well. For SPA, one assumes that x_2 is in steady state, that is, $\dot{x}_2 \equiv 0$, such that

$$\begin{aligned}\dot{x}_1 &= W^T A V x_1 + A_{12} x_2 + W^T B u, \\ 0 &= A_{21} x_1 + A_{22} x_2 + B_2 u.\end{aligned}$$

Solving the second equation for x_2 and inserting it into the first equation yield

$$\begin{aligned}\dot{x}_1 &= \left(W^T A V - A_{12} A_{22}^{-1} A_{21} \right) x_1 \\ &\quad + \left(W^T B - A_{12} A_{22}^{-1} B_2 \right) u.\end{aligned}$$

For the output equation, it follows

$$\tilde{y} = \left(C V - C_2 A_{22}^{-1} A_{21} \right) x_1 + \left(D - C_2 A_{22}^{-1} B_2 \right) u.$$

This ROM makes use of the information in the matrices A_{12} , A_{21} , A_{22} , B_2 , and C_2 discarded by balanced truncation. It fulfills $\tilde{G}(0) = G(0)$ and the error bound (11); moreover, it preserves stability.

Besides SPA, another related truncation method that is not based on projection is *optimal Hankel norm approximation* (HNA). The description of HNA is technically quite involved; for details, see Zhou et al. (1996) and Glover (1984). It should be noted that the so obtained ROM usually has similar stability and accuracy properties as for balanced truncation.

Interpolation-Based Methods

Another family of methods for MOR is based on (rational) interpolation. The unifying feature of the methods in this family is that the original TFM (2) is approximated by a rational matrix function of lower degree satisfying some interpolation conditions (i.e., the original and the reduced-order TFM coincide, e.g., $G(s_0) = \tilde{G}(s_0)$ at some predefined value s_0 for which $A - s_0 E$ is nonsingular). Computationally this is usually realized by certain Krylov subspace methods.

The classical approach is known under the name of *moment-matching* or *Padé(-type) approximation*. In these methods, the transfer functions of the original and the ROMs are expanded into power series, and the ROM is then determined so that the first coefficients in the series expansions match. In this context, the coefficients of the power series are called moments, which explains the term “moment-matching”. One speaks of Padé-approximation if the number of matching moments is maximized for a given degree of the approximating rational function.

Classically, the expansion of the TFM (2) in a power series about an expansion point s_0

$$G(s) = \sum_{j=0}^{\infty} M_j(s_0)(s - s_0)^j \quad (12)$$

is used. The moments $M_j(s_0)$, $j = 0, 1, 2, \dots$, are given by

$$M_j(s_0) = -C [(A - s_0 E)^{-1} E]^j (A - s_0 E)^{-1} B.$$

Consider the (block) Krylov subspace

$$\mathcal{K}_k(F, H) = \text{span}\{H, FH, F^2H, \dots, F^{k-1}H\}$$

for $F = (A - s_0 E)^{-1} E$ and $H = -(A - s_0 E)^{-1} B$ with an appropriately chosen expansion point s_0 which may be real or complex. From the definitions of A , B , and E , it follows that $F \in \mathbb{K}^{n \times n}$ and $H \in \mathbb{K}^{n \times m}$, where $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$ depending on whether s_0 is chosen in $\mathbb{R}$ or in $\mathbb{C}$. Considering $\mathcal{K}_k(F, H)$ column by column,

this leads to the observation that the number of column vectors in $\{H, FH, F^2H, \dots, F^{k-1}H\}$ is given by $r = m \cdot k$, as there are k blocks $F^jH \in \mathbb{K}^{n \times m}$, $j = 0, \dots, k-1$. In the case when all r column vectors are linearly independent, the dimension of the Krylov subspace $\mathcal{K}_k(F, H)$ is $m \cdot k$. Assume that a unitary basis for this block Krylov subspace is generated such that the column space of the resulting unitary matrix $V \in \mathbb{C}^{n \times r}$ spans $\mathcal{K}_k(F, G)$. Applying the Galerkin projection $\Pi = VV^*$ to (1) yields a ROM whose TFM satisfies the following (Hermite) interpolation conditions at s_0 :

$$\tilde{G}^{(j)}(s_0) = G^{(j)}(s_0), \quad j = 0, 1, \dots, k-1.$$

That is, the first $k-1$ derivatives of G and $\tilde{G}$ coincide at s_0 . Considering the power series expansion (12) of the original and the reduced-order TFM, this is equivalent to saying that at least the first k moments $\tilde{M}_j(s_0)$ of the transfer function $\tilde{G}(s)$ of the ROM (3) are equal to the first k moments $M_j(s_0)$ of the TFM $G(s)$ of the original system (1) at the expansion point s_0 :

$$M_j(s_0) = \tilde{M}_j(s_0), \quad j = 0, 1, \dots, k-1.$$

If further the r columns of the unitary matrix W span the block Krylov subspace $\mathcal{K}_k(F, H)$ for $F = (A - s_0E)^{-T}E$ and $H = -(A - s_0E)^{-T}C^T$, applying the Petrov-Galerkin projection $\Pi = V(W^*V)^{-1}W^*$ to (1) yields a ROM whose TFM matches at least the first $2k$ moments of the TFM of the original system.

Theoretically, the matrix V (and W) can be computed by explicitly forming the columns which span the corresponding Krylov subspace $\mathcal{K}_k(F, H)$ and using the Gram-Schmidt algorithm to generate unitary basis vectors for $\mathcal{K}_k(F, H)$. The forming of the moments (the Krylov subspace blocks F^jH) is numerically precarious and has to be avoided under all circumstances. Using Krylov subspace methods to achieve an interpolation-based ROM as described above is recommended. The unitary basis of a (block) Krylov subspace can be computed by employing a (block) Arnoldi or (block) Lanczos method; see, e.g., Antoulas

(2005), Golub and Van Loan (2013), and Freund (2003).

In the case when an oblique projection is to be used, it is not necessary to compute two unitary bases as above. An alternative is then to use the nonsymmetric Lanczos process (Golub and Van Loan 2013). It computes bi-unitary (i.e., $W^*V = I_r$) bases for the abovementioned Krylov subspaces and the ROM as a by-product of the Lanczos process. An overview of the computational techniques for moment-matching and Padé approximation summarizing the work of a decade is given in Freund (2003) and the references therein.

In general, the discussed MOR approaches are instances of rational interpolation. When the expansion point is chosen to be $s_0 = \infty$, the moments are called Markov parameters, and the approximation problem is known as partial realization. If $s_0 = 0$, the approximation problem is known as Padé approximation.

As the use of one single expansion point s_0 leads to good approximation only close to s_0 , it might be desirable to use more than one expansion point. This leads to multipoint moment-matching methods, also called rational Krylov methods; see, e.g., Ruhe and Skoogh (1998), Antoulas (2005), and Freund (2003).

In contrast to balanced truncation, these (rational) interpolation methods do not necessarily preserve stability. Remedies have been suggested; see, e.g., Freund (2003).

The use of complex-valued expansion points will lead to a complex-valued ROM (3). In some applications (in particular, in case the original system is real valued), this is undesired. In that case one can always use complex-conjugate pairs of expansion points as then the entire computations can be done in real arithmetic.

The methods just described provide good approximation quality locally around the expansion points. They do not aim at a global approximation as measured by the $\mathcal{H}_2$ - or $\mathcal{H}_\infty$ -norms. In Gugercin et al. (2008), an iterative procedure is presented which determines locally optimal expansion points w.r.t. the $\mathcal{H}_2$ -norm approximation under the assumption that the order r of the ROM is prescribed and only 0th-

and 1st-order derivatives are matched. Also, for multi-input multi-output systems (i.e., m and p in (1) are both larger than one), no full moment matching is achieved but only tangential interpolation, $G(s_j)b_j = \tilde{G}(s_j)b_j$, $c_j^*G(s_j) = c_j^*\tilde{G}(s_j)$, $c_j^*G'(s_j)b_j = c_j^*\tilde{G}'(s_j)b_j$, for certain vectors b_j, c_j determined together with the optimal s_j by the iterative procedure.

Tools

Almost all commercial software packages for structural dynamics include modal analysis/truncation as a means to compute a ROM. Modal truncation and balanced truncation are available in the MATLAB® Control System Toolbox and the MATLAB® Robust Control Toolbox.

Numerically reliable, well-tested, and efficient implementations of many variants of balancing-based MOR methods as well as Hankel norm approximation and singular perturbation approximation can be found in the Subroutine Library In Control Theory (SLICOT, <http://www.slicot.org>) (Varga 2001). Easy-to-use MATLAB interfaces to the Fortran 77 subroutines from SLICOT are available in the SLICOT Model and Controller Reduction Toolbox (<http://slicot.org/matlab-toolboxes/basic-control>); see Benner et al. (2010). An implementation of moment-matching via the (block) Arnoldi method is available in MOR for ANSYS® (<http://modelreduction.com/Software.html>).

The MORLAB toolbox (<https://www.mpi-magdeburg.mpg.de/projects/morlab>) is a collection of MATLAB routines for different truncation-based MOR methods for LTI systems as well as for DAE and second-order systems.

The MOR Wiki (<https://morwiki.mpi-magdeburg.mpg.de/morwiki/>) is a platform for MOR research and provides discussions of a number of methods, as well as links to further software packages (e.g., MOREMBS, MORPACK, and pyMOR). Moreover, one can find an extensive set of benchmark examples of LTI systems from various applications given in

computer-readable format which can easily be used to test new algorithms and software.

Summary and Future Directions

MOR of LTI systems can now be considered as an established computational technique. Some open issues still remain and are currently investigated. These include methods yielding good approximation in finite frequency or time intervals. Though numerous approaches for these tasks exist, methods with sharp local error bounds are still under investigation. A related problem is the reduction of closed-loop systems and controller reduction. The generalization of the methods discussed in this essay to second-order systems of the form

$$\begin{aligned} M\ddot{x}(t) + D\dot{x}(t) + Kx(t) &= Fu(t), y(t) \\ &= C_px(t) + C_v\dot{x}(t) \end{aligned}$$

with $M, K, D \in \mathbb{R}^{s \times s}$, $F \in \mathbb{R}^{s \times m}$, $C_p, C_v \in \mathbb{R}^{q \times s}$ has only been partially achieved. Of course, linearization to an equivalent first-order system is always possible. But, in general, the ROM in first-order form cannot be transformed into a ROM in second-order form. An important problem class getting a lot of current attention consists of (uncertain) parametric systems, that is, systems of the form

$$\begin{aligned} E(p)\dot{x}(t; p) &= A(p)x(t, p) + B(p)u(t), y(t, p) \\ &= C(p)x(t, p) \end{aligned}$$

where all system matrices (and hence the state and the output) depend on d parameters $p \in \mathbb{R}^d$. While the system is linear in the state, all system matrices may depend nonlinearly on the parameters. Here it is important to preserve parameters as symbolic quantities in the ROM. Most of the current approaches are based in one way or another on interpolation; a recent survey can be found in Benner et al. (2015). MOR for nonlinear systems has also been a research topic for decades. Here often, sampling-based methods such as proper orthogonal decomposition or, in case of parameter-dependent problems, reduced

basis methods are employed. Tutorial-style introductions with references to further reading can be found in Benner et al. (2017). Still, the development of satisfactory methods in the context of control design having computable error bounds and preserving interesting system properties remains a challenging task.

Cross-References

- [Basic Numerical Methods and Software for Computer Aided Control Systems Design](#)
- [Multi-domain Modeling and Simulation](#)

Bibliography

- Antoulas A (2005) Approximation of large-scale dynamical systems. SIAM, Philadelphia
- Benner P (2006) Numerical linear algebra for model reduction in control and simulation. *GAMM Mitt* 29(2):275–296
- Benner P, Stykel T (2017) Model order reduction for differential-algebraic equations: a survey. In: Ilchmann A, Reis T (eds) *Surveys in Differential-algebraic Equations IV, Differential-Algebraic Equations Forum*. Springer International Publishing, Cham, pp 107–160
- Benner P, Quintana-Ortí E, Quintana-Ortí G (2000) Balanced truncation model reduction of large-scale dense systems on parallel computers. *Math Comput Model Dyn Syst* 6:383–405
- Benner P, Mehrmann V, Sorensen D (2005) Dimension reduction of large-scale systems. *Lecture Notes in Computational Science and Engineering*, vol 45. Springer, Berlin/Heidelberg
- Benner P, Kressner D, Sima V, Varga A (2010) Die SLICOT-Toolboxen für Matlab (The SLICOT-Toolboxes for Matlab) [German]. *at-Automatisierungstechnik* 58(1):15–25. English version available as SLICOT working note 2009-1, 2009. <http://slicot.org/working-notes/>
- Benner P, Hochstenbach M, Kürschner P (2011) Model order reduction of large-scale dynamical systems with Jacobi-Davidson style eigensolvers. In: *Proceedings of the International Conference on Communications, Computing and Control Applications (CCCA)*, 3–5 Mar 2011 at Hammamet. IEEE Publications, p 6
- Benner P, Gugercin S, Willcox K (2015) A survey of model reduction methods for parametric systems. *SIAM Rev* 57(4):483–531
- Benner P, Cohen A, Ohlberger M, Willcox K (2017) Model reduction and approximation. *Theory and algorithms*. SIAM, Philadelphia
- Freund R (2003) Model reduction methods based on Krylov subspaces. *Acta Num* 12:267–319
- Glover K (1984) All optimal Hankel-norm approximations of linear multivariable systems and their L^∞ norms. *Internat J Control* 39:1115–1193
- Golub G, Van Loan C (2013) *Matrix computations*, 4th edn. Johns Hopkins University Press, Baltimore
- Gugercin S, Antoulas AC, Beattie C (2008) $\mathcal{H}_2$ model reduction for large-scale dynamical systems. *SIAM J Matrix Anal Appl* 30(2):609–638
- Obinata G, Anderson B (2001) *Model reduction for control system design*. Communications and Control Engineering Series. Springer, London
- Ruhe A, Skoogh D (1998) Rational Krylov algorithms for eigenvalue computation and model reduction. In: *Applied parallel computing. Large scale scientific and industrial problems*. Lecture Notes in Computer Science, vol 1541. Springer, Berlin/Heidelberg, pp 491–502
- Schilders W, van der Vorst H, Rommes J (2008) *Model order reduction: theory, research aspects and applications*. Springer, Berlin/Heidelberg
- Varga A (1991) Balancing-free square-root algorithm for computing singular perturbation approximations. In: *Proceedings of the 30th IEEE CDC*, Brighton, pp 1062–1065
- Varga A (1995) Enhanced modal approach for model reduction. *Math Model Syst* 1(2):91–105
- Varga A (2001) Model reduction software in the SLICOT library. In: Datta B (ed) *Applied and computational control, signals, and circuits*. The Kluwer International Series in Engineering and Computer Science, vol 629. Kluwer Academic, Boston, pp 239–282
- Zhou K, Doyle J, Glover K (1996) *Robust and optimal control*. Prentice Hall, Upper Saddle River

Model Predictive Control for Power Networks

Ian A. Hiskens¹ and Maria Vrakopoulou²

¹University of Michigan, Ann Arbor, MI, USA

²Electrical and Electronic Engineering, University of Melbourne, Melbourne, VIC, Australia

Abstract

Large-scale power systems rely on a hierarchical control structure to operate reliably through a wide variety of disturbances.

As operating strategies adapt to incorporate emerging technologies and distributed resources such as renewable generation and storage, model predictive control (MPC) offers a systematic approach to coordinating diverse controls and providing decision support. MPC uses an approximate system model to determine an optimal control sequence over a finite horizon. The first instance of that control sequence is implemented on the system, the resulting system response is measured, and the process repeats. Power system applications of MPC include prevention of voltage instability, corrective control, and secondary frequency control.

Keywords

Model predictive control · Power system control · Power system dynamics

Introduction

Electrical power systems are large-scale, geographically spread systems that are continually responding to a wide variety of disturbances. Reliable operation of such systems depends upon a hierarchical control structure, from very fast fault protection action to system-wide control centers where operators continually monitor behavior and enact control decisions.

As new technologies emerge and greater investment is directed toward distributed renewable energy and storage assets, traditional operating strategies must adapt accordingly. Faced with a large number of distributed devices, system operators will find it ever more challenging to quickly identify the most useful control actions during system events. Model predictive control (MPC) offers a systematic approach to coordinating a variety of control devices while operating quickly and reliably. Hence, as systems operate closer and closer to their limits, the feedback control actions and decision support provided by MPC will play an invaluable role in responding to stressed network conditions.

Overview of Power System Control

Power systems are affected by disturbances that range from lightning-induced faults on transmission lines to fluctuations in renewable power generation. In the former case, protection associated with the vulnerable components operates almost instantly to prevent, or at least minimize, damage (Blackburn 1998). Because of the requirement for high response speed, decision-making is minimal. Such myopic behavior may, however, weaken the network, further exacerbating the disturbance and potentially initiating an uncontrollable cascade of outages. The blackout of the USA and Canada in August 2003 provides such an example (U.S.-Canada Power System Outage Task Force 2004).

At a slower time-scale, feedback control loops are employed to regulate transmission and distribution voltages, generator active power production, and system frequency (Kundur 1994). Voltage regulation is achieved through application of automatic voltage regulators (AVRs) on generators, tap-changing transformers, and a growing list of power-electronic devices. Governors are used to regulate generator active power. Both AVRs and governors are local feedback control loops, i.e., they measure and respond only to local conditions. In contrast, system frequency regulation is achieved through automatic generator control (AGC), also known as load frequency control (LFC), which is a feedback control strategy that adjusts generators that are widespread across the power system (Jaleeli et al. 1992).

System operators continually monitor power system behavior and enact higher-level control decisions. For example, in response to a transmission line being overloaded, i.e., the current on the line exceeding its allowable maximum, operators may redispatch generation and/or curtail load. However, they might decide to allow the line to remain temporarily overloaded based on forecast changes in renewable generation that will soon relieve the overload. Operator decision-making is becoming more complicated, though, with increasing reliance on distributed generation and storage, and newer technologies such as

dynamic line rating (Douglass and Edris 1996) and highly responsive load control (Callaway and Hiskens 2011). MPC offers a systematic framework for real-time feedback control and/or decision support for power systems that are operating under stressed conditions.

Model Predictive Control (MPC)

MPC uses an internal (approximate) model of the system to predict behavior and establish an optimal control sequence (Rawlings 2000). The control actions from the first step of this sequence are implemented on the actual system. Subsequent measurement of the resulting system behavior provides the initial conditions for MPC to again predict behavior and recalculate an optimal control sequence. The feedback inherent in the repetition of this process allows MPC to control a broad range of devices while effectively satisfying a multi-period constrained optimal power flow problem.

The control actions available to MPC depend upon the application, with further discussion in section “Applications.” In general, though, MPC has the potential to redispatch generation and storage; exercise demand response; adjust set-points for voltage regulating devices such as generators, tap-changing transformers, and Statcoms/SVCs; and adjust angle set-points for phase-shifting transformers.

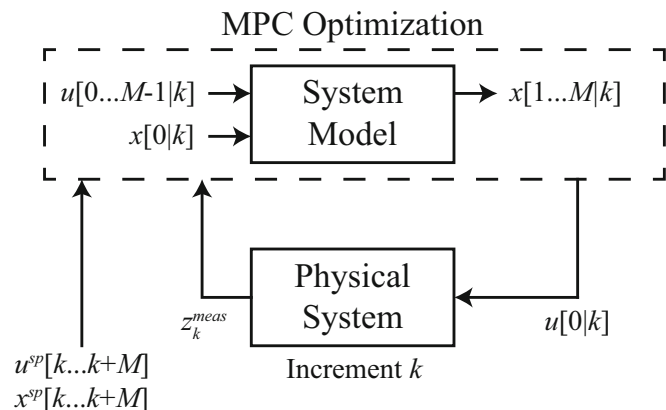
Figure 1 outlines the MPC control strategy. At time k , all measurements z_k^{meas} necessary to

model the system are provided to the controller. These measurements include network configuration, voltage magnitudes, active/reactive power of generators and loads, active/reactive power flows on lines, transformer tap positions, storage state-of-charge, and transmission line temperatures. This information is obtained via SCADA and synchrophasor measurement data monitoring and harmonized through state estimation (Abur and Gómez-Expósito 2004). The power system state estimation process uses nonlinear least-squares to minimize the weighted error between the measurements and the corresponding quantities of the network model. This yields the estimated network state which consists of voltage magnitudes and angles at all nodes across the network. The initial conditions for the MPC system model $x[0|k]$ are derived from the voltage state of the network model, together with other measurements as required for particular applications.

MPC then builds an optimization problem to determine a control sequence $u[0 \dots M-1|k]$ over a horizon of M time-steps while considering its effects on states $x[1 \dots M|k]$. The control sequence is computed to track scheduled/forecast set-point values for both controlled quantities $u^{sp}[k \dots k+M-1]$, such as conventional and renewable generation and controllable demand, and states $x^{sp}[k+1 \dots k+M]$ over the prediction horizon. Forecasts for other exogenous quantities may also be required for certain applications. For example, weather forecasts across the geographical expanse of a power system are important when transmission line ratings are dependent

Model Predictive Control for Power Networks,

Fig. 1 Overview of MPC strategy, from Martin and Hiskens (2017)



upon the thermal characteristics of line conductors, or when load control actions include thermostatically controlled loads (TCLs).

Once an optimal control sequence has been computed, the controls from the first step of that sequence $u[0|k]$ are applied to the physical system. The physical system responds to the controls as time advances from k to $k + 1$, and the process repeats when new system measurements z_{k+1}^{meas} become available.

Power system behavior is inherently nonlinear, with implications for MPC modelling and computational complexity. Care must be taken to develop approximate models that are sufficiently accurate to capture the phenomena of interest, yet still give tractable MPC optimization formulations that provide convergence and performance guarantees (Molzahn and Molzahn 2019). Linear models are most appealing, of course, with the “DC power flow” model forming the basis for various MPC formulations. This simplification of the power flow provides reasonable accuracy for active power flow over transmission lines but does not capture reactive power flows nor voltage magnitudes. Resulting MPC formulations are therefore blind to abnormal bus voltages and situations where high reactive power flow contributes to line overloading.

Other MPC strategies use the Jacobian which is obtained by linearizing the nonlinear power flow equations about the current operating point. The justification is that MPC introduces relatively small changes; therefore, the linearized model is sufficient to describe the resulting behavior. Typically the linearization is repeated at each MPC cycle. MPC formulations have also made use of trajectory sensitivities, which describe (to first order) the change in a trajectory due to perturbations in initial conditions. In this case, the nominal trajectory is computed using a nonlinear system model, with trajectory sensitivities (approximately) describing perturbations in the trajectory due to changes in the control variables.

Incorporating large-scale power systems into multi-period optimization gives high-dimensional problems that may incur unacceptably long solution times and exhibit numerical

instability. However, power system disturbances, such as line outages, tend to impact localized regions of power systems and require regional control actions. Therefore, while it is important for the entire system to be monitored for disturbances, there is no need for MPC to always model the system in its entirety. Rather, MPC should adaptively determine and model the disturbed subsystem, with the composition of that subsystem chosen to ensure that the impact of the disturbance and the potential control actions are adequately captured. An approach based on line-flow sensitivities is presented in Martin and Hiskens (2017).

Applications

The first application of MPC to emergency control of power systems was (Larsson et al. 2002), where voltage stability was achieved through optimal coordination of load shedding, capacitor switching, and tap-changer operation. A tree-based search method was employed to obtain optimal control actions from discrete switching events. Later MPC strategies for alleviating voltage instability employed nonlinear power system models and used trajectory sensitivities to approximate the impact of control actions on system behavior (Zima and Andersson 2005; Hiskens and Gong 2005; Jin et al. 2008).

Early investigations into the use of MPC for relieving thermal overloading of transmission lines were undertaken in Otomega et al. (2007). To return line currents to within their long-term ratings following a contingency, MPC regulated the angle on phase-shifting transformers, redispatched generation, and shed load as a last resort. The study found that both the length of the prediction horizon and the accuracy of the network model can have an appreciable influence on the ability of MPC to meet its control objectives. The work presented in Carneiro and Ferrarini (2010) further expanded on those ideas by incorporating a temperature-based model of transmission lines. Enforcing a temperature limit on a transmission line

while considering actual ambient conditions provides greater flexibility than the conservative assumptions typically used in device ratings. The shift toward distributed renewable generation and energy storage technologies has motivated further investigation of MPC-based cascade mitigation. In Martin and Hiskens (2017); Almassalkhi and Hiskens (2015), brief temperature overloads are relieved using energy storage and curtailment of renewables in addition to the control measures utilized in Otomega et al. (2007); Carneiro and Ferrarini (2010). An approach to extending control to large networks was developed and demonstrated in Martin and Hiskens (2017) using a 4259-bus model of the Californian electricity network.

MPC has also been explored for automatic generation control (AGC). AGC is used in most interconnected power systems to restore system frequency to its set-point value and regulate tie-line interchange (Jaleeli et al. 1992). These objectives are achieved by adjusting the active power set-points of generators throughout the system, taking into account restrictions on the amount and rate of generator power deviations. To cope with the expansive nature of power systems, a distributed control structure has been adopted for AGC. The current form of AGC may not, however, be well suited to future power systems where greater utilization of intermittent renewable resources brings with it power flow fluctuations that are difficult to regulate. Distributed MPC offers an effective means of achieving the desired controller coordination and performance improvements while alleviating the organizational and computational burden associated with centralized control (Venkat et al. 2008).

Summary and Future Directions

MPC offers a systematic approach to coordinating diverse power system resources and providing decision support. Power system applications of MPC range from prevention of voltage collapse and cascading failures in large-scale

power systems to optimally regulating energy storage in local energy hubs. As power systems grow in complexity, the predictive capability of MPC will become increasingly important for ensuring appropriate response following large disturbances. Furthermore, the structural characteristics of power systems suggest a greater role for distributed forms of MPC.

Cross-References

- [Active Power Control of Wind Power Plants for Grid Integration](#)
- [Cascading Network Failure in Power Grid Blackouts](#)
- [Lyapunov Methods in Power System Stability](#)
- [Power System Voltage Stability](#)

Recommended Reading

Excellent overviews of MPC are provided by Rawlings (2000); Camacho and Bordons (2004), while Scattolini (2009), Negenborn and Maestre (2014) provide thorough reviews of distributed MPC. Power system applications of non-centralized MPC are presented in Hermans et al. (2012), and a comparison of various implementation schemes is undertaken.

References

- Abur A, Gómez-Expósito A (2004) Power system state estimation. Marcel Dekker, New York
- Almassalkhi M, Hiskens I (2015) Model-predictive cascade mitigation in electric power systems with storage and renewables—Part I: theory and implementation. *IEEE Trans Power Syst* 30(1):67–77
- Blackburn J (1998) Protective relaying principles and applications, 2nd edn. Marcel Dekker, New York
- Callaway D, Hiskens I (2011) Achieving controllability of electric loads. *Proc IEEE* 99(1):184–199
- Camacho E, Bordons C (2004) Model predictive control, 2nd edn. Springer, New York

- Carneiro J, Ferrarini L (2010) Preventing thermal overloads in transmission circuits via model predictive control. *IEEE Trans Control Syst Technol* 18(6): 1406–1412
- Douglass D, Edris A (1996) Real-time monitoring and dynamic thermal rating of power transmission circuits. *IEEE Trans Power Delivery* 11(3):1407–1418
- Hermans R, Jokić A, Alessio A, van den Bosch P, Hiskens I, Bemporad A (2012) Assessment of non-centralised model predictive control techniques for electrical power networks. *Int J Control* 85(8): 1162–1177
- Hiskens I, Gong B (2005) MPC-based load shedding for voltage stability enhancement. In: *Proceedings of the 44th IEEE conference on decision and control*, Seville, pp 4463–4468
- Jaleeli N, VanSlyck L, Ewart D, Fink L, Hoffmann A (1992) Understanding automatic generation control. *IEEE Trans Power Syst* 7(3):1106–1122
- Jin L, Kumar R, Elia N (2008) Security constrained emergency voltage stabilization: a model predictive control based approach. In: *Proceedings of the 47th IEEE conference on decision and control*, Cancun, pp 2469–2474
- Kundur P (1994) *Power system stability and control*. EPRI power system engineering series. McGraw Hill, New York
- Larsson M, Hill D, Olsson G (2002) Emergency voltage control using search and predictive control. *Electr Power Energy Syst* 24:121–130
- Martin J, Hiskens I (2017) Corrective model-predictive control in large electric power systems. *IEEE Trans Power Syst* 32(2):1651–1662
- Molzahn D, Molzahn I (2019) A survey of relaxations and approximations of the power flow equations. *Found Trends Electr Energy Syst* 4(1–2):1–221
- Negenborn R, Maestre J (2014) Distributed model predictive control: an overview and roadmap of future research. *IEEE Control Syst Mag* 34(4):87–97
- Otomega B, Marinakis A, Glavic M, Van Cutsem T (2007) Model predictive control to alleviate thermal overloads. *IEEE Trans Power Syst* 22(3):1384–1385 (2007)
- Rawlings J (2000) Tutorial overview of model predictive control. *IEEE Control Syst Mag* 20(3):38–52
- Scattolini R (2009) Architectures for distributed and hierarchical model predictive control – a review. *J Process Control* 19:723–731
- U.S.-Canada Power System Outage Task Force (2004) *Final report on the August 14, 2003 blackout in the United States and Canada: causes and recommendations*
- Venkat A, Hiskens I, Rawlings J, Wright S (2008) Distributed MPC strategies with application to power system automatic generation control. *IEEE Trans Control Syst Technol* 16(6):1192–1206
- Zima M, Andersson G (2005) Model predictive control employing trajectory sensitivities for power systems applications. In: *Proceedings of the 44th IEEE conference on decision and control*, Seville, pp 4452–4456

Model Predictive Control in Practice

Thomas A. Badgwell¹ and S. Joe Qin²

¹ExxonMobil Research and Engineering, Annandale, NJ, USA

²University of Southern California, Los Angeles, CA, USA

Abstract

Model predictive control (MPC) refers to a class of computer control algorithms that utilize an explicit mathematical model to optimize the predicted behavior of a process. At each control interval, an MPC algorithm computes a sequence of future process adjustments that optimize a specified control objective. The first adjustment is implemented and then the calculation is repeated at the next control cycle. Originally developed to meet the particular needs of petroleum refinery and power plant control problems, MPC technology has evolved significantly in both capability and scope and can now be found in many other control application domains.

Keywords

Predictive control · Computer control · Mathematical programming

Introduction

Model predictive control (MPC) refers to a class of computer control algorithms that utilize an explicit mathematical model to predict future process behavior. At each control interval, in the most general case, an MPC algorithm solves a sequence of nonlinear programs to answer three essential questions: where is the process heading (prediction), where should the process go (steady-state target optimization), and what is the best sequence of adjustments to send it to the right place (dynamic optimization)? The first adjustment is then implemented, and then the

entire calculation sequence is repeated at the subsequent control cycles.

MPC technology arose first in the context of petroleum refinery and power plant control problems for which the well-known linear quadratic Gaussian (LQG) methods were found to be inadequate (Richalet et al. 1978; Cutler and Ramaker 1979). Specific needs that drove the development of MPC technology include the need for economic optimization and strict enforcement of safety and equipment constraints. Promising early results led to a wave of successful industrial applications, sparking the development of several commercial offerings (Qin and Badgwell 2003) and generating intense interest from the academic community (Mayne et al. 2000). Today MPC technology permeates the refining and chemical industries and has gained increasing acceptance in a wide variety of areas including chemicals, automotive, aerospace, and food processing applications. The total number of MPC applications in the petrochemical industry was estimated in 2010 to be 5,000–10,000 with total benefits of 1–2 billion \$/year (Badgwell 2010).

This entry focuses on how MPC is implemented in refining and chemical plant applications. A more detailed summary of the history of MPC technology development, as well as a survey of commercial offerings through 2003, can be found in the review article by Qin and Badgwell (2003). Darby and Nikolaou present a more recent summary of MPC practice (Darby and Nikolaou 2012). Textbook descriptions of MPC theory and design, suitable for classroom

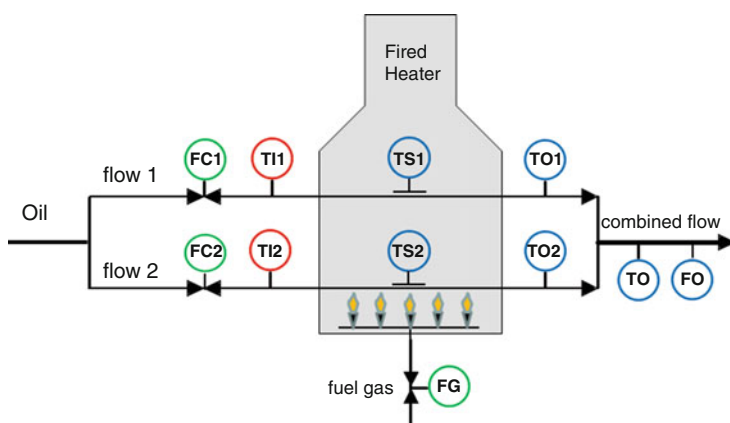
use, include Maciejowski (2002) and Rawlings et al. (2017).

MPC Fired Heater Example

An example from petroleum refining will be used to introduce basic nomenclature and concepts and illustrates many of the issues that arise in the practice of MPC. Consider the simplified fired heater in Fig. 1. A feed stream of cold oil splits into two tubes that pass over burners fired by fuel gas. The tubes then recombine to produce a heated oil stream that continues on for further processing. The circles indicate measurements and flow controllers most relevant to heater operation. These fall into three basic groups. The green group, known as manipulated variables (MVs), correspond to variables that the MPC algorithm (or a human operator) may adjust to achieve the desired control objectives. These are the setpoint of the tube 1 flow controller FC1, the setpoint of the tube 2 flow controller FC2, and the setpoint of the fuel gas flow controller FG. The red group, known as disturbance variables (DVs), are measurements that provide feedforward information for the control algorithm. Here the DVs are the two tube oil inlet temperatures TI1 and TI2. The blue group, known as controlled variables (CVs), are measurements of the quantities that must be controlled. This includes the external skin temperature of each tube TS1 and TS2, the oil outlet temperature for each tube TO1 and TO2, and the combined oil outlet temperature TO and fuel gas outlet temperature FO.

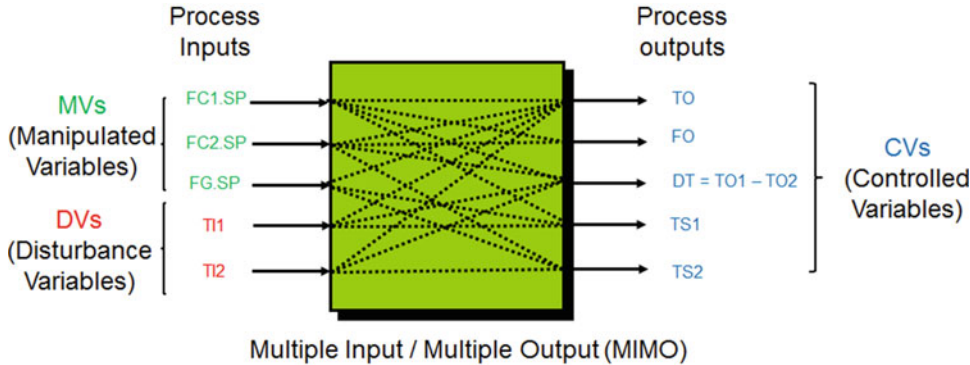
Model Predictive Control in Practice,

Fig. 1 Simplified fired heater warms up an oil stream by passing it over a set of burners fired by fuel gas. Manipulated variables (MVs), disturbance variables (DVs), and controlled variables (CVs) are highlighted



Model Predictive Control in Practice, Table 1 Fired heater MPC design specification

Variable	Description (Units)	Type	Priority	Specification
TS1	Tube 1 skin temperature (F)	CV	1	Max. limit
TS2	Tube 2 skin temperature (F)	CV	1	Max. limit
TO	Combined outlet temperature (F)	CV	2	Setpoint
FO	Combined outlet flow rate (BPH)	CV	3	Setpoint
TO1	Tube 1 outlet temperature (F)	–	–	–
TO2	Tube 2 outlet temperature (F)	–	–	–
DT	Delta temperature TO1–TO2 (F)	CV	4	Setpoint (0)
FC1.SP	Flow controller tube 1 setpoint (BPH)	MV	–	Max/Min/ROC
FC2.SP	Flow controller tube 2 setpoint (BPH)	MV	–	Max/Min/ROC
FG.SP	Fuel gas flow controller setpoint (MSCFH)	MV	–	Max/Min/ROC
TI1	Inlet temperature tube 1 (F)	DV	–	–
TI2	Inlet temperature tube 2 (F)	DV	–	–



Model Predictive Control in Practice, Fig. 2 Fired heater process inputs and output

and TO2, and the combined oil temperature TO and flow rate FO exiting the heater.

The fired heater MPC design specification is presented in Table 1, which lists the key process variables and corresponding specifications. A vital aspect of the MPC design is the requirement to prioritize the CVs. This is done to make sure that the control focuses first on the most important goals and then tries to achieve remaining goals in a prioritized order only if enough degrees of freedom are available. The highest priority specifications involve constraints associated with safety. Lower-level priorities typically include meeting desired product specifications and saving energy. For the fired heater example, the top priority is to make sure that tube skin temperatures TS1 and TS2 do not exceed a maximum safety limit. The second and third priorities are to achieve the desired combined

oil outlet temperature and flow rate. A fourth priority, when conditions allow, is to balance the tube outlet temperatures TO1 and TO2 in order to maximize heat transfer efficiency. For this purpose we define a delta temperature variable DT as the difference between the tube 1 and 2 oil outlet temperatures. Note that maximum, minimum, and rate of change limits are provided for each MV. These do not have an associated priority because they can always be enforced.

After laying out the MPC design specification, the next step is to develop a mathematical model that predicts the effects of MV and DV changes on the CVs. Figure 2 illustrates how the model can be viewed in terms of process inputs (MVs and DVs) and process outputs (CVs). Such a model is often referred to as multiple-input/multiple-output (MIMO). The vast majority of industrial MPC applications rely

on a linear empirical MIMO model developed from process step response data.

Figure 3 illustrates a process step test for the fired heater. The process inputs are plotted in the left column, and the process outputs are shown in the right column. Time in minutes is plotted on the horizontal axis. At the 10 min mark, we increase the fuel gas flow FG from 95 to 97 SCFH and then return it to 95 SCFH 10 min later. In the right column, the effects of this pulse on the CVs can be seen. The combined oil outlet temperature TO increases by a little more than 10 F and then comes back down again. The combined oil flow rate FO and the tube outlet temperature difference DT remain unchanged, while both tube skin temperatures TS1 and TS2 rise and then fall. With this information we can derive all of the models related to fuel gas flow FG. In a similar way, we make changes to the two tube flow controllers FC1 and FC2 and observe the responses of the CVs. Since we cannot directly change the tube inlet temperature DVs, we must either search historical data, or, as shown here, we assume that we are able to make changes to upstream equipment to increase the oil inlet temperature. In the ideal case shown here, we are able to make significant, independent changes to each process input, and we collect significant, essentially noise-free data that shows the effects on each output.

The step test data are then passed into appropriate modeling software to develop a dynamic model. The data must first be edited to remove effects that we do not want to include in the model, such as might happen during a process upset. The data are then typically band-pass filtered to remove the steady-state information and any high-frequency noise. A subspace identification method (Ljung 1999) can then be used to identify a linear state-space model. The step response of the resulting model is illustrated in Fig. 4. In this representation, the process inputs appear as columns, and the process outputs appear as rows. Each column shows how the outputs would respond to a pure unit step change increase in the given input. For example, the middle column shows how the CVs respond to a unit increase in the fuel

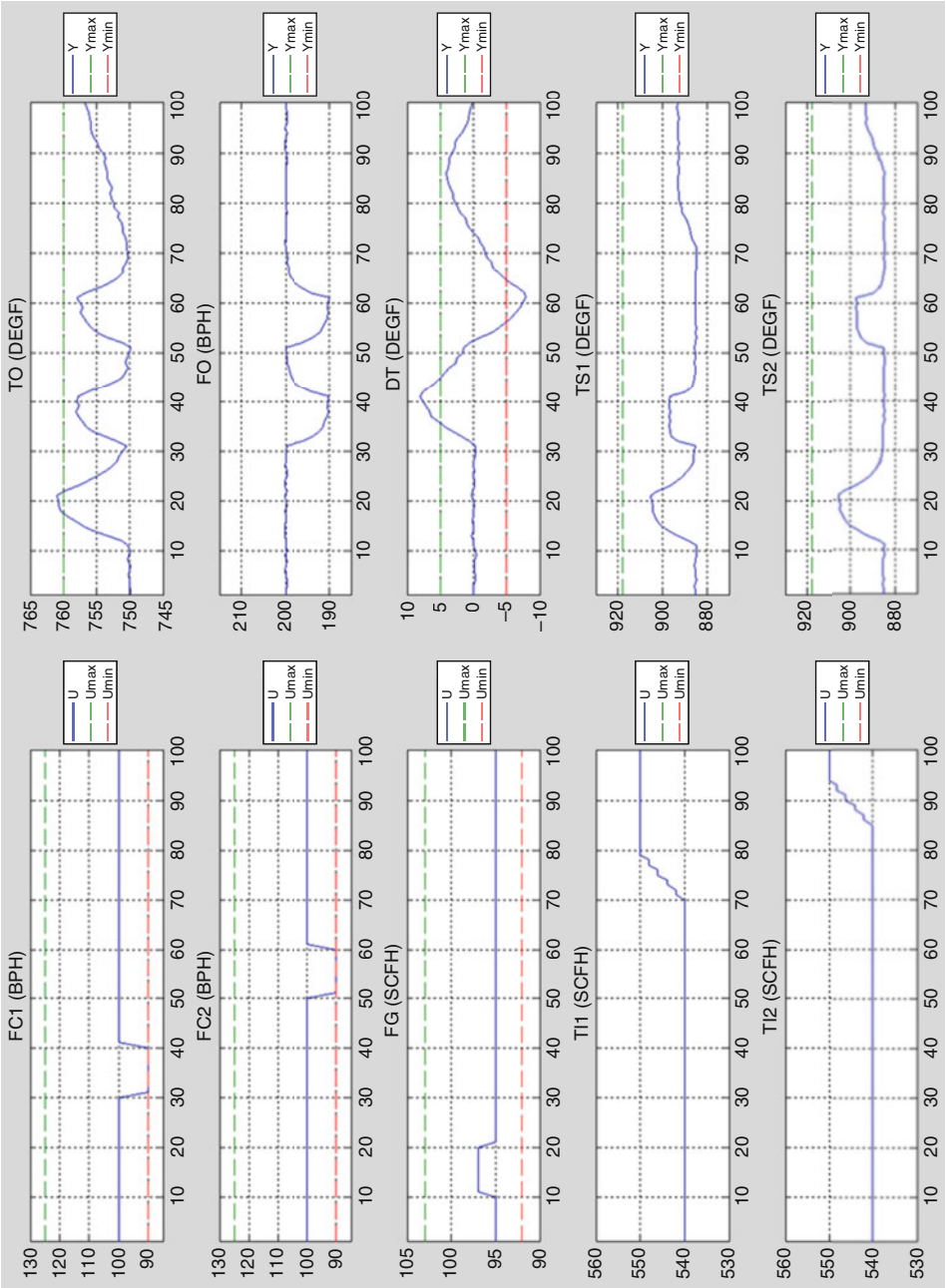
gas flow rate FG. The final value in each cell is known as the process gain. The first cell in the middle column shows that the combined outlet temperature TO increases by about 6 F, so the gain for this model is 6 DEGF/SCFH. The second and third cells show no changes for the combined outlet flow FO and the tub outlet delta temperature DT. The last two cells show that the tube skin temperatures both increase by about 10 F. Of course these responses match the data that was recorded during the step test.

Figure 5 shows how the fired heater MPC application is configured. The application runs at a 1-minute frequency in a dedicated PC, communicating with the process instrumentation through a standard protocol such as *OPC*. Every minute it reads in current values of the CVs and DVs and then sends an adjustment out to the MVs.

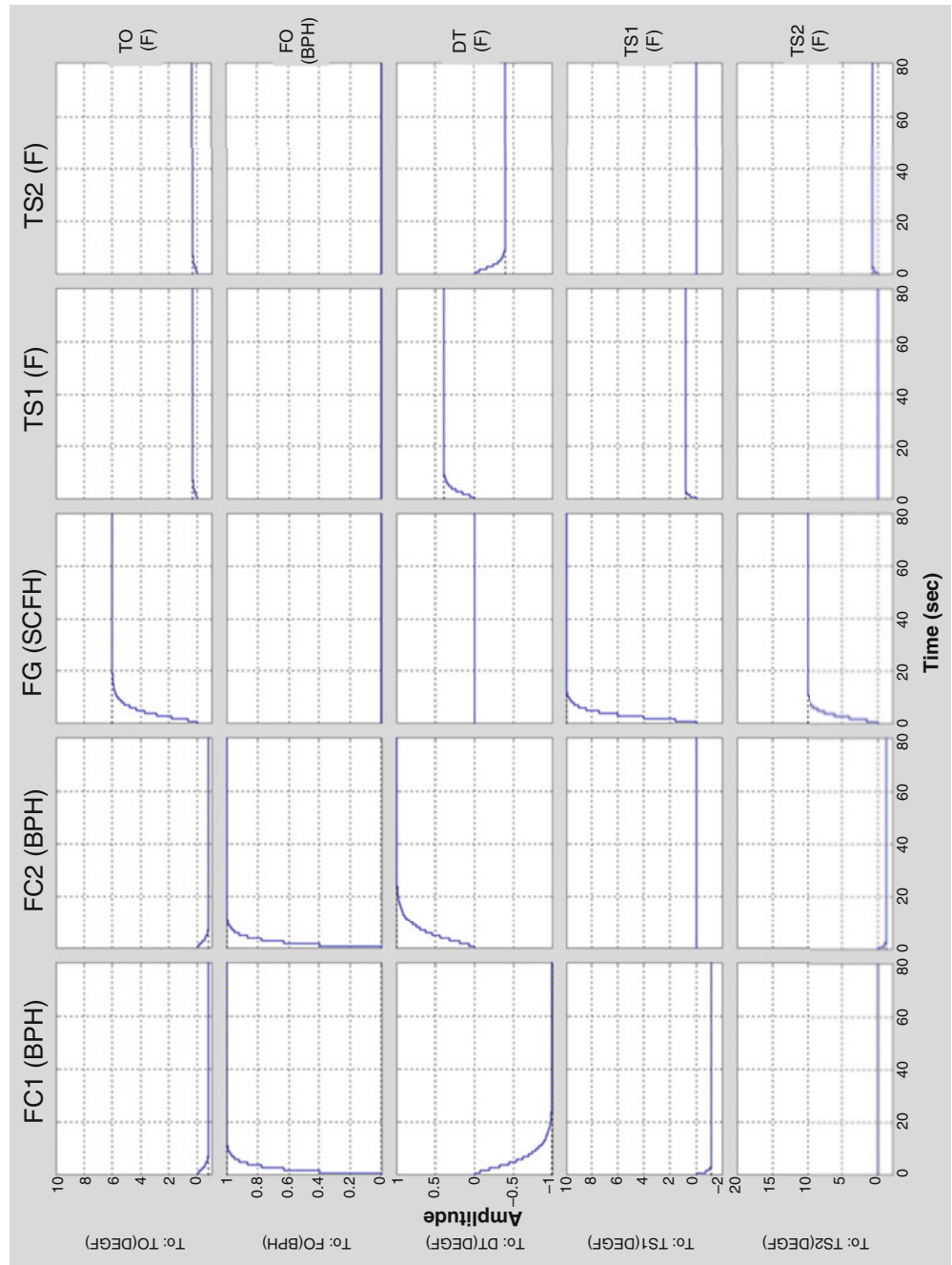
Results from a closed-loop test of the fired heater MPC are shown in Fig. 6. Again the process inputs are shown in the left column, and process outputs appear in the right column. The first change occurs at the 10-minute mark, when the operator increases the TO setpoint by 5 F. The control responds by sharply increasing FG from 95 to over 97 SCFH and then bringing it down to settle around 96 SCFH. This has the effect of increasing TO quickly up to the new setpoint of 755 F. There is, of course, no change in FO or DT, but TS1 and TS2 both increase as expected.

The next change comes at the 30 min mark when the operator asks to increase FO by 20 BPH without changing TO or DT. The MPC accomplishes this by increasing both FC1 and FC2 while also carefully increasing FG in such a way that TO remains constant.

Having successfully added 20 BPH of feed, the operator tries to do the same thing at the 50 min mark. This time, however, the MPC determines right away that it will not be able to reach the new setpoint of 240 BPH. The steady-state target for FO (green line) only goes up to around 235 BPH. MPC stops short on the FO because it predicts that it will run out of room on the tube skin temperatures TS1 and TS2. We have configured these limits as the top priority control specifications, so the MPC will not allow these limits to be violated. This places a limit on the

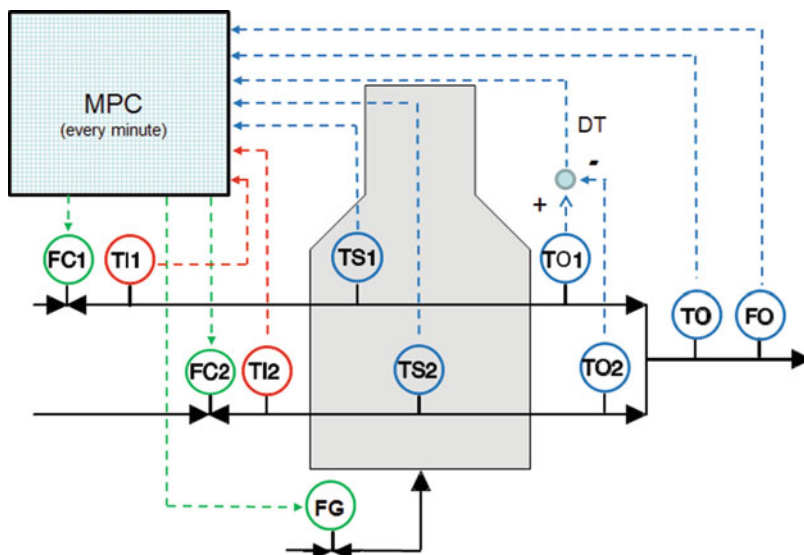


Model Predictive Control in Practice, Fig. 3 Fired heater step test



Model Predictive Control in Practice, Fig. 4 Fired heater step response model

Model Predictive Control in Practice, Fig. 5 Fired heater MPC configuration



available heat input, and we have said that the next highest priority is to maintain the heater outlet temperature TO, so the control gives up on the FO setpoint. This illustrates how priorities are used to resolve conflicting control objectives in a MPC application.

At the 70 min mark, we can see that the oil entering the first tube begins to warm up. This means that less energy will be required to meet the TO setpoint so the control is able to push a little more feed into the heater. It increases FC1 and FC2 in order to keep DT as close to zero as possible. At 85 min the other tube inlet temperature increases enough to allow the FO setpoint to be reached.

MPC Control Hierarchy

In a modern chemical plant or refinery, the MPC controller is part of a multilevel hierarchy, as illustrated in Fig. 7. Moving from the top level to the bottom, the control functions execute at a higher frequency but cover a smaller geographic scope. At bottom level, referred to as Level 0, basic proportional-integral-derivative (PID) controllers execute once a second within distributed control system (DCS) hardware. These controllers adjust individual valves to

maintain desired flows (FC), pressures (PC), levels (LC), and temperatures (TC).

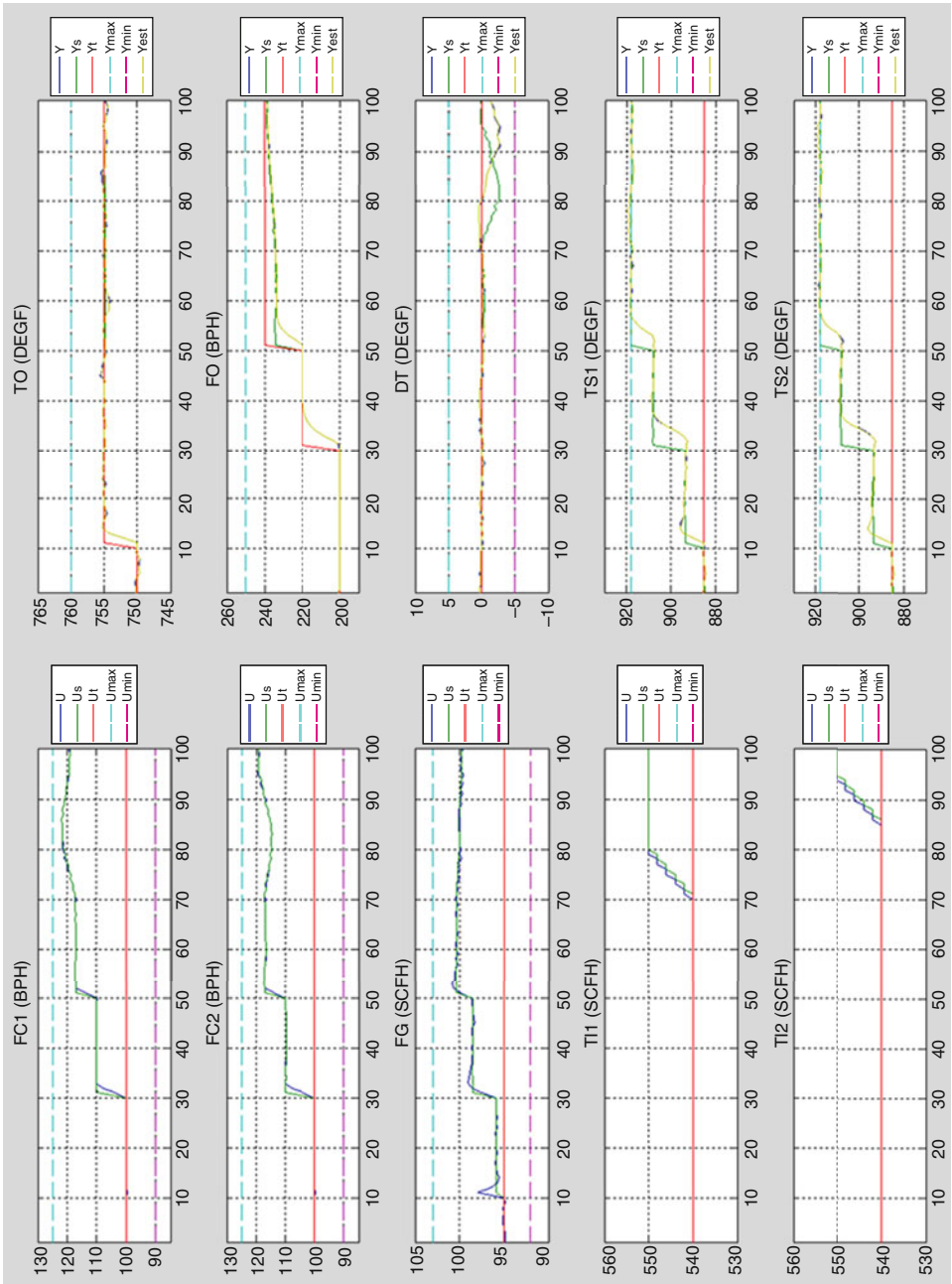
At Level 1 a MPC controller runs once a minute to perform dynamic constraint control for an individual processing unit, such as crude distillation unit or a fluid catalytic cracker (Gary et al. 2007). It has the job of holding the unit at the best economic operating point in the face of dynamic disturbances and operational constraints.

At Level 2 a real-time optimizer (RTO) runs hourly to calculate optimal steady-state targets for a collection of processing units. It uses a rigorous first-principles steady-state model to calculate targets for key operating variables such as unit temperatures and feed rates. These are typically passed down to several MPC controllers for implementation.

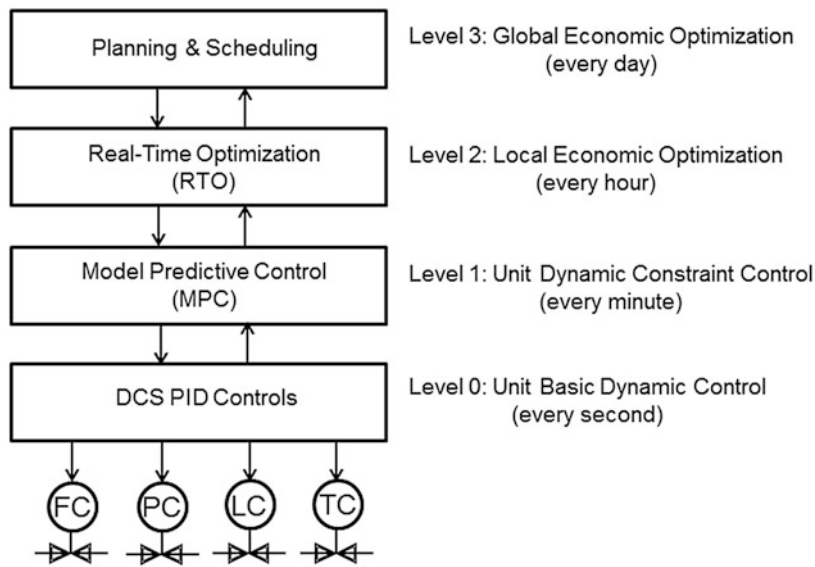
At Level 3 planning and scheduling functions are carried out daily to optimize economics for an entire chemical plant or refinery. Key operating targets and economic data are typically passed to several RTO applications for implementation.

MPC Algorithms

MPC algorithms function in much the same way that an experienced human operator would approach a control problem. Figure 8 illustrates the flow of information for a typical MPC

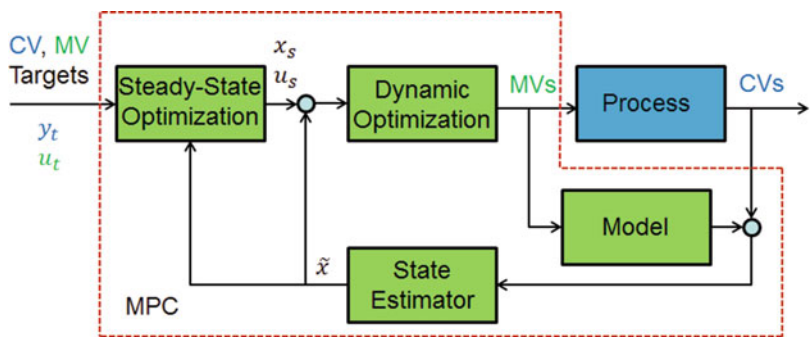


Model Predictive Control in Practice, Fig. 6 Fired heater MPC closed-loop test (U – process input, U_s – input steady-state target (computed by MPC), U_t – input target (entered by operator), $U_{\max}$ – input maximum limit, $U_{\min}$ – input minimum limit, Y – process output, Y_s – output steady-state target (computed by MPC), Y_t – output target (entered by operator), $Y_{\max}$ – output maximum limit, $Y_{\min}$ – output minimum limit)



Model Predictive Control in Practice, Fig. 7 Hierarchy of control functions in a refinery/chemical plant

Model Predictive Control in Practice, Fig. 8 Information flow for MPC algorithm



implementation. At each control interval, the algorithm compares the current model output prediction to the measured output and passes this information to a state estimator. The state estimate, which includes an estimate of the process disturbances, is then passed to a steady-state optimization to determine the best operating point for the unit. The steady-state state and input targets are then passed, along with the state estimate, to a dynamic optimization to compute the best trajectory of future MV adjustments. This trajectory of future MV adjustments is often referred to as a move plan. The first computed MV adjustment is then implemented, and the entire calculation sequence is repeated at the next

control interval. The various commercial MPC algorithms differ in such details as the mathematical form of the dynamic model and the specific formulations of the state estimation, steady-state optimization, and dynamic optimization problems (Qin and Badgwell 2003). A general formulation that illustrates the most important concepts is presented in the subsections below.

Model Types

A state-space model used in MPC can be written in the following form:

$$\begin{aligned}
x_{k+1} &= f(x_k, u_k) + w_k && \text{state dynamics} \\
y_k &= g(x_k, u_k) + v_k && \text{output response} \\
u &\in R^m && \text{inputs (MVs and DVs)} \\
y &\in R^l && \text{outputs (CVs)} \\
x &\in R^n && \text{states} \\
w &\sim N(0, Q_e) && \text{state disturbance} \\
v &\sim N(0, R_e) && \text{measurement noise}
\end{aligned}$$

This is a discrete model form in which k indexes the control intervals. The state transition function $f(x_k, u_k)$ and the measurement function $g(x_k, u_k)$ are typically developed through some combination of first-principles knowledge and step test data.

State Estimation

The state estimation problem can be posed as an optimization over a window of past process behavior, defined by past inputs u_k and past measured outputs $\hat{y}_k$:

$$\begin{aligned}
\min_{x_{k-N_e} \cdots x_k} \quad & \Phi = P_{k-N_e}(x_{k-N_e}) \\
& + \sum_{j=k-N_e}^{k-1} \left(w_j^T Q_e^{-1} w_j \right) \\
& + \sum_{j=k-N_e}^k \left(v_j^T R_e^{-1} v_j \right) \\
s.t. : \quad & x_{j+1} = f(x_j, u_j) + w_j; \\
& \hat{y}_j = g(x_j, u_j) + v_j; x_j \in X
\end{aligned}$$

This formulation, known as moving horizon estimation (MHE), is solved at each sample time and provides a very flexible way to enforce realistic constraints on the estimated state. The goal is to estimate the sequence of states that best optimizes the trade-off between process disturbances and measurement noise.

Steady-State Optimization

The steady-state optimization problem can be written as follows:

$$\begin{aligned}
\min_{x_s, u_s} \quad & \Phi = (y_t - y_s)^T Q_c (y_t - y_s) \\
& + (u_t - u_s)^T R_c (u_t - u_s) \\
s.t. : \quad & x_s = f(x_s, u_s); \\
& y_s = g(x_s, u_s); x_s \in X; u_s \in U
\end{aligned}$$

where u_t and y_t are desirable targets from upper-level optimization results. Here the goal is to drive the steady-state target as close as possible to these targets.

Dynamic Optimization

The dynamic optimization problem can be posed as an optimization over future process behavior:

$$\begin{aligned}
\min_{u_k \cdots u_{k+N_c-1}} \quad & \Phi \\
& = \sum_{j=0}^{N_c-1} \left[(y_s - y_{k+j})^T Q_c (y_s - y_{k+j}) \right. \\
& \quad + (u_s - u_{k+j})^T R_c (u_s - u_{k+j}) \\
& \quad \left. + \Delta u_{k+j}^T S_c \Delta u_{k+j} \right] + P_{k+N_c}(z_{k+N_c}) \\
s.t. : \quad & z_{j+1} = f(z_j, u_j); \\
& y_j = g(z_j, u_j); \\
& z_0 = x_k; z_j \in X; u_j \in U
\end{aligned}$$

An optimal sequence of future input adjustments is computed by trading off between three competing goals: driving the outputs to their steady-state targets, driving the inputs to their steady-state targets, and minimizing input movement.

Numerical Solution Methods

In the general case, the MPC algorithm must solve the three optimization problems outlined above at each control interval. For the case of linear models and reasonable tuning parameters, these problems take the form of a convex quadratic program (QP) with a constant, positive definite Hessian. As such they can be solved relatively easily using standard optimization codes. For the case of a linear state-space model, the structure can be exploited even further to develop a specialized solution algorithm using an interior point method (Rao et al. 1998).

For the case of nonlinear models, these problems take the form of a nonlinear program for which the solution domain is no longer convex. This greatly complicates the numerical solution. A typical strategy is to iterate on a linearized

version of the problem until convergence (Bielger 2010).

MPC Implementation

The combined experience of thousands of MPC applications in the process industries has led to a near consensus on the steps required for a successful implementation. These are justification, pretest, step test, modeling, control configuration, commissioning, postaudit, and sustainment. These steps must be carried out by a carefully selected project team that typically includes, in addition to the MPC expert, an engineer with detailed knowledge of the process and an operator with considerable operations experience. The following subsections describe these steps for a typical implementation of MPC with a linear model on a refinery or chemical process unit.

Justification

Application of MPC technology in industry can only be justified by the resulting economic benefits. Therefore the first step in an MPC application is to estimate the potential economic benefits and associated project costs. MPC technology makes money primarily by minimizing variations in key variables and then moving the operation closer to constraints. Consider the example of controlling the amount of impurity in a product, illustrated in Fig. 9. Under operator control the variation in the impurity level may be quite large, so that the average operating point must be moved far below the maximum specification limit in order to prevent limit violations. With a good multivariable control, the variation is reduced significantly, but there is no benefit until the average amount of impurity in the product is increased. This is what happens under MPC control. The average amount of impurity in the product increases significantly without violating the maximum specification limit. This in turn allows more profitable operation, either by allowing more feed to be pushed through the unit or by lowering the operating cost.

Project expenditures include engineering expenses, hardware and software costs, additional manpower required for step tests and commissioning, lost-opportunity costs associated with step testing, and charges for operator training, postaudit, and sustainment.

In addition to increased throughput and/or lower operating costs, there are often less tangible benefits that result from an application of MPC technology. These include such things as fixing broken instruments and valves, de-bottlenecking the limiting constraints, and increased reliability from more consistent operation. Representative estimates of the net benefits are typically in the range of \$100,000 to \$1,000,000 per year, with payback times for the entire project in the range of 3 months to 1 year.

Pretest

The goal of the pretest is to evaluate overall unit operation and reach a consensus on a preliminary control design. This includes testing each of the MVs to see if there are tuning or valve problems and monitoring the quality and timing of the CV measurements. It is quite common to find MV controllers (flow, temperature, pressure, and level) that must be returned or valves that must be replaced prior to control commissioning. CV measurements may be too noisy or may be available only intermittently. Disturbances to the unit must be identified along with corresponding feedforward measurements (DV's), if any. Control hardware or software issues must be identified so that they can be addressed prior to control commissioning. Control design discussions must include prioritization of the CV specifications. As in the fired-heater example discussed earlier, the CVs can typically be grouped into three categories: safety limits, product specifications, and economic optimization.

Step Test

The goal of the step test is to produce a high-quality data set for use in model development. In the best case, these data will include the full range of typical operation and will show clearly how adjustments in each of the MVs and DVs affect each of the CVs. The tests must be conducted

munication between the MPC control computer and the process hardware. Initially conservative tuning is then entered, and the MPC components are tested using a combination of process data and closed-loop simulation. For the simulations, the model is typically assumed to be perfect, and each of the key process specifications is activated by changing limits. The timing and magnitude of the resulting MV and CV responses can then be adjusted changing the tuning factors.

Commissioning

The goal of commissioning is to verify correct control behavior for the most important control scenarios and to finalize control tuning for long-term operation. The state estimator is turned on first to verify reasonable behavior for the CV predictions. Estimator tuning may be adjusted manually or calculated from process operating data (Odelson et al. 2006). MV connections are then verified, one at a time, by activating the MVs and adjusting the limits in a region very close to current operation. MV limits are then widened incrementally while testing each of the key CV specifications. CV response testing usually leads to significant changes in control tuning.

The duration of commissioning is dictated by the closed-loop response time of CVs. Typically several tests of each CV specification are required to verify correct closed-loop behavior. Operator training is often conducted during this period.

Postaudit

After operating for a significant amount of time, typically several months, a postaudit is conducted to verify estimated benefits and to catalogue areas for improving the MPC controller. This activity usually results in a ranked list of additional CVs to be included in the controller.

Sustainment

Inevitable process and disturbance changes necessitate an aggressive program of daily analysis and maintenance to sustain the economic benefits of an MPC controller. This often includes analysis of carefully defined metrics for the overall controller and for each MV and CV. Maintenance activities include controller

returning, model adjustments, and addition of new MVs and CVs.

Summary and Future Directions

Model predictive control is now a mature technology in the process industries. A representative MPC algorithm in this domain includes a state estimator, a steady-state optimization, and a dynamic optimization, running once a minute. A successful MPC application usually starts with a careful economic justification, includes significant participation from process engineers and operators, and is maintained with an aggressive sustainment program. Many thousands of such applications are currently operating around the world, generating billions of dollars per year in economic benefits.

Likely future directions for MPC practice include increasing use of nonlinear models, improved state estimation through unmeasured disturbance modeling (Pannocchia and Rawlings 2003), and development of more efficient numerical solution methods (Zavala and Biegler 2009).

Cross-References

- [Industrial MPC of Continuous Processes](#)
- [Linear Quadratic Optimal Control](#)
- [Nominal Model-Predictive Control](#)

Bibliography

- Badgwell TA (2010) Estimating the state of model predictive control. In: Presented at the AIChE annual meeting, Salt Lake City, 7–10 Nov 2010
- Biegler LT (2010) Nonlinear programming, concepts, algorithms, and applications to chemical processes. SIAM, Philadelphia
- Cutler CR, Ramaker BL (1979) Dynamic matrix control – a computer control algorithm. In: Paper presented at the AIChE national meeting, Houston, Apr 1979
- Darby ML, Nikolaou M (2012) MPC: current practice and challenges. *Control Eng Pract* 20:328–342
- Gary JH, Handwerk, GE, Kaiser, MJ (2007) Petroleum refining: technology and economics. CRC Press, New York
- Ljung L (1999) System identification: theory for the user. Prentice Hall, Upper Saddle River

- Maciejowski JM (2002) Predictive control with constraints. Pearson Education Limited, Essex
- Mayne DQ, Rawlings JB, Rao CV, Sokaert POM (2000) Constrained model predictive control: stability and optimality. *Automatica* 36:789–814
- Odelson BJ, Rajamani MR, Rawlings JB (2006) A new autocovariance least-squares method for estimating noise covariances. *Automatica* 42:303–308
- Pannocchia G, Rawlings JB (2003) Disturbance models for offset-free model predictive control. *AIChE J* 49:426–437
- Rawlings JB, Mayne DQ, Diehl M (2017) Model predictive control: theory, computation, and design, 2nd edn. Nob Hill Publishing, Madison
- Qin SJ, Badgwell TA (2003) A survey of industrial model predictive control technology. *Control Eng Practice* 11:733–764
- Rao CV, Wright SJ, Rawlings JB (1998) Application of interior-point methods to model predictive control. *J Optim Theory Appl* 99:723–757
- Richalet J, Rault A, Testud JL, Papon J (1978) Model predictive heuristic control: applications to industrial processes. *Automatica* 14:413–428
- Zavala VM, Biegler LT (2009) The advanced-step NMPC controller: optimality, stability, and robustness. *Automatica* 45: 86–93

Model Reference Adaptive Control

Jing Sun

University of Michigan, Ann Arbor, MI, USA

Abstract

The fundamentals and design principles of model reference adaptive control (MRAC) are described. The controller structure and adaptive algorithms are delineated. Stability and convergence properties are summarized.

Keywords

Certainty equivalence · Lyapunov-SPR design · MIT rule

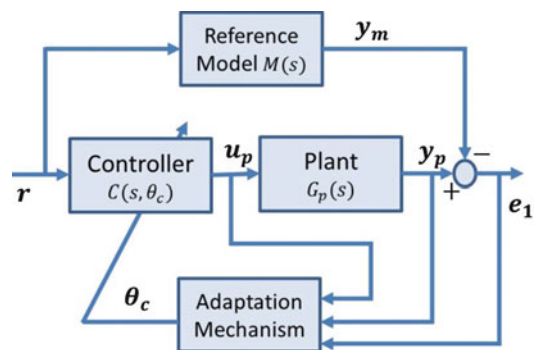
Synonyms

MRAC

Introduction

Model reference adaptive control (MRAC) is an important adaptive control approach, supported by rigorous mathematical analysis and effective design toolsets. It is made up of a feedback control law that contains a controller $C(s, \theta_c)$ and an adjustment mechanism that generates the controller parameter updates $\theta_c(t)$ online. While different MRAC configurations can be found in the literature, the structure shown in Fig. 1 is commonly used and includes all the basic components of an MRAC system. The prominent features of MRAC are that it incorporates a reference model which represents the desired input–output behavior and that the controller and adaptation law are designed to force the response of the plant, y_p , to track that of the reference model, y_m , for any given reference input r .

Different approaches have been used to design MRAC, and each may lead to a different implementation scheme. The implementation schemes fall into two categories: direct and indirect MRAC. The former updates the controller parameters θ_c directly using an adaptive law, while the latter updates the plant parameters θ_p first using an estimation algorithm and then updates θ_c by solving, at each time t , certain algebraic equations that relate θ_c with the online estimates of θ_p . In both direct and indirect MRAC schemes, the controller structure is kept the same



Model Reference Adaptive Control, Fig. 1 Schematic of MRAC

as that which would be used in the case that the plant parameters are known.

MRC Controller Structure

Consider the design objective of model reference control (MRC) for linear time-invariant systems: Given a reference model $M(s)$, find a control law such that the closed-loop system is stable and $y_p \rightarrow y_m$ as $t \rightarrow \infty$ for any bounded reference signal r .

For the case of known plant parameters, the MRC objective can be achieved by designing the controller so that the closed-loop system has a transfer function equal to $M(s)$. This is the so-called model matching condition. To assure the existence of a causal controller that meets the model matching condition and guarantees internal stability of the closed-loop system, the following assumptions are essential:

- A1. The plant has a stable inverse, and the reference model is chosen to be stable.
- A2. The relative degree of $M(s)$ is equal to or greater than that of the plant $G_p(s)$. Herein, the relative degree of a transfer function refers to the difference between the orders of the denominator and numerator polynomials.

It should be noted that these assumptions are imposed to the MRC problem so that there is enough structural flexibility in the plant and in the reference model to meet the control objectives. A1 is necessary for maintaining internal stability of the system while meeting the model matching condition, and A2 is needed to ensure the causality of the controller. Both assumptions are essential for non-adaptive applications when the plant parameters are known, let alone for the adaptive cases when the plant parameters are unknown.

The reference model plays an important role in MRAC, as it will define the feasibility of MRAC design as well as the performance of

the resulting closed-loop MRAC system. The reference model should reflect the desired closed-loop performance. Namely, any time-domain or frequency-domain specifications, such as time constant, damping ratio, natural frequency, bandwidth, etc., should be properly reflected in the chosen transfer function $M(s)$.

The controller structure for the MRAC is derived with these assumptions for the known plant case and extended to the adaptive case by combining it with a proper adaptive law. Under assumptions A1–A2, there exist infinitely many control solutions C that will achieve the MRC design objective for a given plant transfer function $G_p(s)$. Nonetheless, only those extendable to MRAC with the simplest structure are of interest. It is known that if a special controller structure with the following parametrization is imposed, then the solution to the model matching condition, in terms of the ideal parameter θ_c^* , will be unique:

$$u_p = \theta_1^{*T} \omega_1 + \theta_2^{*T} \omega_2 + \theta_3^{*T} y_p + c_0^* r = \theta_c^{*T} \omega$$

where $\theta_c^* \in R^{2n}$, n is the order of the plant,

$$\theta_c^* = \begin{bmatrix} \theta_1^* \\ \theta_2^* \\ \theta_3^* \\ c_0^* \end{bmatrix}, \omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ y_p \\ r \end{bmatrix}$$

and $\omega_1, \omega_2 \in R^{n-1}$ are signals internal to the controller generated by stable filters (Ioannou and Sun 1996).

This MRC control structure is particularly appealing for adaptive control development, as the parameters appear linearly in the control law expression, leading to a convenient linear parametric model for adaptive algorithm development.

Adaptation Algorithm

Design of adaptive algorithms for parameter updating can be pursued in several different approaches, thereby resulting in different MRAC schemes. Three direct design approaches,

namely, the Lyapunov-SPR, the certainty equivalence, and the MIT rule, will be briefly described together with indirect MRAC.

Lyapunov-SPR Design

One popular MRAC algorithm is derived using Lyapunov's direct method and the Meyer-Kalman-Yakubovich (MKY) Lemma based on the strictly positive real (SPR) argument. The concept of SPR transfer functions originates from network theory and is related to the driving point impedance of dissipative networks. The MKY Lemma states that given a stable transfer function $M(s)$ and its realization (A, B, C, d) where $d \geq 0$ and all eigenvalues of the matrix A are in the open left half plane: If $M(s)$ is SPR, then for any given positive definite matrix $L = L^T > 0$, there exists a scalar $\nu > 0$, a vector q , and a $P = P^T > 0$ such that

$$\begin{aligned} A^T P + P A &= -q q^T - \nu L \\ P B - C &= \pm q \sqrt{2d} \end{aligned}$$

By choosing $M(s)$ to be SPR, one can formulate a Lyapunov function consisting of the state tracking and parameter estimation errors and use the MKY Lemma to define the adaptive law that will force the derivative of the Lyapunov function to be semi-negative definite. The resulting adaptive law has the following simple form:

$$\dot{\theta} = -\Gamma e_1 \omega \text{sign}(c_0^*)$$

where $e_1 = y_p - y_m$ is simply the tracking error and $c_0^* = k_m/k_p$ with k_m, k_p being the high frequency gain of the transfer function for the reference model $M(s)$ and the plant $G_p(s)$, respectively. This algorithm, however, applies only to systems with relative degree equal to 0 or 1, which is implied by the SPR condition imposed on $M(s)$ and assumption A2.

The Lyapunov-SPR-based MRAC design is mathematically elegant in its stability analysis but is restricted to a special class of systems. While it can be extended to more general cases with relative degrees equal to 2 and 3, the resulting

control law and adaptive algorithm become much more complicated and cumbersome as efforts must be made to augment the control signal in such a way that the MKY Lemma is applicable to the "reformulated" reference model.

Certainty Equivalence Design

For more general cases with a high relative degree, another design approach based on "certainty equivalence" (CE) principle is preferred, due to the simplicity in its design as well as its robustness properties in the presence of modeling errors. This approach treats the design of the adaptive law as a parameter estimation problem, with the estimated parameters being the controller parameter vector θ_c^* . Using the specific linear formulation of the control law and assuming that θ_c^* satisfies the model matching condition, one can show that the ideal controller parameter satisfies the following parametric equation:

$$z = \theta_c^{*T} \omega_p$$

with

$$z = M(s)u_p, \omega_p = \begin{bmatrix} M(s)\omega_1 \\ M(s)\omega_2 \\ M(s)y_p \\ y_p \end{bmatrix}$$

This parametric model allows one to derive adaptive laws to estimate the unknown controller parameter θ_c^* using standard parameter identification techniques, such as the gradient and least squares algorithms. The corresponding MRAC is then implemented in the CE sense where the unknown parameters are replaced by their estimated value. It should be noted that a CE design does not guarantee closed-loop stability of the resulting adaptive system, and additional analysis has been carried out to establish closed-loop stability.

MIT Rule

Besides the Lyapunov-SPR and CE approaches mentioned earlier, the direct MRAC problem can

also be approached using the so-called MIT rule, an early form of MRAC developed in the 1950s–1960s in the Instrumentation Laboratory at MIT for flight control. The designer defines a cost function, e.g., a quadratic function of tracking error, and then adjusts parameters in the direction of steepest descent. The negative gradient of the cost function is usually calculated through the sensitivity derivative approach. The formulation is quite flexible, as different forms of MIT rule can be derived by changing the cost function following the same procedure and reusing the same sensitivity functions. Despite its effectiveness in some practical applications, MRAC systems designed with MIT rule have had stability and robustness issues.

Indirect MRAC

While most of the MRAC systems are designed as direct adaptive systems, indirect MRAC systems can also be developed which explicitly estimate the plant parameter θ_p as an intermediate step. The adaptive law for an indirect MRAC includes two basic components: one for estimating the plant parameters and another for calculating the controller parameters based on the estimated plant parameters. This approach would be preferred if the plant transfer function is partially known, in which case the identification of the remaining unknown parameters represents a less complex problem. For example, if the plant has no zeros, the indirect scheme estimates $n + 1$ parameters, while the direct scheme has to estimate $2n$ parameters.

Indirect MRAC is a CE-based design. As such, the design is intuitive but the design process does not guarantee closed-loop stability, and separate analysis has to be carried out to establish stability. Except for systems with a low number of zeros, the “feasibility” problem could also complicate the matter, in the sense that the MRC problem may not have a solution for the estimated plant at some time instants even though the solution exists for the real plant. This problem is unique to the indirect design, and several mitigating

solutions have been found at the expense of more complicated adaptation or control algorithms.

Stability, Robustness, and Parameter Convergence

Stability for MRAC often refers to the properties that all signals are bounded and tracking error converges to zero asymptotically. Robustness for adaptive systems implies that signal boundedness and tracking error convergence (to a small residue set) will be preserved in the presence of small perturbations such as disturbances, un-modeled dynamics, and time-varying parameters. For different MRAC schemes, different approaches are used to establish their properties.

For the Lyapunov-SPR-based MRAC systems, stability is established in the design process where the adaptive law is derived to enforce a Lyapunov stability condition. For CE-based designs, establishing stability for the closed-loop MRAC system is a nontrivial exercise for both direct and indirect schemes. Using properly normalized adaptive laws for parameter estimation, however, stability can be proved for direct and indirect MRAC schemes. For MRAC systems designed with the MIT rule, local stability can be established under more restrictive conditions, such as when the parameters are close to the ideal ones.

It should be noted that the adaptive control algorithm in the original form has been shown to have robustness issues, and extensive publications in the 1980s and 1990s were devoted to robust adaptive control in attempts to mitigate the problem. Many modifications have been proposed and shown to be effective in “robustifying” the MRAC; interested readers are referred to the article on ► [Robust Adaptive Control](#) for more details.

Parameter convergence is not an intrinsic requirement for MRAC, as tracking error convergence can be achieved without parameter convergence. It has been shown, however, that parameter convergence could enhance robustness, particularly for indirect schemes. As in the case for parameter identification, a persistent excitation (PE) condition needs to

be imposed on the regression signal to assure parameter convergence in MRAC. In general, PE is accomplished by properly choosing the reference input r . It can be established for most MRAC approaches that parameter convergence is achieved if, in addition to conditions required for stability, the reference input r is sufficiently rich of order $2n$, $\dot{r}$ is bounded, and there is no pole-zero cancelation in the plant transfer function. A signal is called to be sufficiently rich of order m if it contains at least $m/2$ distinct frequencies.

Summary and Future Directions

MRAC incorporates a reference model to capture the desired closed-loop responses and designs the control law and adaptation algorithm to force the output of the plant to follow the output of the reference model. Several different design approaches are available. Stability, robustness, and parameter convergence have been established for different MRAC designs with appropriate assumptions.

MRAC had been a very active and fruitful research topic from the 1960s to 1990s, and it formed important foundations for modern adaptive control theory. It also found many successful applications ranging from chemical process controls to automobile engine controls. More recent efforts have been mostly devoted to integrating it with other design approaches to treat nonstandard MRAC problems for nonlinear and complex dynamic systems.

Cross-References

- [Adaptive Control of Linear Time-Invariant Systems](#)
- [Adaptive Control: Overview](#)
- [History of Adaptive Control](#)
- [Robust Adaptive Control](#)

Recommended Reading

MRAC has been well covered in several textbooks and research monographs. Astrom

and Wittenmark (1994) presented different MRAC schemes in a tutorial fashion. Narendra and Annaswamy (1989) focused on stability of deterministic MRAC systems. Ioannou and Sun (1996) covered the detailed derivation and analysis of different MRAC schemes and provided a unified treatment for their stability and robustness analysis. MRAC systems for discrete-time (Goodwin and Sin 1984) and for nonlinear (Krstic et al. 1995) processes are also well explored.

Bibliography

- Astrom KJ, Wittenmark B (1994) Adaptive control. Second edition. Prentice Hall, Englewood Cliffs
- Goodwin GC, Sin KS (1984) Adaptive filtering, prediction and control. Prentice Hall, Englewood Cliffs
- Ioannou PA, Sun J (1996) Robust adaptive control. Prentice Hall, Upper Saddle River
- Krstic K, Kanellakopoulos I, Kokotovic PV (1995) Nonlinear and adaptive control design. Wiley, New York
- Narendra KS, Annaswamy AM (1989) Stable adaptive systems. Prentice Hall, Englewood Cliffs

Model-Based Performance Optimizing Control

Sebastian Engell

Fakultät Bio- und Chemieingenieurwesen,
Technische Universität Dortmund, Dortmund,
Germany

Abstract

In many applications, e.g., in chemical process control, the purpose of control is to achieve an optimal performance of the controlled system despite the presence of significant uncertainties about its behavior and of external disturbances. Tracking of set points is often required for lower-level control loops, but at the system level in most cases, this is not the primary concern and may even be counterproductive. In this article,

the use of dynamic online optimization on a moving finite horizon to realize optimal system performance is discussed. By real-time optimization, a performance-oriented or economic cost criterion is minimized or maximized over a finite horizon, while the usual control specifications enter as constraints but not as set points. This approach integrates the computation of optimal set-point trajectories and of the regulation to these trajectories.

Keywords

MPC · Performance optimizing control ·
Process control · Integrated RTO and control

Introduction

From an applications point of view, the purpose of automatic feedback control (and that of manual control as well) in many cases is *not* primarily to keep the controlled variables at their set points as well as possible, or to track dynamic set-point changes, but to operate the controlled system such that its *performance* is optimized in the presence of disturbances and uncertainties, by exploiting the information gained in real time from the available measurements. This holds generally for the higher control layers in the process industries but similarly for many other applications. Suppose that for example, the availability of cooling water at a lower temperature than assumed as a worst case during plant design enables plant operation at a higher throughput. In this case what sense does it make to enforce the nominal operating point by tight feedback control? For a combustion engine, the goal is to achieve the desired torque with minimum consumption of fuel. For a cooling system, the goal is to keep the temperature of the goods or of a room within certain bounds with minimum consumption of energy, possibly weighted against the wear of the equipment. To regulate some variables to their set points may help to achieve these goals, but it is not the real performance target for the overall system. Feedback control loops

therefore usually are part of control hierarchies that establish good performance of the overall system and the meeting of constraints on its operation.

There are four main approaches to the integration of feedback control with system performance optimization:

- Choice of regulated variables such that, implicitly via the regulation of these variables to their set points, the performance of the overall system is close to optimal (see the entry on Control Structure Selection by Skogestad);
 - Tracking of necessary conditions of optimality where variables which determine the optimal operating policy are kept at or close to their constraints. This is a widespread approach especially in chemical batch processes where, e.g., the feeding of reactants is such that the maximum cooling power available is used (Finkler et al. 2014); see also the entry on Optimizing Control of Batch Processes by Bonvin.
- In these two approaches, the choice of the optimal set points or constraints to be tracked is done offline, and they are then implemented by the feedback layer of the process control hierarchy (see the entry on Control Hierarchy of Large Processing Plants by de Prada).
- Combination of a regulatory (tracking) feedback control layer with an optimization of the set points or system trajectories (called real-time optimization in the process industries) (see the entry on Real-time Optimization by Trierweiler);
 - Reformulation of model-predictive control such that the control target is not the tracking of references but the optimization of the system performance over a finite horizon, taking constraints of system variables or inputs into account directly within the online optimization. Here, the optimization is performed with a dynamic model, in contrast to performing steady-state optimization in real-time optimization or in the choice of self-optimizing control structures.

All four approaches are currently applied in the process industries. Tracking of necessary conditions of optimality is usually designed based on process insight rather than on a rigorous analysis, and the same holds for the selection of regulatory control structures. The combination of RTO with a regulatory control layer (often model-predictive control, MPC) is applied in particular to petrochemical plants. The last approach, performance optimizing control, has found applications so far mostly in semi-batch polymerization processes where the batch time is minimized under constraints on the reactor temperature (e.g., see Finkler et al. 2014). It is challenging in terms of the required models and algorithms and computing power, but it also has the highest potential in terms of the resulting performance of the controlled system. Moreover, it is structurally simple and easier to tune because the natural performance specification does not have to be translated into controller tunings, weights in MPC cost functions, etc. Therefore the idea of direct model-based performance optimizing control has found much attention in process control in recent years.

The four approaches above are discussed in more detail below. We also provide some historical notes and outline some areas of continuing research.

Performance Optimization by Regulation to Fixed Set Points

Morari et al. (1980) stated that the objective in the synthesis of a control structure is “to translate the economic objectives into process control objectives.” A subgoal in this “translation” is to select the regulatory control structure of a process such that steady-state optimality of process operations is realized to the maximum extent possible by driving the selected controlled variables to suitably chosen set points. A control structure with this property was termed “self-optimizing control” by Skogestad (2000). It should adjust the manipulated variables by keeping a function of the measured variables constant such that the process is operated at the economically optimal steady state in the presence of disturbances.

Ideally, in the steady state, a similar performance is obtained as it would be realized by optimizing the stationary values of the operational degrees of freedom of the system for known disturbances d and a perfect model. By regulating the controlled variables to their set points at the steady state in the presence of disturbances, a mapping $u = f(y_{\text{set}}, d)$ is implicitly realized which should be an approximation of the performance optimizing inputs $u_{\text{opt}}(d)$. The choice of the self-optimizing control structure takes only the steady-state performance into account, not the dynamic reaction of the controlled plant. An extension of the approach to include also the dynamic behavior can be found in Pham and Engell (2011).

Tracking of Necessary Conditions of Optimality

Very often, the optimal operation of a system in a certain phase of its evolution or under certain conditions is defined by some variables being at their constraints. If these variables are known and the conditions can be monitored, a switching control structure can be built that keeps the (possibly changing) set of critical variables at their constraints despite inaccuracies of the model, external disturbances, etc. In fact it turns out that such control schemes can, in the case of varying parameters and in the presence of disturbances, perform as good as sophisticated model-based optimization schemes (e.g., see Finkler et al. 2013).

Performance Optimization by Steady-State Optimization and Regulation

A well-established approach to create a link between regulatory control and the optimization of the performance of a system is to compute the set points of the controllers by an optimization layer. In process operations, this layer is called real-time optimization (RTO) (e.g., see Marlin and Hrymak (1997) and the references therein). An RTO system is a model-based, upper-

level control system that is operated in closed loop and provides set points to the lower-level control systems in order to maintain the process operation as close as possible to the economic optimum. It usually comprises an estimation of the plant states and plant parameters from the measured data and an economic or otherwise performance-related optimization of the operating point using a detailed nonlinear steady-state model.

As the RTO system employs a stationary process model and the optimization is only performed if the plant is approximately in a steady state, the time between successive RTO steps must be large enough for the plant to reach a new steady state after the last commanded move.

This structure is based upon a separation of concerns and of timescales between the RTO system and the process control system. The RTO system optimizes the system economics on a medium timescale (shifts to days), while the control system provides tracking and disturbance rejection on shorter timescales from seconds to hours.

As a an approximation to real-time optimization with a nonlinear rigorous plant model, in many MPC implementations nowadays, an optimization of the steady-state values based on the linear model that is used in the MPC controller is implemented. Then the gain matrix of the model must be estimated carefully to obtain good results.

Performance Optimizing Control

Model-predictive control has become the standard solution for demanding control problems in the process industries (Qin and Badgwell 2003) and increasingly is used also in other domains. The core idea is to employ a model to predict the effect of the future manipulated variables on the future controlled variables over a finite horizon and to use optimization to determine sequences of inputs which minimize a cost function over the so-called prediction horizon. In the unconstrained case with linear plant model and a quadratic cost function, the optimal control moves can

be computed by a closed-form solution. When constraints on inputs, outputs, and possibly also state variables are present, for a quadratic cost function and linear plant model, the optimization problem becomes a quadratic program (QP) that has to be solved in real time.

When the system dynamics are nonlinear and linear models are not sufficiently accurate over the operating region of interest, as is the case in many chemical processes, nonlinear model-predictive control which is based on nonlinear models of the process dynamics provides superior performance and therefore has met increasing interest both in theory and in practice. The classical formulation of nonlinear model-predictive tracking control (TC) is:

$$\min_u \varphi_{TC}(\hat{y}, u)$$

$$\varphi_{TC} = \sum_{n=1}^N \left(\sum_{i=1}^P \gamma_{n,i} (y_{n,ref}(k-i) - \hat{y}_n(k+i))^2 \right) + \sum_{l=1}^R \left(\sum_{j=1}^M \alpha_{l,j} \Delta u_l^2(k+j) \right)$$

s.t.

$$x(i+1) = f(x(i), z(i), u(i), i), \quad i = k, \dots, k+P$$

$$0 = g(x(i), z(i), u(i), i), \quad i = k, \dots, k+P$$

$$y(i+1) = h(x(i+1), u(i)), \quad i = k, \dots, k+P$$

$$x_{\min} \leq x(i) \leq x_{\max}, \quad i = k, \dots, k+P$$

$$y_{\min} \leq \hat{y}(i) \leq y_{\max}, \quad i = k, \dots, k+P$$

$$u_{\min} \leq u(i) \leq u_{\max}, \quad i = k, \dots, k+M$$

$$-\Delta u_{\min} \leq \Delta u(i) \leq \Delta u_{\max}, \quad i = k, \dots, k+M$$

$$u(i) = u(i-1) + \Delta u(i), \quad i = k, \dots, k+M$$

$$u(i) = u(k+M), \quad \forall i > k+M$$

Here f and g represent the plant model in the form of a system of differential-algebraic equations and h is the output function. P is the length of the prediction horizon, and M is the length of the control horizon; $y_1, \dots, y_N$ are the predicted control outputs; $u_1, \dots, u_R$ are the

control inputs. α and γ represent the weights on the control inputs and the control outputs, respectively. y_{ref} refers to the set point or the desired output trajectory; $\hat{y}(i)$ are the corrected model predictions. N is number of the controlled outputs; R is the number of the control inputs. Compensation for plant model mismatch and unmeasured disturbances is usually done using the bias correction equations:

$$d(k) = y^{\text{meas}}(k) - y(k),$$

$$\hat{y}(k+i) = y(k+i) + d(k), \quad i = k, \dots, k+P$$

The idea of direct performance optimizing control (POC) is to replace this formulation by a performance-related objective function:

$$\begin{aligned} \min_u \varphi_{POC}(y, u) \\ \varphi_{POC} = \sum_{l=1}^R \left(\sum_{j=1}^M \alpha_{l,j} \Delta u_l^2(k+j) \right) \\ - \left(\sum_{i=1}^P \beta_i \psi(k+i) \right). \end{aligned}$$

Here $\psi(k+i)$ represents the value of the performance cost criterion at the time step $[k+i]$. The optimization of the future control moves is subject to the same constraints as before. In addition, instead of reference tracking, constraints are formulated for all outputs that are critical for the operation of the system or its performance, e.g., product quality specifications or limitations of the equipment. In contrast to reference tracking, these constraints usually are one-sided (inequalities) or define operation bands. By this formulation, e.g., the production revenues can be maximized online over a finite horizon, considering constraints on product purities and waste streams. Feedback enters into the computation by the initialization of the model with a new initial state that is estimated from the available measurements of system variables and by the bias correction. Thus direct performance optimizing control realizes an online optimization of all operational degrees of freedom in a feedback

structure without tracking of a priori fixed set points or reference trajectories. The regularization term that penalizes control moves is added to the purely economic objective function to obtain smoother solutions.

This approach has several advantages over a combined steady-state optimization/ linear MPC scheme:

- Immediate reaction to disturbances, no waiting for the plant to reach a steady state is required.
- “Overregulation” is avoided – no variables are forced to fixed set points and all degrees of freedom can be used to improve the (economic) performance of the plant.
- Performance goals and process constraints do not have to be mapped to a control cost that defines a compromise between different goals. In this way, the formulation of the optimization problem and the tuning are facilitated compared to achieving good performance by tuning of the weights of a tracking formulation.
- More constraints than available manipulated variables can be handled as well as more manipulated variables than variables that have to be regulated.
- No inconsistency arises from the use of different models on different layers.
- The overall scheme is structurally simple.

Similar to any NMPC controller that is designed for reference tracking, a successful implementation requires careful engineering such that as many uncertainties as possible are compensated by simple feedback controllers and only the key dynamic variables are handled by the optimizing controller based on a rigorous model of the essential dynamics and of the stationary relations of the plant.

History and Examples

The idea of economic or performance optimizing control originated from the process control

community. The first papers on directly integrating economic considerations into model-predictive control (Zanin et al. 2000) proposed to achieve a better economic performance by adding an economic term to a classical tracking performance criterion and applied this to the control of a fluidized bed catalytic cracker. Helbig et al. (2000) discussed different ways to integrate optimization and feedback control including direct dynamic optimization for the example of a semi-batch reactor. Toumi and Engell (2004) and Erdem et al. (2004) demonstrated that online performance optimizing control schemes for simulated moving bed (SMB) chromatographic separations in lab-scale SMB processes are periodic processes and constitute prototypical examples where additional degrees of freedom can be used to simultaneously optimize *system* performance and to meet product specifications. Bartusiak (2005) already reported industrial applications of carefully engineered performance optimizing NMPC controllers.

Direct performance optimizing control was suggested as a promising general new control paradigm for the process industries by Rolandi (2005), Engell (2006, 2007), Rawlings and Amrit (2009), and others. Meanwhile it has been demonstrated in many simulation studies that direct optimization of a performance criterion can lead to superior economic performance compared to classical tracking (N)MPC (e.g., Ochoa et al. (2010) for a bioethanol process, Idris and Engell (2012) for a reactive distillation column).

An application study for the drying of milk powder can be found in Petersen et al. (2017). Haßkerl et al. (2018) demonstrated performance-optimizing control for a real pilot-scale reactive distillation process that performs a complex trans-esterification process based upon a rigorous process model consisting of 186 differential states and 288 algebraic equations. It was shown that the performance optimizing controller can drive the process to the optimum quickly and can perform a transition between two different main products automatically and with minimized losses.

Further Issues

Modeling and Robustness

In a direct performance optimizing control approach, sufficiently accurate dynamic nonlinear process models are needed. While in the process industries nonlinear steady-state models are nowadays available for many processes because they are built and used extensively in the process design phase, there is still a considerable additional effort required to formulate, implement, and validate nonlinear dynamic process models. The effort for rigorous or semi-rigorous modeling usually dominates the cost of an advanced control project. To use data-based black box or gray box models may be effective for regulatory control where the model only has to capture the essential dynamics of the plant near an operating point, but it seems to be less suitable for optimizing control where the optimal plant performance is aimed at, and hence the operating point varies dynamically.

Model inaccuracies always have to be taken into account. They may not only lead to suboptimal performance but also can cause that the constraints even on measured variables cannot be met in the future because of an insufficient back-off from the constraints. A new approach to deal with uncertainties about model parameters and future influences on the process is multistage MPC which is based on optimization with recourse. Here the model uncertainties are represented by a set of scenarios of parameter variations, and the future availability of additional information is taken into account. It has been demonstrated that this is an effective tool to handle model uncertainties and to automatically generate the necessary back-off without being overly conservative Lucia et al. (2013). An application to a complex biotechnological process with significant model uncertainty can be found in (Hebing et al. 2020). Here two structurally different models, based on classical Monod-type kinetics, the other on a gray box model, are employed.

State Estimation

For the computation of economically optimal process trajectories based upon a rigorous non-

linear process model, the state variables of the system at the beginning of the prediction horizon must be known. As not all states will be measured in a practical application, state estimation is a key ingredient of a performance optimizing controller. Extended Kalman filters are the standard solution used in the process industries. If the nonlinearities are significant, unscented Kalman filters or particle filters may be used. A novel approach is to formulate the state estimation problem also as an optimization problem on a moving horizon (Rao et al. 2003). The estimation of some important varying unknown model parameters can be included in this formulation. As accurate state estimation is at least as critical for the performance of the closed-loop system as the exact tuning of the optimizing controller, attention must be paid to the investigation of the performance of state estimation schemes in realistic situations with model-plant mismatch.

Stability

Optimization of a cost function over a finite horizon in general assures neither optimality of the complete trajectory beyond this horizon nor stability of the closed-loop system. Closed-loop stability has been addressed extensively in the theoretical research in nonlinear model-predictive control. Stability can be assured by a proper choice of the stage cost within the prediction horizon and the addition of a cost on the terminal state and the restriction of the terminal state to a suitable set. In performance optimizing MPC, there is no a priori known steady state to which the trajectory should converge, and the economic cost function may not satisfy the usual conditions for closed-loop stability, e.g., because it only involves some of the inputs and not necessarily the states of the system under control. In recent years, important results on closed-loop stability guaranteeing formulations have been obtained, involving terminal constraints or a quasi-infinite horizon (Diehl et al. 2011; Angeli et al. 2012; Grüne 2013).

Reliability and Transparency

Nowadays quite large nonlinear dynamic optimization problems can be solved in real time,

not only for slow processes as they are found in the chemical industry but also in mechatronics and automotive control. So this issue does no longer prohibit the application of a performance optimizing control scheme to complex systems. A practically very important limiting issue however is that of reliability and transparency. It is difficult to guarantee that a nonlinear optimizer will always provide a solution which at least satisfies the constraints and gives a reasonable performance for all possible input data. While for an RTO scheme an inspection of the commanded set points by the operators usually will be feasible, this is less likely to be realistic in a dynamic situation. Hence automatic result filters are necessary as well as a backup scheme that stabilizes the process in the case where the result of the optimization is not considered safe. In the process industries, the operators will continue to supervise the operation of the plant in the foreseeable future, so a control scheme that includes performance optimizing control must be structured into modules, the outputs of which can still be understood by the operators so that they build up trust in the optimization. Good operator interfaces that display the predicted moves and the predicted reaction of the plant and enable comparisons with the operators' intuitive strategies are believed to be essential for practical success.

Effort vs. Performance

The gain in performance by a more sophisticated control scheme always has to be traded against the increase in cost due to the complexity of the control scheme – a complex scheme will not only cause cost for its implementation, but it will need more maintenance by better qualified people than a simple one. If a carefully chosen standard regulatory control layer leads to a close-to-optimal operation, there is no need for optimizing control. If the disturbances that affect profitability and cannot be handled well by the regulatory layer (in terms of economic performance) are slow, the combination of regulatory control and RTO is sufficient. In a more dynamic situation or for complex nonlinear multivariable plants, the idea of direct performance optimizing control should

be explored and implemented if significant gains can be realized in simulations.

Cross-References

- [Control and Optimization of Batch Processes](#)
- [Control Hierarchy of Large Processing Plants: An Overview](#)
- [Control Structure Selection](#)
- [Economic Model Predictive Control](#)
- [Extended Kalman Filters](#)
- [Kalman Filters](#)
- [Model Predictive Control in Practice](#)
- [Moving Horizon Estimation](#)
- [Particle Filters](#)
- [Real-Time Optimization of Industrial Processes](#)

Bibliography

- Angeli D, Amrit R, Rawlings J (2012) On average performance and stability of economic model predictive control. *IEEE Trans Autom Control* 57(7):1615–1626
- Bartusiak RD (2005) NMPC: a platform for optimal control of feed- or product-flexible manufacturing. In: *International Workshop on Assessment and Future Directions of NMPC*, Freudenstadt, Preprints, pp 3–14
- Diehl M, Amrit R, Rawlings J, Angeli D (2011) A Lyapunov function for economic optimizing model predictive control. *IEEE Trans Autom Control* 56(3):703–707
- Engell S (2006, 2007). Feedback control for optimal process operation. In: *Plenary Paper IFAC ADChEM*, Gramado, 2006. *J Process Control* 17:203–219
- Erdem G, Abel S, Morari M, Mazzotti M, Morbidelli M (2004) Automatic control of simulated moving beds. Part II: nonlinear isotherms. *Ind Eng Chem Res* 43:3895–3907
- Finkler T, Lucia S, Dogru M, Engell S (2013) A simple control scheme for batch time minimization of exothermic semi-batch polymerizations. *Indus Eng Chem Res* 52(2013):5906–5920
- Finkler TF, Kawohl M, Piechottka U, Engell S (2014) Realization of online optimizing control in an industrial semi-batch polymerization. *J Process Control* 24:399–414
- Grüne L (2013) Economic receding horizon control without terminal constraints. *Automatica* 49:725–734
- Haßkerl D, Lindscheid C, Subramanian S, Markert S, Gorak A, Engell S (2018) Dynamic performance optimization of a pilot-scale reactive distillation process by economics optimizing control. *Indus Eng Chem Res* 57(36):12165–12181
- Hebing L, Tran F, Brandt H, Engell S (2020) Robust optimizing control of fermentation processes based on a set of structurally different process models. *Indus Eng Chem Res* 59(6):2566–2580
- Helbig A, Abel O, Marquardt W (2000) Structural concepts for optimization based control of transient processes. In: *Nonlinear model predictive control*. Birkhäuser, Basel, pp 295–311
- Idris IAN, Engell S (2012) Economics-based NMPC strategies for the operation and control of a continuous catalytic distillation process. *J Process Control* 22:1832–1843
- Lucia S, Finkler T, Basak D, Engell S (2013) A new robust NMPC scheme and its application to a semi-batch reactor example. *J Process Control* 23(2013):1306–1319
- Marlin TE, Hrymak AN (1997) Real-time operations optimization of continuous processes. In: *Proceedings of CPC V, AIChE Symposium Series* 93, pp 156–164
- Morari M, Stephanopoulos G, Arkun Y (1980) Studies in the synthesis of control structures for chemical processes, part I. *AIChE J* 26:220–232
- Ochoa S, Wozny G, Repke JU (2010) Plantwide optimizing control of a continuous bioethanol production process. *J Process Control* 20:983–998
- Petersen LN, Poulsen NK, Niemann HH, Utzen C, Jørgensen JB (2017) Comparison of three control strategies for optimization of spray dryer operation. *J Process Control* 57:1–14
- Pham LC, Engell S (2011) A procedure for systematic control structure selection with application to reactive distillation. In: *Proceedings of 18th IFAC World Congress*, Milan, pp 4898–4903
- Qin SJ, Badgwell TA (2003) A survey of industrial model predictive control technology. *Control Eng Pract* 11:733–764
- Rao CV, Rawlings JB, Mayne DQ (2003) Constrained state estimation for nonlinear discrete-time systems. *IEEE Trans Autom Control* 48:246–258
- Rawlings J, Amrit R (2009) *Optimizing process economic performance using model predictive control*. In: *Nonlinear model predictive control – towards new challenging applications*. Springer, Berlin/Heidelberg, pp 119–138
- Rolandi PA, Romagnoli JA (2005) A framework for online full optimizing control of chemical processes. In: *Proceedings of ESCAPE 15*, Elsevier, pp 1315–1320
- Skogestad S (2000) Plantwide control: the search for the self-optimizing control structure. *J Process Control* 10:487–507
- Toumi A, Engell S (2004) Optimization-based control of a reactive simulated moving bed process for glucose isomerization. *Chem Eng Sci* 59:3777–3792
- Zanin AC, Tvrzská de Gouvea M, Odloak D (2000) Industrial implementation of a real-time optimization strategy for maximizing production of LPG in a FCC unit. *Comput Chem Eng* 24:525–531

Model-Free Reinforcement Learning-Based Control for Continuous-Time Systems

Kyriakos G. Vamvoudakis

The Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Abstract

This book entry discusses model-free reinforcement learning algorithms based on a continuous-time Q-learning framework. The presented schemes solve infinite-horizon optimization problems of linear time-invariant systems with completely unknown dynamics and single or multiple players/controllers. We first formulate the appropriate Q-functions (action-dependent value functions) and the tuning laws based on actor/critic structures for several cases including optimal regulation; Nash games, multi-agent systems, and cases with intermittent feedback.

Keywords

Q-learning · Games · Optimal control · Autonomy · Model-free · Intermittent learning

Introduction

Machine learning has been used for enabling adaptive autonomy (Vamvoudakis et al. 2015). Machine learning is grouped, in supervised, unsupervised, or reinforcement (RL), depending on the amount and quality of feedback about the system or task. In supervised learning, the feedback information provided to learning algorithms is a labeled training data set, and the objective is to build the system model representing the learned relation between the input, output, and system parameters. In unsupervised learning, no feedback information is provided to the algorithm, and the objective is to classify the

sample sets to different groups based on the similarity between the input samples. Finally, RL is a goal-oriented learning tool wherein the agent or decision-maker learns a policy to optimize a long-term reward by interacting with the environment. At each step, an RL agent gets evaluative feedback about the performance of its action, allowing it to improve the performance of subsequent actions (Sutton and Barto 2018; Bertsekas and Tsitsiklis 1996; Cao 2007; Liu et al. 2017).

In a control engineering context, RL bridges the gap between traditional optimal control and adaptive control algorithms (Krstić and Kanellakopoulos 1995; Tao et al. 2003; Ioannou and Fidan 2006; Hovakimyan and Cao 2010; Vamvoudakis et al. 2017b). The goal is to learn the optimal policy and value function for a potentially uncertain physical system. Unlike traditional optimal control, RL finds the solution to the Hamilton-Jacobi-Bellman (HJB) equation online in real time. On the other hand, unlike traditional adaptive controllers, which are not usually designed to be optimal in the sense of minimizing cost functionals, RL algorithms are optimal. This has motivated control system researchers to enable adaptive autonomy in an optimal manner by developing RL-based controllers. In continuous-time linear systems with multiple decision-makers and quadratic costs, one has to rely on solving complicated matrix Riccati equations that require complete knowledge of the system matrices and needs to be solved offline and then implemented online in the controller. In the era of complex and big data systems, modeling the processes exactly is most of the time infeasible, and offline solutions make the systems vulnerable to parameter changes (drift).

Q-learning is a model-free RL technique developed primarily for discrete-time systems (Watkins 1989). It learns an action-dependent value function that ultimately gives the expected utility of taking a given action in a given state and following the optimal policy thereafter. When such an action-dependent value function is learned, the optimal policy can be computed easily. The *biggest strength* of Q-learning is that

it is model-free. It has been proven in Watkins (1989) that for any finite Markov decision process, Q-learning eventually finds an optimal policy. In complex-systems, Q-learning needs to store a lot of data, which makes the algorithm infeasible. This problem can be solved effectively by using adaptation techniques. Specifically, Q-learning can be improved by using the universal function approximation property that allows us to solve difficult optimization problems online and forward in time. This makes it possible to apply the algorithm to larger problems, even when the state space is continuous and infinitely large.

The purpose of this entry is to show how to solve optimal control problems (single player games), multiplayer games with intermittent and continuous feedback, online by Q-learning in real time using data measured along the trajectories without requiring any information of the dynamics. It provides a truly model-free and dynamic framework for decision-making, since learning agents can change their objectives or optimality criteria on the fly.

Notation We denote $\underline{\lambda}(A)(\bar{\lambda}(A))$ as the minimum (maximum) eigenvalue of a matrix A . The half-vectorization, $\text{vech}(A)$, of a symmetric $n \times n$ matrix A is the $n(n+1)/2$ column vector obtained by vectorizing only the upper (or lower) triangular part of A , $\text{vech}(A) = [A_{1,1} \cdots A_{n,1} \ A_{2,2} \cdots A_{n,2} \cdots A_{n-1,n-1} \ A_{n-1,n} \ A_{n,n}]^T$. Finally, $(\cdot)^*$ symbolizes the optimal solution.

Optimal Regulation

The infinite horizon optimal control problem is formulated as a Q-learning (model-free) problem by using a quadratic in the state and control parametrization. An integral reinforcement learning approach will be used to design and derive tuning formulas for a critic approximator estimating the action-dependent value of the Q-function and for an actor approximator estimating the optimal control. The actor and the critic approximators are tuned simultaneously by measuring

information along the state and the control trajectories *without the need of an initial admissible (i.e., stabilizing) policy*; this is in contrast to existing Q-learning approaches that use sequential policy iteration algorithms which delay the convergence and require admissibility of the initial policy.

Consider the following linear time-invariant continuous-time system,

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad t \geq 0, \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is a measurable state vector, $u(t) \in \mathbb{R}^m$ is the control input, and $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are the plant and input matrices, respectively, that will be considered uncertain/unknown. It is assumed that the pair (A, B) is controllable. Our goal is to find a controller u that minimizes a cost functional of the form, $J(x(0); u) = \frac{1}{2} \int_0^\infty (x^T M x + u^T R u) dt$ with user-defined matrices $M \geq 0$, $R > 0$ of appropriate dimensions and $(\sqrt{M}, A)$ detectable.

The ultimate goal is to find the optimal value function V^* defined by,

$$V^*(x(t)) := \min_u \int_t^\infty \frac{1}{2} (x^T M x + u^T R u) d\tau, \quad \forall t, \quad (2)$$

given (1) but without any information of the system dynamics.

Model-Free Formulation

The value function needs to be parametrized as a function of the state x and the control u to represent the Q-function. The optimal value given by (2) after adding the Hamiltonian can be written as the following advantage function $Q(x, u) : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$,

$$\begin{aligned} Q(x, u) := & V^*(x) + \frac{1}{2} x^T P (Ax + Bu) \\ & + \frac{1}{2} (Ax + Bu)^T P x + \frac{1}{2} u^T R u \\ & + \frac{1}{2} x^T M x, \quad \forall x, u, \end{aligned} \quad (3)$$

where the optimal cost is $V^*(x) = \frac{1}{2} x^T P x$ with $P > 0$.

The Q-function (3) can be written in a compact quadratic in the state x and control u form as follows,

$$\begin{aligned} Q(x, u) &= \frac{1}{2} U^T \begin{bmatrix} P + M + PA + A^T P & PB \\ B^T P & R \end{bmatrix} U \\ &:= \frac{1}{2} U^T \begin{bmatrix} Q_{xx} & Q_{xu} \\ Q_{ux} & Q_{uu} \end{bmatrix} U \\ &:= \frac{1}{2} U^T \bar{Q} U, \quad \forall x, u, \end{aligned} \quad (4)$$

where $U := [x^T u^T]^T$, $Q_{xx} = P + M + PA + A^T P$, $Q_{xu} = PB$, $Q_{ux} = Q_{xu}^T = B^T P$, $Q_{uu} = R$, and $\bar{Q} = \begin{bmatrix} Q_{xx} & Q_{xu} \\ Q_{ux} & Q_{uu} \end{bmatrix} \in \mathbb{R}^{(n+m) \times (n+m)}$.

A model-free formulation can be found by solving $\frac{\partial Q(x, u)}{\partial u} = 0$ to write,

$$u^*(x) = \arg \min_u Q(x, u) = -Q_{uu}^{-1} Q_{ux} x. \quad (5)$$

The critic approximator will approximate the Q-function (4), and the actor approximator will approximate the optimal controller (5). Specifically $Q^*(x, u^*)$ can be written as,

$$\begin{aligned} Q^*(x, u^*) &= \frac{1}{2} U^T \begin{bmatrix} Q_{xx} & Q_{xu} \\ Q_{ux} & Q_{uu} \end{bmatrix} U \\ &:= \frac{1}{2} \text{vech}(\bar{Q})^T (U \otimes U) \end{aligned} \quad (6)$$

where $\text{vech}(\bar{Q}) \in \mathbb{R}^{\frac{1}{2}(n+m)(n+m+1)}$ a half-vectorization of the matrix $\bar{Q}$ where the off-diagonal elements are taken as $2\bar{Q}_{ij}$ and $\otimes$ denotes the Kronecker product quadratic polynomial basis vector with the elements $\{U_i U_j\}_{i=1 \dots n+m; j=1 \dots n+m}$.

By denoting as $W_c := \frac{1}{2} \text{vech}(\bar{Q})$, we can write (6) in a compact form as $Q^*(x, u^*) = W_c^T (U \otimes U)$, with $W_c \in \mathbb{R}^{\frac{1}{2}(n+m)(n+m+1)}$ the ideal weights given as $\text{vech}(Q_{xx}) := W_c[1 : \frac{n(n+1)}{2}]$, $\text{vech}(Q_{xu}) := W_c[\frac{n(n+1)}{2} + 1 : \frac{n(n+1)}{2} + nm]$ and $\text{vech}(Q_{uu}) := W_c[\frac{n(n+1)}{2} + 1 + nm : \frac{(n+m)(n+m+1)}{2}]$.

Since the ideal weights for computing Q^* and u^* are unknown, one must consider the following *weight estimates*. The critic approximator with $\hat{W}_c := \text{vech}(\hat{Q})$ can be written as,

$$\hat{Q}(x, u) = \hat{W}_c^T (U \otimes U),$$

where $\hat{W}_c \in \mathbb{R}^{\frac{1}{2}(n+m)(n+m+1)}$ are the estimated weights and $U := [x^T u^T]^T$. Similarly the actor approximator can be expressed as,

$$\hat{u}(x) = \hat{W}_a^T x,$$

where $\hat{W}_a \in \mathbb{R}^{n \times m}$ are the weight estimates; note also that the state x is serving as an activation function for the action approximator.

We shall define an error $e \in \mathbb{R}$ that we would like to eventually drive to zero by picking appropriately the tuning law for $\hat{W}_c$. In order to do that, we shall take into consideration the integral Bellman equation, and using actual values for the Q-function, we can rewrite the integral Bellman error as,

$$\begin{aligned} e &:= \hat{Q}(x(t), u(t)) - \hat{Q}(x(t-T), u(t-T)) \\ &\quad + \frac{1}{2} \int_{t-T}^T (x^T M x + u^T R u) d\tau \\ &= \hat{W}_c^T (U(t) \otimes U(t)) \\ &\quad + \frac{1}{2} \int_{t-T}^T (x^T M x + u^T R u) d\tau \\ &\quad - \hat{W}_c^T (U(t-T) \otimes U(t-T)). \end{aligned}$$

Now for the actor approximator, we can define the error $e_a \in \mathbb{R}^m$ as follows, $e_a := \hat{W}_a^T x + \hat{Q}_{uu}^{-1} \hat{Q}_{ux} x$, where the values of $\hat{Q}_{uu}^{-1}$ and $\hat{Q}_{ux}$ are going to be extracted from the vector $\hat{W}_c$. Now we shall find tuning updates for $\hat{W}_c$ and $\hat{W}_a$ such that the errors e and e_a go to zero.

By following adaptive control techniques, we can define the squared norm of errors e and e_a as,

$$K_1 = \frac{1}{2} \|e\|^2, \quad (7)$$

$$K_2 = \frac{1}{2} \|e_a\|^2. \quad (8)$$

The estimate of $\hat{W}_c$ (in order to guarantee that $e \rightarrow 0$ and $\hat{W}_c \rightarrow W_c$) for the critic weights can be constructed by applying gradient descent in (7), using the chain rule and normalizing (following adaptive control theory) as,

$$\dot{\hat{W}}_c = -\alpha_c \frac{\partial K_1}{\partial \hat{W}_c} = -\alpha_c \frac{\sigma}{(1 + \sigma^T \sigma)^2} e^T,$$

where $\sigma := U(t) \otimes U(t) - U(t-T) \otimes U(t-T)$ and $\alpha_c \in \mathbb{R}^+$ is a constant gain that determines the speed of convergence.

Similarly, the gradient descent estimate of $\hat{W}_a$ for the actor weights can be constructed by differentiating (8) as,

$$\dot{\hat{W}}_a = -\alpha_a \frac{\partial K_2}{\partial \hat{W}_a} = -\alpha_a x e_a^T,$$

where $\alpha_a \in \mathbb{R}^+$ is a constant gain that determines the speed of convergence.

Algorithm 1 gives the pseudocode for the approach.

Algorithm 1 Q-learning for optimal regulation

- 1: **procedure**
 - 2: Start with initial conditions $x(0)$, and random initial weights $\hat{W}_c(0)$, $\hat{W}_a(0)$, for the critic and the actor.
 - 3: Propagate t , $x(t)$.
 - 4: Propagate $\hat{W}_c(t)$, $\hat{W}_a(t)$.
 - 5: Compute the Q-function $\hat{Q}$ and the optimal control $\hat{u}$.
 - 6: **end procedure**
-

More details about model-free optimal regulation are given in Vamvoudakis (2017b), and a model-free kinodynamic motion planning framework is provided in Kontoudis and Vamvoudakis (2019).

Nash Games

Consider N players playing in a Nash dynamic game. The N agent/decision-makers can be non-cooperative and also can cooperate in teams.

Each of the agents has access to the full state of the system. Since solving Nash game requires complete knowledge of the centralized system dynamics and complicated offline computation, we present a completely mode-free algorithm to solve the coupled Riccati equations arising in multi-player non-zero sum Nash games in deterministic systems. A parametrized Q-function is derived for every of the N -players in the game that depends on the state and the control inputs of all the players. Then we derive mode-free controllers based on the Q-function parametrization, and then we use $2N$ -universal approximators such to approximate the cost and the control of every player, by using, namely, a critic and an actor network.

Consider the following linear time-invariant continuous-time system,

$$\dot{x}(t) = Ax(t) + \sum_{j=1}^N B_j u_j(t), \quad x(0) = x_0, \quad t \geq 0, \quad (9)$$

where $x(t) \in \mathbb{R}^n$ is a measurable state vector, $u_j(t) \in \mathbb{R}^{m_j}$, $j \in \mathcal{N} := \{1, \dots, N\}$ is each control input (or player), and $A \in \mathbb{R}^{n \times n}$, $B_j \in \mathbb{R}^{n \times m_j}$, $j \in \mathcal{N}$ are the plant and input matrices, respectively, that will be considered uncertain/unknown. It is assumed that the unknown pairs (A, B_j) , $\forall j \in \mathcal{N}$ are controllable. The ultimate goal is to find the optimal value functions V_i^* , $\forall i \in \mathcal{N}$ defined by,

$$V_i^*(x(t)) := \min_{u_i} \int_t^\infty \frac{1}{2} \left(x^T H_i x + \sum_{j=1}^N u_j^T R_{ij} u_j \right) d\tau, \quad \forall t, \quad (10)$$

with user-defined matrices $H_i \succeq 0$, $R_{ij} \succ 0$, $\forall i, j \in \mathcal{N}$ of appropriate dimensions and

$(\sqrt{H_i}, A)$, $\forall i \in \mathcal{N}$ are detectable, and without any information of the system matrices A and B_i , $\forall i \in \mathcal{N}$ that appear in (9).

Model-Free Formulation

The value functions need to be parametrized as a function of the state x and the controls u_i , $\forall i \in \mathcal{N}$ to represent the Q-function for each player in the game. The optimal value given by (10) after adding the Hamiltonian can be written as the following advantage function $Q_i(x, u_i, u_{-i}) : \mathbb{R}^{n+\sum_{j=1}^N m_j} \rightarrow \mathbb{R}$,

$$\begin{aligned} Q_i(x, u_i, u_{-i}) &:= V_i^*(x) + \frac{1}{2}x^T P_i \left(Ax + \sum_{j=1}^N B_j u_j \right) \\ &\quad + \frac{1}{2} \left(Ax + \sum_{j=1}^N B_j u_j \right)^T P_i x \\ &\quad + \frac{1}{2} \sum_{j=1}^N u_j^T R_{ij} u_j + \frac{1}{2}x^T H_i x, \\ &\quad \forall x, u_i, u_{-i}, \forall i \in \mathcal{N}, \end{aligned}$$

where the optimal cost is $V_i^*(x) = x^T P_i x$ with $P_i \succ 0$.

Each player's Q-function can be written in a compact quadratic in the state x and controls u_i , $i \in \mathcal{N}$ form as shown next.

$$\begin{aligned} Q_i(x, u_i, u_{-i}) &= \frac{1}{2} U_i^T \begin{bmatrix} P_i + H_i + P_i A + A^T P_i & P_i B_i & P_i B_1 & \cdots & P_i B_{i-1} & P_i B_{i+1} & \cdots & P_i B_N \\ B_i^T P_i & R_{ii} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ B_1^T P_i & 0 & R_{i1} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ B_{i-1}^T P_i & 0 & \cdots & \cdots & R_{ii-1} & 0 & \cdots & 0 \\ B_{i+1}^T P_i & 0 & \cdots & \cdots & \cdots & R_{ii+1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ B_N^T P_i & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & R_{iN} \end{bmatrix} U_i \\ &:= \frac{1}{2} U_i^T \begin{bmatrix} Q_{xx}^i & Q_{xu_i}^i & Q_{xu_1}^i & \cdots & Q_{xu_{i-1}}^i & Q_{xu_{i+1}}^i & \cdots & Q_{xu_N}^i \\ Q_{u_i x}^i & Q_{u_i u_i}^i & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ Q_{u_1 x}^i & 0 & Q_{u_1 u_1}^i & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ Q_{u_{i-1} x}^i & 0 & \cdots & \cdots & Q_{u_{i-1} u_{i-1}}^i & 0 & \cdots & 0 \\ Q_{u_{i+1} x}^i & 0 & \cdots & \cdots & \cdots & Q_{u_{i+1} u_{i+1}}^i & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ Q_{u_N x}^i & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & Q_{u_N u_N}^i \end{bmatrix} U_i \\ &:= \frac{1}{2} U_i^T \bar{Q}^i U_i, \quad \forall x, u_i, \forall i \in \mathcal{N}, \end{aligned}$$

where $U_i := [x^T \ u_i^T \ u_1^T \ \cdots \ u_{i-1}^T \ u_{i+1}^T \ \cdots \ u_N^T]^T$.

A model-free formulation of the best policies can be found by solving $\frac{\partial Q_i(x, u_i, u_{-i})}{\partial u_i} = 0$ to write,

$$\begin{aligned} u_i^*(x) &= \arg \min_{u_i} Q_i(x, u_i, u_{-i}) \\ &= -(Q_{u_i u_i}^i)^{-1} Q_{u_i x}^i x, \quad \forall i \in \mathcal{N}. \end{aligned}$$

Since we do not know the optimal values, we should estimate Q_i^* and u_i^* with the following actual values for the critic with $\hat{W}_{ic} := \frac{1}{2} \text{vech}(\hat{\bar{Q}}^i)$,

$$\hat{Q}_i(x, u_i, u_{-i}) = \hat{W}_{ic}^T (U_i \otimes U_i), \quad \forall i \in \mathcal{N},$$

where $\hat{W}_{ic} \in \mathbb{R}^{\frac{1}{2}(n+\sum_{j=1}^N m_j)(n+\sum_{j=1}^N m_j+1)}$ are the estimated critic weights and $U_i := [x^T u_i^T u_1^T \cdots u_{i-1}^T u_{i+1}^T \cdots u_N^T]^T$, and for the actor

$$\hat{u}_i(x) = \hat{W}_{ia}^T x, \forall i \in \mathcal{N},$$

where $\hat{W}_{ia} \in \mathbb{R}^{n \times m_i}$ are the estimated actor weights; note also that the state x is serving as an activation function for the action network. The tuning laws for the actor and the critic can be defined in a similar way with the Optimal Regulation part.

Algorithm 2 gives the pseudocode for the approach.

Algorithm 2 Nash Q-learning

- 1: **procedure**
 - 2: Start with initial conditions for every agent $x(0)$ and random initial weights $\hat{W}_{ic}(0), \hat{W}_{ia}(0)$ for the critic and actor approximators.
 - 3: Propagate $t, x(t)$.
 - 4: Propagate $\hat{W}_{ic}(t), \hat{W}_{ia}(t)$.
 - 5: Compute the Q-function $\hat{Q}_i$ and the control $\hat{u}_i$.
 - 6: **end procedure**
-

More details about model-free Nash games are given in Vamvoudakis (2015).

Multi-agent Systems

Multi-agent system interactions are modeled by a fixed strongly connected *graph* $\mathcal{G} = (\mathcal{V}_{\mathcal{G}}, \mathcal{E}_{\mathcal{G}})$ defined by a finite set $\mathcal{V}_{\mathcal{G}} = \{n_1, \dots, n_N\}$ of N nodes that represent agents and a set of edges $\mathcal{E}_{\mathcal{G}} \subseteq \mathcal{V}_{\mathcal{G}} \times \mathcal{V}_{\mathcal{G}}$ that represent interagent information exchange links. The set of *neighbors* of a node n_i is defined as the set of nodes with edges incoming to n_i and is denoted by $\mathcal{N}_i = \{n_j : (n_j, n_i) \in \mathcal{E}_{\mathcal{G}}\}$. The graph is simple (i.e., it has not repeated edges) and has no self-loops (i.e., $(n_i, n_i) \notin \mathcal{E}_{\mathcal{G}}, \forall n_i \in \mathcal{V}_{\mathcal{G}}$). The *connectivity matrix* $\mathcal{A}_{\mathcal{G}} = [\alpha_{ij}]_{N \times N}$ of the graph $\mathcal{G}$ is an $N \times N$ matrix of the form

$$\alpha_{ij} \begin{cases} > 0 & \text{if } (n_j, n_i) \in \mathcal{E}_{\mathcal{G}} \\ = 0 & \text{otherwise,} \end{cases}$$

where α_{ij} is the *weight* associated with the edge $(n_j, n_i) \in \mathcal{E}_{\mathcal{G}}$. The *degree* matrix D of the graph $\mathcal{G}$ is an $N \times N$ diagonal matrix whose i th entry of the main diagonal is the weighted degree $d_i = \sum_{j \in \mathcal{N}_i} \alpha_{ij}$ of node i (i.e., i th row sum of $\mathcal{A}_{\mathcal{G}}$). The number of neighbors of agent i is denoted by $|\mathcal{N}_i|$ which is equal to d_i . A cooperative Q-learning approach is now developed to enable the agents in large networks to synchronize to the behavior of an unknown leader by each optimizing a distributed performance criterion that depends only on a subset of the agents in the network. The novel distributed Q-functions are parametrized as functions of the tracking error, control, and adversarial inputs in the neighborhood. In the developed approach, the agents coordinate with the neighbors in order to pick their minimizing model-free policies in such a way to guarantee convergence to a graphical Nash equilibrium and also attenuation of maximizing worst-case adversarial inputs. A structure of 2-actors and a single critic approximators are used for each agent in the network. This eventually solves the complexity issues that arise in Q-learning. The 2-actors are used to approximate the optimal control input and the worst-case adversarial input, while the critic approximator is used to approximate the optimal cost of each of the coupled optimizations. Effective tuning laws are proposed to solve the model-free cooperative game problem while also guaranteeing closed-loop stability of the equilibrium point.

Consider a networked-system $\mathcal{G}$, consisting of N agents each modeled $\forall i \in \mathcal{N} := \{1, \dots, N\}$ by the following dynamics,

$$\begin{aligned} \dot{x}_i(t) &= Ax_i(t) + B_i u_i(t) + D_i v_i(t), \\ x_i(0) &= x_{i0}, t \geq 0, \end{aligned}$$

where $x_i(t) \in \mathbb{R}^n$ is a measurable state vector available for feedback by each agent and known initial conditions $x_i(0), u_i(t) \in \mathbb{R}^{m_i}, i \in \mathcal{N} := \{1, \dots, N\}$ is each control input (or minimizing player), $v_i(t) \in \mathbb{R}^{l_i}, i \in \mathcal{N} := \{1, \dots, N\}$

is each adversarial input (or maximizing player), and $A \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{n \times m_i}$, $D_i \in \mathbb{R}^{n \times l_i}$, $i \in \mathcal{N}$ are the plant, control input, and adversarial input matrices, respectively, that will be considered uncertain/unknown. It is assumed that the unknown pairs (A, B_i) , $\forall i \in \mathcal{N}$ are stabilizable. We have a total of $2N$ players/controllers that select values for $u_i(t)$, $t \geq 0$, $i \in \mathcal{N}$, and $v_i(t)$, $t \geq 0$, $i \in \mathcal{N}$. The agents in the network seek to cooperatively asymptotically track the state of a leader node/exosystem with dynamics $\dot{x}_L = Ax_L$, i.e., $x_i(t) - x_L(t) \rightarrow 0$, $\forall i \in \mathcal{N}$, while *simultaneously satisfying user-defined distributed performances*. Now we shall proceed to the design of the user-defined *distributed performances*. For that reason, we shall define the following *neighborhood tracking error* for every agent,

$$e_i := \sum_{j \in \mathcal{N}_i} a_{ij}(x_i - x_j) + g_i(x_i - x_L), \forall i \in \mathcal{N}, \quad (11)$$

where $g_i \in \mathbb{R}^+$ is the pinning gain that shows if an agent is pinned to the leader node (i.e., $g_i \neq 0$) and it is non-zero for at least one node.

The dynamics of (11) are given by,

$$\begin{aligned} \dot{e}_i &= Ae_i + (d_i + g_i)(B_i u_i + D_i v_i) \\ &- \sum_{j \in \mathcal{N}_i} a_{ij}(B_j u_j + D_j v_j), \forall i \in \mathcal{N}, \end{aligned} \quad (12)$$

with $e_i \in \mathbb{R}^n$.

The cost functionals associated with each agent $i \in \mathcal{N}$, which depend on the tracking

error e_i , the control u_i , the controls in the neighborhood of agent i given as $u_{\mathcal{N}_i} := \{u_j : j \in \mathcal{N}_i\}$, the adversarial input v_i , and the adversarial inputs in the neighborhood of agent i given as $v_{\mathcal{N}_i} := \{v_j : j \in \mathcal{N}_i\}$, with an ultimate goal to find the distributed optimal value functions V_i^* , $\forall i \in \mathcal{N}$ defined by,

$$\begin{aligned} V_i^*(e_i(t)) &:= \min_{u_i} \max_{v_i} \int_t^\infty \frac{1}{2} \left(e_i^\top H_i e_i \right. \\ &\quad \left. + (u_i^\top R_{ii} u_i - \gamma_{ii}^2 v_i^\top v_i) \right. \\ &\quad \left. + \sum_{j \in \mathcal{N}_i} (u_j^\top R_{ij} u_j - \gamma_{ij}^2 v_j^\top v_j) \right) d\tau, \\ &\quad \forall t, \forall i \in \mathcal{N}, \end{aligned} \quad (13)$$

given (12), but without any information of the system matrices A , B_i , D_i , $\forall i \in \mathcal{N}$ and pinning gains g_i , $\forall i \in \mathcal{N}$ and matrices $H_i \succeq 0$, $R_{ii} \succ 0$, $R_{ij} \succeq 0$, $\forall i, j \in \mathcal{N}$ of appropriate dimensions, $\gamma_{ii}, \gamma_{ij} \in \mathbb{R}^+$, $\forall i \in \mathcal{N}$ and $(\sqrt{H_i}, A)$, $\forall i \in \mathcal{N}$ are detectable.

Model-Free Formulation

The value functions need to be parametrized as functions of the neighborhood tracking error e_i , the controls u_i and $u_{\mathcal{N}_i}$, and the adversarial inputs v_i and $v_{\mathcal{N}_i}$ to represent the distributed Q-function, i.e., sparse cooperative learning, for each agent in the game. The optimal value given by (13) after adding the Hamiltonian can be written as the following distributed advantage function $Q_i(e_i, u_i, u_{\mathcal{N}_i}, v_i, v_{\mathcal{N}_i}) : \mathbb{R}^{n+m_i+l_i+\sum_{j \in \mathcal{N}_i}(m_j+l_j)} \rightarrow \mathbb{R}$,

$$\begin{aligned} &Q_i(e_i, u_i, u_{\mathcal{N}_i}, v_i, v_{\mathcal{N}_i}) \\ &:= V_i^*(e_i) + \frac{\partial V_i^*}{\partial e_i} \left(Ae_i + (d_i + g_i)(B_i u_i + D_i v_i) - \sum_{j \in \mathcal{N}_i} a_{ij}(B_j u_j + D_j v_j) \right) \\ &\quad + \frac{1}{2} \left(e_i^\top H_i e_i + (u_i^\top R_{ii} u_i - \gamma_{ii}^2 v_i^\top v_i) + \sum_{j \in \mathcal{N}_i} (u_j^\top R_{ij} u_j - \gamma_{ij}^2 v_j^\top v_j) \right), \\ &\quad \forall e_i, u_i, u_{\mathcal{N}_i}, v_i, v_{\mathcal{N}_i}, \forall i \in \mathcal{N}, \end{aligned}$$

where the optimal cost is $V_i^*(e_i) = \frac{1}{2}e_i^T P_i e_i, \forall i \in \mathcal{N}$.

Each agent's distributed Q-function can be written in a compact quadratic in the neigh-

borhood tracking error e_i , controls $u_i, u_{\mathcal{N}_i}$, and adversarial inputs $v_i, v_{\mathcal{N}_i}$ distributed form as,

$$Q_i(e_i, u_i, u_{\mathcal{N}_i}, v_i, v_{\mathcal{N}_i}) = \frac{1}{2}U_i^T \begin{bmatrix} P_i + H_i + P_i A + A^T P_i & (d_i + g_i)P_i B_i & -\text{row}(a_{ij} P_i B_j)_{j \in \mathcal{N}_i} & (d_i + g_i)P_i D_i & -\text{row}(a_{ij} P_i D_j)_{j \in \mathcal{N}_i} \\ (d_i + g_i)B_i^T P_i & R_{ii} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\text{col}(a_{ij} B_j^T P_i)_{j \in \mathcal{N}_i} & \mathbf{0} & \text{diag}(R_{ij})_{j \in \mathcal{N}_i} & \mathbf{0} & \mathbf{0} \\ (d_i + g_i)D_i^T P_i & \mathbf{0} & \mathbf{0} & -\gamma_{ii}^2 & \mathbf{0} \\ -\text{col}(a_{ij} D_j^T P_i)_{j \in \mathcal{N}_i} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\text{diag}(\gamma_{ij}^2)_{j \in \mathcal{N}_i} \end{bmatrix} U_i$$

$$:= \frac{1}{2}U_i^T \begin{bmatrix} Q_{e_i e_i}^i & Q_{e_i u_i}^i & Q_{e_i u_{\mathcal{N}_i}}^i & Q_{e_i v_i}^i & Q_{e_i v_{\mathcal{N}_i}}^i \\ Q_{u_i e_i}^i & Q_{u_i u_i}^i & Q_{u_i u_{\mathcal{N}_i}}^i & Q_{u_i v_i}^i & Q_{u_i v_{\mathcal{N}_i}}^i \\ Q_{u_{\mathcal{N}_i} e_i}^i & Q_{u_{\mathcal{N}_i} u_i}^i & Q_{u_{\mathcal{N}_i} u_{\mathcal{N}_i}}^i & Q_{u_{\mathcal{N}_i} v_i}^i & Q_{u_{\mathcal{N}_i} v_{\mathcal{N}_i}}^i \\ Q_{v_i e_i}^i & Q_{v_i u_i}^i & Q_{v_i u_{\mathcal{N}_i}}^i & Q_{v_i v_i}^i & Q_{v_i v_{\mathcal{N}_i}}^i \\ Q_{v_{\mathcal{N}_i} e_i}^i & Q_{v_{\mathcal{N}_i} u_i}^i & Q_{v_{\mathcal{N}_i} u_{\mathcal{N}_i}}^i & Q_{v_{\mathcal{N}_i} v_i}^i & Q_{v_{\mathcal{N}_i} v_{\mathcal{N}_i}}^i \end{bmatrix} U_i := \frac{1}{2}U_i^T \bar{Q}^i U_i, \forall i \in \mathcal{N},$$

where $U_i := [e_i^T u_i^T u_{\mathcal{N}_i}^T v_i^T v_{\mathcal{N}_i}^T]^T$, $\mathbf{0}$ are zero matrices of appropriate dimensions, $\text{diag}(R_{ij})_{j \in \mathcal{N}_i}$ and $\text{diag}(\gamma_{ij})_{j \in \mathcal{N}_i}$ are stacked diagonal neighborhood matrices, $\text{col}(a_{ij} B_j^T P_i)_{j \in \mathcal{N}_i}$ and $\text{col}(a_{ij} D_j^T P_i)_{j \in \mathcal{N}_i}$ are stacked column neighborhood matrices, and $\text{row}(a_{ij} P_i B_j)_{j \in \mathcal{N}_i}$ and $\text{row}(a_{ij} P_i D_j)_{j \in \mathcal{N}_i}$ are stacked row neighborhood matrices.

A model-free formulation of the minimizing player can be found by solving $\frac{\partial Q_i(e_i, u_i, u_{\mathcal{N}_i}, v_i, v_{\mathcal{N}_i})}{\partial u_i} = 0$ to write,

$$u_i^*(e_i) = \arg \min_{u_i} Q_i(e_i, u_i, u_{\mathcal{N}_i}, v_i, v_{\mathcal{N}_i})$$

$$= -(Q_{u_i u_i}^i)^{-1} Q_{u_i e_i}^i e_i, \forall i \in \mathcal{N},$$

and a model-free formulation of the maximizing player can be found by solving $\frac{\partial Q_i(e_i, u_i, u_{\mathcal{N}_i}, v_i, v_{\mathcal{N}_i})}{\partial v_i} = 0$ to write,

$$v_i^*(e_i) = \arg \max_{v_i} Q_i(e_i, u_i, u_{\mathcal{N}_i}, v_i, v_{\mathcal{N}_i})$$

$$= -(Q_{v_i v_i}^i)^{-1} Q_{v_i e_i}^i e_i, \forall i \in \mathcal{N}.$$

The critic approximator is given by,

$$\hat{Q}_i(e_i, u_i, u_{\mathcal{N}_i}, v_i, v_{\mathcal{N}_i}) = \hat{W}_{ic}^T (U_i \otimes U_i), \forall i \in \mathcal{N},$$

where $\hat{W}_{ic}$ are estimated critic weights of the optimal weights and $U_i := [e_i^T u_i^T u_{\mathcal{N}_i}^T v_i^T v_{\mathcal{N}_i}^T]^T$; the control actor approximator is given as

$$\hat{u}_i(e_i) = \hat{W}_{ia}^T e_i, \forall i \in \mathcal{N},$$

where $\hat{W}_{ia} \in \mathbb{R}^{n \times m_i}$ are estimated actor weights of the optimal weights; and the adversarial actor approximator is given by,

$$\hat{v}_i(e_i) = \hat{W}_{id}^T e_i, \forall i \in \mathcal{N},$$

where $\hat{W}_{id} \in \mathbb{R}^{n \times l_i}$ are estimated actor weights of the optimal weights.

The tuning laws for the critic and the actor can be found by using gradient descent as in the Optimal Regulation part.

Algorithm 3 gives the pseudocode for the approach.

Algorithm 3 Multi-agent Q-learning

-
- 1: **procedure**
 - 2: Start with initial conditions for every agent $x_i(0)$, and random initial weights $\hat{W}_{ic}(0)$, $\hat{W}_{ia}(0)$, $\hat{W}_{id}(0)$, for the critic, actor, and worst-case adversarial approximator for each agent.
 - 3: Propagate t , $x_i(t)$, and $x_j(t)$ in the neighborhood to compute e_i .
 - 4: Propagate $\hat{W}_{ic}(t)$, $\hat{W}_{ia}(t)$, $\hat{W}_{id}(t)$.
 - 5: Compute the Q-function $\hat{Q}_i$, the control $\hat{u}_i$, and the adversarial input $\hat{v}_i$ for each agent.
 - 6: **end procedure**
-

More details about multi-agent model-free learning can be found in Vamvoudakis and Hespanha (2018), Vamvoudakis (2017a), and Yang et al. (2018b).

Networked Systems with Intermittent/Hybrid Feedback

In conventional implementations of control systems, control tasks are usually executed by periodically sampling the plant's output and updating the control inputs. Selection of sampling period is traditionally performed during the control design stage, having in mind the trade-off between the reconstruction of the continuous-time signal and the load on the computer (Heemels et al. 2012). In many applications, however, conventional design methods may not represent an efficient solution. In network control systems (NCS) where communication channels are shared with multiple and possibly remotely located sensors, actuators, and controllers (Hespanha et al. 2007), periodic sampling, computation, and transmission of data could result in inefficient use of resources of bandwidth and energy. We show a new model-free event-triggered optimal control algorithm for continuous-time linear systems. In order to provide a model-free solution, we adopt a Q-learning framework with a critic network to approximate the optimal cost and a zero-order hold actor network to approximate the optimal control.

Consider the system described by (1). In order to save resources, the controller will work with a sampled version of the state. For that reason, one needs to introduce a *sampled-data component*

that is characterized by a monotone increasing sequence of sampling instants (broadcast release times) $\{r_j\}_{j=0}^{\infty}$, where r_j is the j -th consecutive sampling instant (impulse times) that satisfies $0 \leq t_0 < t_1 < \dots < r_j < \dots$ and $\lim_{j \rightarrow \infty} r_j = \infty$. The output of the *sampled-data component* is a sequence of sampled states $\hat{x}$ with $\hat{x} = x(r_j)$ for all $t \in (r_j, r_{j+1}]$ and $j \in \mathbb{N}$. The controller maps the sampled state onto a control vector u , which after using a zero-order hold (ZOH) becomes a continuous-time input signal. For simplicity we will assume that the sampled-data system has zero task delays.

In order to decide when to trigger an event, we will define the gap between the current state $x(t)$ and the sampled state $\hat{x}(t)$ as, $e(t) := \hat{x}(t) - x(t)$, where,

$$\hat{x}(t) = \begin{cases} x(r_j), & \forall t \in (r_j, r_{j+1}] \\ x(t), & t = r_j. \end{cases}$$

The ultimate goal is to find a value function with a quantified guaranteed performance that is ϵ -close to the time-triggered optimal value function V^* (value with infinite controller bandwidth) as defined in (2) and without any information of the system matrices of (1).

In order to reduce the communication between the controller and the plant, one needs to use an event-triggered version of the Riccati equation by introducing a *sampled-data component* with aperiodic controller updates that ensure a certain condition on the state of the plant to guarantee stability and performance as we will see in the subsequent analysis. For that reason, the control input uses the sampled-state information instead of the true one to write,

$$u_d^*(\hat{x}) = -R^{-1}B^T P_d \hat{x}, \quad \forall \hat{x},$$

where with $P_d \in \mathbb{R}^{n \times n}$ a unique symmetric positive matrix. Note that as the bandwidth approaches infinity, one has $V^*(\hat{x}) \rightarrow V^*(x)$.

Model-Free Formulation

The Q-function is given as in (3), but instead of u , we shall use u_d that uses the sampled-state information instead of the true one.

In contrast to existing event-triggered designs, where complete knowledge of the system dynamics is required, the aforementioned method is able to obviate this requirement by using the intermittent Q-learning algorithm. The actor-critic approximator structure is employed to co-design the event-triggering condition and controller. The combined closed-loop system can be written as an impulsive system, which is proved to have an asymptotically stable equilibrium point without any Zeno behavior. A qualitative performance analysis of the Q-learning is given in comparison to the continuous optimal feedback and the degree of sub-optimality is established.

The update for the critic is the same as the one for the Q-function given in (3), but the actor follows an impulsive update of the form,

$$\begin{cases} \dot{\hat{W}}_a = 0, & \forall t \in (r_j, r_{j+1}] \\ \hat{W}_a^+ = \hat{W}_a - \alpha_a \frac{1}{(1+x(t)^T x(t))} \frac{\partial K_2}{\partial \hat{W}_a} \\ \quad = \hat{W}_a - \alpha_a \frac{x(t)}{(1+x(t)^T x(t))} e_a^T, & t = r_j \end{cases}$$

where $\alpha_a \in \mathbb{R}^+$ is a constant gain that determines the speed of convergence.

Algorithm 4 gives the pseudocode for the approach.

Algorithm 4 Intermittent Q-learning for networked control

- 1: **procedure**
 - 2: Start with initial conditions $x(0)$, $\hat{x}(0)$, and random initial weights $\hat{W}_c(0)$, $\hat{W}_a(0)$, for the critic and the actor.
 - 3: Propagate t , $x(t)$, $\hat{x}(t)$, $\forall t \in (r_j, r_{j+1}]$.
 - 4: **if** $\|e\|^2 = \frac{(1-\beta^2)\lambda(M)}{L^2\lambda(R)} \|x\|^2 + \frac{\lambda(R)}{L^2\lambda(R)} \|\hat{W}_a^T x\|^2$ (where $\beta \in (0, 1)$ and L is a Lipschitz constant) **then**, then $\hat{x}(t) := x(t)$, $\forall t \in (r_j, r_{j+1}]$, and compute the control $\hat{u}_d = \hat{W}_a^T \hat{x}$.
 - 5: **else** $\hat{x}(t) = x(r_j)$, $\forall t \in (r_j, r_{j+1}]$, and use the previous control input with zero-order hold.
 - 6: **end if**
 - 7: Propagate $\hat{W}_c(t)$, $\hat{W}_a(t)$.
 - 8: Compute the Q-function $\hat{Q}$.
 - 9: **end procedure**
-

More details about model-based and model-free intermittent RL algorithms are given in Vamvoudakis (2014), Vamvoudakis and Ferraz (2018), Yang et al. (2019), Yang et al. (2018a), and Vamvoudakis and Ferraz (2016).

Optimal Tracking with Intermittent Feedback

We have extended the work presented before, to consider event-triggered optimal tracking. In order to do that, we have augmented the system and the reference dynamics to convert the tracking problem into a regulator one by appropriately choosing the discount factor and the weighting of the tracking error in the performance. For more details, see Vamvoudakis et al. (2017a) and Vamvoudakis (2016).

Summary and Future Directions

The use of Q-learning to solve model-free optimization-based control problems is an area that has attracted a significant amount of research attention in recent years and will continue to grow as autonomy becomes a necessity. The developed methods implicitly solve the required design equations without ever explicitly solving them. Different aspects of the control inputs (decision-makers) in terms of cooperation, altruistic vs. selfish behavior, and intermittent feedback, are explored.

The integration of Q-learning with control methods could advance the human-agent feedback loops while optimizing performance and advancing data decision models. Note that all the aforementioned algorithms use full-state feedback. An interesting direction to pursue would be to extend the approaches to robot motion planning that combine game theory, formal control synthesis, and learning. Finally, existing Q-learning techniques for large-scale dynamic systems require spending a long time to gather large amount of data for learning an optimal solution. Moreover, availability of large amounts of data requires novel decision and control schemes that

focus selectively on the data that are relevant for the current situation and ignore relatively unimportant details. The demands for fast response based on available information in a large-scale system impose new requirements for fast and efficient decision. The future work is to design new classes of learning-based controllers that lead to satisfactory and good enough solutions that deliver prescribed aspiration levels of satisfactory for large-scale systems that require making fast and skillful decisions in highly constrained, uncertain, and adversary environments with overload information.

Cross-References

- [Event-Triggered and Self-Triggered Control](#)
- [Optimal Control and Mechanics](#)
- [Multiagent Reinforcement Learning](#)

Acknowledgments This material is based upon the work supported by NSF under Grant Numbers CPS-1851588, SATC-1801611, and S&AS-1849198, by ARO under Grant Number W911NF1910270, and by Minerva Research Initiative under Grant Number N00014-18-1-2160.

Bibliography

- Bertsekas DP, Tsitsiklis JN (1996) *Neuro-dynamic programming*. Athena Scientific, Belmont
- Cao X-R (2007) *Stochastic learning and optimization*. Springer US, New York
- Heemels W, Johansson KH, Tabuada P (2012) An introduction to event-triggered and self-triggered control. In: 2012 IEEE 51st IEEE conference on decision and control (CDC), pp 3270–3285. IEEE
- Hespanha JP, Naghshtabrizi P, Xu Y (2007) A survey of recent results in networked control systems. *Proc IEEE* 95(1):138–162
- Hovakimyan N, Cao C (2010) *L1 adaptive control theory: guaranteed robustness with fast adaptation*. Advances in design and control. Society for Industrial and Applied Mathematics (SIAM), Philadelphia
- Ioannou P, Fidan B (2006) *Adaptive control tutorial*. Advances in design and control. Society for Industrial and Applied Mathematics (SIAM), Philadelphia
- Kontoudis GP, Vamvoudakis KG (2019) Kinodynamic motion planning with continuous-time Q-learning: an online, model-free, and safe navigation framework. *IEEE Trans Neural Netw Learn Syst* 30(12):3803–3817
- Krstić M, Kanellakopoulos I (1995) *Nonlinear and adaptive control design*. Adaptive and learning systems for Signal processing, communication and control. Wiley, New York
- Liu D, Wei Q, Wang D, Yang X, Li H (2017) *Adaptive dynamic programming with application in optimal control*. Springer-Verlag, London
- Sutton RS, Barto AG (2018) *Reinforcement learning: an introduction vol 1*, 2nd edn. MIT Press, Cambridge
- Tao G (2003) *Adaptive control design and analysis*. Adaptive and cognitive dynamic systems: signal processing, learning, communication and control. Wiley, New York
- Vamvoudakis KG (2014) Event-triggered optimal adaptive control algorithm for continuous-time nonlinear systems. *IEEE/CAA J Automat Sin* 1(3):282–293
- Vamvoudakis KG (2015) Non-zero sum Nash Q-learning for unknown deterministic continuous-time linear systems. *Automatica* 61:274–281
- Vamvoudakis KG (2016) Optimal trajectory output tracking control with a Q-learning algorithm. In: 2016 American control conference (ACC), pp 5752–5757
- Vamvoudakis KG (2017a) Q-learning for continuous-time graphical games on large networks with completely unknown linear system dynamics. *Int J Robust Nonlinear Control* 27(16):2900–2920
- Vamvoudakis KG (2017b) Q-learning for continuous-time linear systems: a model-free infinite horizon optimal control approach. *Syst Control Lett* 100:14–20
- Vamvoudakis KG, Ferraz H (2016) Event-triggered h-infinity control for unknown continuous-time linear systems using Q-learning. In: 2016 IEEE 55th conference on decision and control (CDC), pp 1376–1381
- Vamvoudakis KG, Ferraz H (2018) Model-free event-triggered control algorithm for continuous-time linear systems with optimal performance. *Automatica* 87:412–420
- Vamvoudakis KG, Hespanha JP (2018) Cooperative Q-learning for rejection of persistent adversarial inputs in networked linear quadratic systems. *IEEE Trans Autom Control* 63:1018–1031
- Vamvoudakis K, Antsaklis P, Dixon W, Hespanha J, Lewis F, Modares H, Kiumarsi B (2015) Autonomy and machine intelligence in complex systems: a tutorial. In: American control conference (ACC), pp 5062–5079
- Vamvoudakis KG, Mojoyodi A, Ferraz H (2017a) Event-triggered optimal tracking control of nonlinear systems. *Int J Robust Nonlinear Control* 27(4):598–619
- Vamvoudakis KG, Modares H, Kiumarsi B, Lewis FL (2017b) Game theory-based control system algorithms with real-time reinforcement learning: how to solve multiplayer games online. *IEEE Control Syst Mag* 37:33–52
- Watkins C (1989) *Learning from delayed rewards*. Ph.D. thesis, Cambridge University, Cambridge, England
- Yang Y, Vamvoudakis KG, Modares H, Ding D, Yin Y, Wunsch DC (2018a) Dynamic intermittent suboptimal control: performance quantification and comparisons.

In: 2018 37th Chinese control conference (CCC), pp 2017–2022

Yang Y, Modares H, Vamvoudakis KG, Yin Y, Wunsch DC (2018b) Model-free event-triggered containment control of multi-agent systems. In: 2018 annual American control conference (ACC), pp 877–884

Yang Y, Vamvoudakis KG, Ferraz H, Modares H (2019) Dynamic intermittent Q-learning-based model-free suboptimal co-design of π -stabilization. *Int J Robust Nonlinear Control* 29(9):2673–2694

Modeling Hybrid Systems

Michael S. Branicky

Electrical Engineering and Computer Science
Department, University of Kansas, Lawrence,
KS, USA

Abstract

Hybrid systems combine continuous and discrete inputs, outputs, states, and dynamics. Models of hybrid systems combine models of continuous dynamics, such as differential equations, with models of discrete dynamics, such as finite automata. These rich entities can be used to model a wide variety of real-world phenomena. Hybrid models can be viewed as being built up from constituent continuous models with discrete phenomena added or vice versa.

Keywords

Hybrid dynamical systems · Hybrid automata · Cyber-physical systems · Embedded control · Networked control systems · Differential equations · Finite automata

Introduction

Hybrid systems arise in physical systems that make and break contacts (e.g., a bouncing ball) or when there is a logical switching among dynamics (e.g., shifting gears in a car). They arise

whenever one uses computations or finite-state logic to govern continuous physical processes (as in embedded control or cyber-physical systems) or from topological and network constraints interacting with continuous control (as in networked control systems).

Hybrid systems can be used to model systems with relays, switches, and hysteresis (Tavernini 1987), computer disk drives (Gollu and Varaiya 1989), transmissions and stepper motors (Brockett 1993), constrained robotic systems (Back et al. 1993), flight controllers and air traffic management systems (Meyer 1994; Tomlin et al. 1998), multi-vehicle formations and coordination (Tabuada et al. 2001), analog/digital circuits (Maler and Yovine 1996), and biological applications (Ghosh and Tomlin 2004).

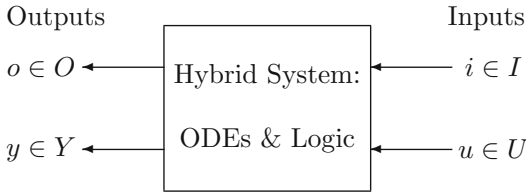
In the case of a hybrid *control* system, models contain both continuous and discrete inputs, and outputs plus internal components themselves may be hybrid; see Fig. 1. There, I and O are discrete (countable) sets of symbols and U and Y are continua.

As an example, consider a flight control system organized around the *modes* Take-Off, Ascend, Cruise, Descend, and Land. A hybrid controller would orchestrate flight among these different regimes, producing flight paths considering air traffic, weather, fuel economy, and passenger comfort.

The rest of this entry provides a reference to the mathematical modeling of hybrid systems, including how they may be built up, step by step, by adding discrete phenomena to continuous dynamical models and by adding continuous phenomena to discrete models. The most commonly used continuous dynamical models are ordinary differential equations (ODEs); the most common discrete models are finite automata.

From Continuous Toward Hybrid

In this section, we start with ordinary differential equations (ODEs) as a base continuous model and then add various discrete phenomenon to them, working toward more and more expressive “hybrid dynamical systems” models. One can



Modeling Hybrid Systems, Fig. 1 Block diagram of a hybrid control system

also use difference equations as the base continuous model (Branicky 1995).

Base Continuous Model: ODEs

Consider a continuous dynamical system defined by *ordinary differential equations (ODEs)*:

$$\dot{x}(t) = f(x(t)), \quad (1)$$

where $x(t) \in X \subset \mathbf{R}^n$. The function $f: X \rightarrow \mathbf{R}^n$ is called a *vector field* on $\mathbf{R}^n$. We assume existence and uniqueness of solutions; see Hirsch and Smale (1974) for conditions.

The system of ODEs in Eq. (1) is called *autonomous* or *time-invariant* because its vector field does not depend explicitly on time. If it did depend explicitly on time, it would be *nonautonomous* or *time-varying*, denoted as follows:

$$\dot{x}(t) = f(x(t), t), \quad (2)$$

An ODE with inputs and outputs (Luenberger 1979; Sontag 1990) is given by

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), \\ y(t) &= h(x(t), u(t)), \end{aligned} \quad (3)$$

where $x(t) \in X \subset \mathbf{R}^n$, $u(t) \in U \subset \mathbf{R}^m$, $y \in Y \subset \mathbf{R}^p$, $f: \mathbf{R}^n \times \mathbf{R}^m \rightarrow \mathbf{R}^n$, and $h: \mathbf{R}^n \rightarrow \mathbf{R}^p$. The functions $u(\cdot)$ and $y(\cdot)$ are the *inputs* and *outputs*, respectively. When inputs are present as in Eq. (3), we say $f(\cdot)$ is a *controlled vector field*.

Differential Inclusions

A *differential inclusion* permits the derivative to belong to a set and is written as:

$$\dot{x}(t) \in F(x(t)),$$

where $F(x(t))$ is a set of vectors in $\mathbf{R}^n$. A differential inclusion can model nondeterminism, including that arising from modeling uncertainty, disturbances, or controls. For example, the controlled differential equation of Eq. (3) can be viewed as an inclusion by setting $F(x) = \bigcup_{u \in U} f(x, u)$.

A *rectangular inclusion* admits derivatives in an interval. For example, an inexact clock with a time-varying rate between 0.9 and 1.1 can be modeled by the rectangular inclusion $\dot{x} \in [0.9, 1.1]$.

Adding Discrete Phenomena

In this section, we discuss the discrete phenomena that arise in hybrid systems:

1. autonomous switching, where the vector field changes discontinuously;
2. autonomous jumps, where the state changes discontinuously;
3. controlled switching, where a control switches vector fields discontinuously;
4. controlled jumps, where a control changes a state discontinuously.

Autonomous Switching

Autonomous switching is the phenomenon where the vector field $f(\cdot)$ changes discontinuously when the continuous state $x(\cdot)$ hits certain “boundaries” (Antsaklis et al. 1993; Nerode and Kohn 1993; Tavernini 1987; Witsenhausen 1966).

Example 1 (Thermostat) The temperature dynamics of a furnace may be quite complicated, depending on outside temperature and humidity; insulation and layout; whether doors are closed, vents are open, or people are present; etc. We abstract its behavior to two modes. When the furnace is **On**, the dynamics are $\dot{x}(t) = f_1(x(t))$, where $x(t)$ is the temperature at time t ; when the furnace is **Off**, the dynamics are governed by $\dot{x}(t) = f_0(x(t))$. This results in a *switched system*:

$$\dot{x}(t) = f_{q(t)}(x(t)),$$

where $q(t) = 0$ or 1 depending on whether the furnace is **Off** or **On**, resp.

A particularly rich class of hybrid systems are *switched linear systems*:

$$\dot{x}(t) = A_{q(t)}x(t), \quad q \in \{1, \dots, N\},$$

where $x(t) \in \mathbf{R}^n$ and each $A_q \in \mathbf{R}^{n \times n}$. Such a system is autonomous if the switching is a function of $x(t)$. The “switching rules” may interact with the constituent dynamics to produce unexpected results, such as instability arising from switching between stable linear systems (Branicky 1998).

Autonomous Jumps

An *autonomous jump* or *autonomous impulse* occurs when the continuous state $x(\cdot)$ jumps discontinuously on hitting prescribed regions of the state space (Back et al. 1993; Bainov and Simeonov 1989). Simple examples of this phenomenon are systems involving collisions.

Example 2 (Bouncing Ball) Consider the vertical position and velocity of a ball of mass m under gravity:

$$\begin{aligned} \dot{x} &= v, \\ \dot{v} &= -mg. \end{aligned}$$

Upon hitting the ground (assuming downward velocity), we instantly set v to $-\rho v$, where $\rho \in [0, 1]$ is the coefficient of restitution. The jump in velocity can be encoded as a rule:

$$\text{“If at time } t, x(t) = 0 \text{ and } v(t) < 0, \text{ then } v(t^+) = -\rho v(t).”$$

In this case, $v(\cdot)$ is piecewise continuous (from the right), with discontinuities occurring when $x = 0$. This “rule” notation is general but cumbersome. It can be replaced with the following notation:

$$v^+(t) = -\rho v(t), \quad (x(t), v(t)) \in \{(0, v) \mid v < 0\}.$$

The equation uses the discrete-time transition notation of Sontag (1990) to denote the “successor” of $x(t)$.

More generally, a system with autonomous impulses may be written as:

$$\begin{aligned} \dot{z}(t) &= f(z(t)), & z(t) &\notin A, \\ z^+(t) &= G(z(t)), & z(t) &\in A. \end{aligned} \quad (4)$$

This system evolves according to the differential equation while z is in the complement of the *autonomous jump set* A ; the state is immediately reset according to the map G when z hits the set A .

Controlled Switching

Controlled switching is the phenomenon where the vector field $f(\cdot)$ changes abruptly in response to a control input. This can be interpreted as switching between different vector fields (Zabczyk 1973). Controlled switching arises when one is allowed to pick among a number of vector fields:

$$\dot{x} = f_q(x), \quad q \in Q \simeq \{1, 2, \dots, N\}.$$

Here, the discrete state q that is active at any given time is chosen by the controller.

Example 3 (Transmission) Consider a simplified model of a manual transmission (Brockett 1993):

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= [-a(x_2/v) + u]/(1 + v), \end{aligned}$$

where x_1 is the ground speed, x_2 is the engine RPM, $u \in [0, 1]$ is the throttle position, and $v \in \{1, 2, 3, 4\}$ is the gear shift position. The function a is positive for positive argument.

Controlled Jumps

A *controlled jump* or *controlled impulse* is the phenomenon where the continuous state $x(\cdot)$ changes discontinuously in response to a control input.

Example 4 (Inventory Management) In a simple inventory management model (Bensoussan and Lions 1984), there are a “discrete” set of restocking times $\theta_1 < \theta_2 < \dots$ and associated order amounts $\alpha_1, \alpha_2, \dots$. The equations governing the

stock at any given moment are:

$$\dot{x}(t) = -\mu(t) + \sum_i \delta(t - \theta_i) \alpha_i,$$

where μ represents continuous degradation or utilization and δ is the Dirac delta.

From Discrete Toward Hybrid

In this section, we start with finite automata as a base discrete model. We then successively add time and continuous dynamics to them, working toward more and more expressive “hybrid automata” models.

Base Discrete Model: Finite Automata

An *input-less automaton* is a dynamical system with a discrete state space:

$$q_{k+1} = v(q_k),$$

where $q_k \in Q$, a finite set.

Augmenting with finite inputs, we arrive at a (*deterministic*) *finite automaton (DFA)* which is a four-tuple $\mathcal{A} = (Q, I, v, q_0)$, where Q is a finite set of *states*; I is a finite set of symbols (known as an *alphabet*), called the *input alphabet*; v is the *transition function* mapping $Q \times I$ into Q ; and $q_0 \in Q$ is the *initial state*.

A DFA begins in state q_0 and then responds to sequences of input symbols (known as *strings* or *words*) over I . In one move, a DFA in state q receives symbol $a \in I$ and enters state $v(q, a)$. On input word $w = a_1 a_2 \cdots a_n$, the DFA in state r_0 successively receives symbols and sequences through states $r_1, r_2, \dots, r_n$, such that

$$r_{k+1} = v(r_k, a_{k+1}); \quad k = 0, 1, 2, \dots, n-1. \quad (5)$$

This sequence of states is a *run* of the DFA over w .

A *nondeterministic finite automaton (NFA)* allows a *set* of start states and a set-valued transition function mapping $Q \times I$ into 2^Q , so that at any stage, the automaton may be in a set of states consistent with its transition function and

the symbols seen so far. In one move, an NFA in state q receives symbol a and nondeterministically enters any one of the states in $v(q, a)$. On input word $w = a_1 a_2 \cdots a_n$, an NFA in state r_0 nondeterministically sequences through states $r_1, r_2, \dots, r_n$ such that

$$r_{k+1} \in v(r_k, a_{k+1}); \quad k = 0, 1, 2, \dots, n-1.$$

Again, we call this sequence a *run* of the NFA over w . In general, NFAs have many runs associated with each string; a DFA only has one.

General Automata

DFAs and NFAs can be augmented with marked states (Hopcroft and Ullman 1979) or process infinite length strings of symbols, known as ω -strings (read “omega strings”) or ω -words over an alphabet I . They can also allow outputs, as in Mealy and Moore machines (Hopcroft and Ullman 1979). Finally, allowing a countable number of states covers models such as pushdown automata and Turing machines (Hopcroft and Ullman 1979) and Petri nets (Cassandras and Lafortune 1999).

A model encompassing these is a *discrete automaton*, which is a five-tuple (Q, I, v, O, η) , consisting of the state space, input alphabet, transition function, output alphabet, and output function, respectively. We assume that Q , I , and O are each countable. When these sets are finite, the result is a finite automaton with output. In any case, the functions involved are $v : Q \times I \rightarrow Q$ and $\eta : Q \times I \rightarrow O$. The *dynamics* of the automaton are specified by

$$\begin{aligned} q_{k+1} &= v(q_k, i_k), \\ o_k &= \eta(q_k, i_k). \end{aligned} \quad (6)$$

Adding Continuous Phenomena

To move toward hybrid systems, we add successively more complex continuous phenomena to finite and discrete automata:

1. *Global Time*, where one adds a single global clock with unit rate.
2. *Timed Automata*, where one adds a set of unit-rate clocks and the ability to reset them,

3. *Skewed-Clock Automata*, where each clock has a different, uniform (in each location), rational rate.
4. *Multi-Rate Automata*, where each clock variable may take on different, rational rates in each location.

For ease of discussion, note that discrete states are also referred to as *modes*, *phases*, or *locations*.

Global Time

Normally, discrete automata are thought of as evolving in “abstract time,” where only the ordering of symbols or “events” matters. See Eq. (6). We may add the notion of time by associating with the k th transition the time, t_k , at which it occurs (Cassandras and Lafortune 1999):

$$\begin{aligned} q(t_{k+1}) &= v(q(t_k), i(t_k)), \\ o(t_k) &= \eta(q(t_k), i(t_k)). \end{aligned}$$

This automaton may be thought of as operating in “continuous time,” where the state, input, and output symbols are piecewise continuous functions. Again, using the transition notation of Sontag (1990), we obtain:

$$\begin{aligned} q^+(t) &= v(q(t), i(t)), \\ o(t) &= \eta(q(t), i(t)). \end{aligned} \quad (7)$$

Here, the state q and output o are piecewise continuous functions that only change when the input symbol $i(t)$ changes.

Timed Automata

Timed automata (Alur and Dill 1994) process *timed* words. A *timed word* (over input alphabet I) is a finite sequence of symbols and their respective times of occurrence: $(i_1, t_1), (i_2, t_2), \dots, (i_N, t_N)$, where each $i_k \in I$, each $t_k \in \mathbf{R}_+$, and times *strictly increase*: $t_{k+1} > t_k$. A *timed ω -word*, $(i_1, t_1), (i_2, t_2), \dots$, adds the constraint that time *progresses* without bound: for all $T \in \mathbf{R}_+$, there exists a k such that $t_k > T$. The latter condition is necessary to avoid so-called *Zeno behavior*. For example, the sequence

of times $1/2, 3/4, 7/8, 15/16, 31/32, \dots$ is not a valid time sequence.

A *timed automaton* is a DFA plus a finite number of real-valued clocks, and the ability to reset clocks and test constraints on clocks when traversing edges. Each *clock constraint*, χ , may take one of the forms:

$$(x \leq c), \quad (c \leq x), \quad \neg\chi_0, \quad \chi_1 \wedge \chi_2,$$

where x is a clock variable, c is a rational constant, $\neg$ denotes logical negation, $\wedge$ denotes logical “and,” and χ_i are themselves valid clock constraints. These primitives plus the rules of logic allow one to build up more complicated tests, including $(x = c)$, $(x < c)$, $\chi_1 \vee \chi_2$, and **True**.

In drawings of timed automata, the notation $? \chi$ is used to denote the fact that the edge *may* be taken only if the clock constraint χ is true. We have found it useful to also add $! \chi$, meaning that the edge *must* be taken when χ is true.

Many decision and verification problems for timed automata may be translated directly into questions about (generally much larger) finite-state automata (Alur and Dill 1994).

Skewed-Clock Automata

Summarizing, a timed automaton has a finite set, C , of clocks, $x_i \in C$ such that $\dot{x}_i = 1$ for all clocks and all locations. A *skewed-clock automaton* allows $\dot{x}_i = k_i$ where each k_i is a rational number.

Remark 1 Skewed-clock automata are equivalent to timed automata. That is, any skewed-clock automaton can be converted into an equivalent timed automaton and vice versa (by replacing clock i with a unit-rate clock and dividing any constant to which x_i is compared by k_i).

Multi-rate Automata

A *multi-rate automaton* allows $\dot{x}_i = k_{i,q}$ at location $q \in Q$, where each $k_{i,q}$ is a rational number; see Fig. 2. Note that for multi-rate automata:

- Some variables might have the same rates in all states, e.g., w in Fig. 2.

- Some variables might be *stopwatches* (derivative either 0 or 1) measuring a particular time. E.g., x in Fig. 2 is a stopwatch measuring the elapsed time spent in the upper left state.
- Not all dynamics change at every transition. E.g., y in Fig. 2 changes dynamics in transitioning along Edges 2 and 3 but not along Edge 1.
- Every skewed-clock automaton is a multi-rate automaton that simply has $k_{i,q} = k_i$ for all $q \in Q$.

Multi-rate automata may exhibit *Zeno behavior* (Zhang et al. 2000). Their behavior may even be computationally undecidable (Henzinger et al. 1998). However, a simple constraint on multi-rate automata yields a class that permits analysis: *initialized* multi-rate automata add the constraint that a variable must be reset when traversing an edge if its dynamics changes while crossing that edge. This would be the case for the automaton in Fig. 2 if x and z were initialized on Edge 1, y and z were initialized on Edge 2, and x , y , and z were initialized on Edge 3.

Remark 2 An initialized multi-rate automaton can be converted into a timed automaton (Henzinger et al. 1998).

Other Refinements

Rectangular Automata

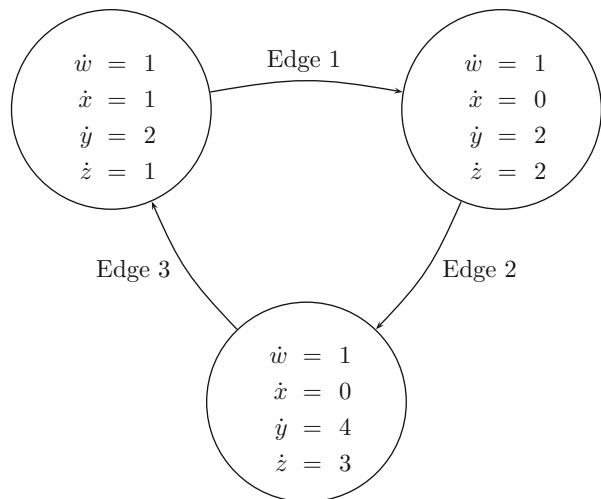
Progressing from rates to rectangular inclusions permits rectangular and multi-rectangular automata. A *rectangular automaton* has each $\dot{x}_i \in [l_i, u_i]$, where l_i and u_i are rational constants, uniformly over all locations. Therefore, they generalize skewed-clock automata. Indeed, setting $l_i = u_i = k_i$ recovers them. Similarly, multi-rate automata can be generalized to *multi-rectangular automata*. Here, each variable satisfies a (generally different) rectangular inclusion in each location. Rectangular automata also allow the setting of initial values and the resetting of variables to be performed within rectangles. *Initialized* multi-rectangular automata must reset a variable along any edge that changes the rectangular inclusion governing its dynamics.

Remark 3 An initialized multi-rectangular automaton can be converted to an initialized multi-rate automaton and hence a timed automaton (Henzinger et al. 1998).

Adding Control

A rich control theory for general automata models can be built by allowing the disabling of certain *controllable* input symbols. See Ramadge and Wonham (1989) and Cassandras and Lafor-
tune (1999).

Modeling Hybrid Systems, Fig. 2 Example multi-rate automaton (edge data suppressed)



Hybrid Dynamical Systems and Hybrid Automata

A hybrid dynamical system is an indexed collection of ODEs along with maps for “jumping” among them (switching dynamical system and/or resetting the state). The jumping occurs whenever the state satisfies certain conditions, given by its membership in a specified subset of the state space. The entire system can be thought of as a sequential patching together of ODEs with initial and final states, with jumps performing a reset to a (generally different) initial state of a (generally different) ODE whenever a final state is reached; see Fig. 3.

More formally, a *hybrid dynamical system* (HDS) is a four-tuple $H = (Q, \Sigma, A, G)$:

- Q is a countable set of *index* or *discrete states*.
- $\Sigma = \{\Sigma_q\}_{q \in Q}$ is a collection of *systems* of ODEs. Thus, each system Σ_q has vector fields $f_q : X_q \rightarrow \mathbf{R}^{d_q}$, representing *continuous dynamics*, evolving on $X_q \subset \mathbf{R}^{d_q}$, $d_q \in \mathbf{Z}_+$, which are *continuous state spaces*.
- $A = \{A_q\}_{q \in Q}$, $A_q \subset X_q$ for each $q \in Q$, is a collection of *autonomous jump sets*.

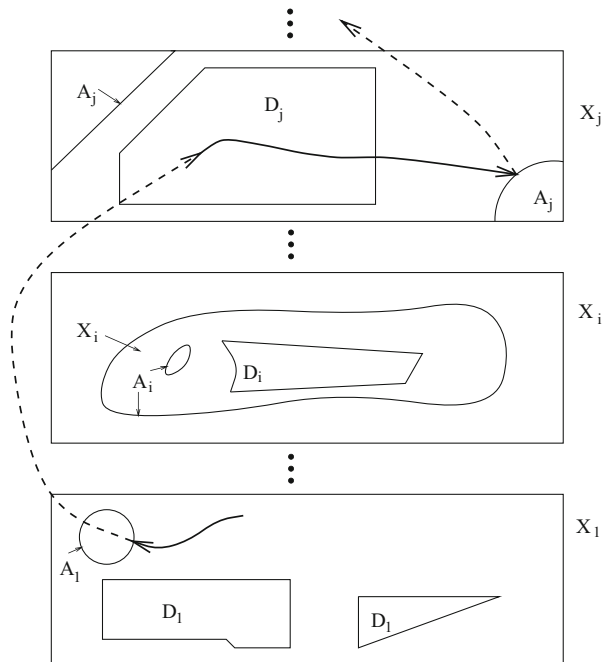
- $G = \{G_q\}_{q \in Q}$, where $G_q : A_q \rightarrow S$, are *autonomous jump transition maps* that represent the *discrete dynamics*.

Thus, $S = \bigcup_{q \in Q} X_q \times \{q\}$ is the *hybrid state space* of H . For convenience, the following shorthand is used: $S_q = X_q \times \{q\}$, and $A = \bigcup_{q \in Q} A_q \times \{q\}$ is the *autonomous jump set*; $G : A \rightarrow S$ is the *autonomous jump transition map*, constructed component-wise. The *jump destination sets* $D = \{D_q\}_{q \in Q}$ are defined by $D_q = \pi_1[G(A) \cap S_q]$, where π_i is projection onto the i th coordinate. The *switching manifolds* or *transition manifolds*, $M_{q,p} \subset A_q$, are given by $M_{q,p} = G_q^{-1}(D_p, p)$, i.e., the set of states for which transitions from index q to index p can occur.

The dynamics of HDS H are as follows. The system starts in some hybrid state in $S \setminus A$, say $s_0 = (x_0, q_0)$. It evolves according to $\dot{x} = f_{q_0}(x)$ until the state enters – if ever – A_{q_0} at the point $s_1^- = (x_1^-, q_0)$. At this time, it is instantly transferred according to the transition map to $G_{q_0}(x_1^-) = (x_1, q_1) \equiv s_1$, from which the process continues.

M

Modeling Hybrid Systems, Fig. 3 Hybrid dynamical system



We now collect some notes about HDS:

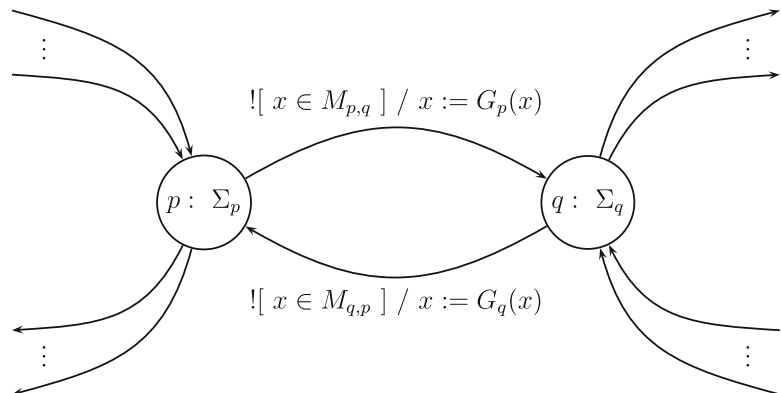
- *ODEs and automata.* In the case $|Q| = 1$ and $A = \emptyset$, we recover all differential equations. With $|Q| = N$ and each $f_q \equiv 0$, we recover finite automata.
- *Outputs.* It is straightforward to add continuous and discrete output maps for each constituent system.
- *Changing state space and overlaps.* The state space and its dimension may change, which is useful in modeling component failures or changes in dynamical description based on autonomous – or later, controlled – events which change it. Examples include inelastic collisions or aircraft mode transitions that change the variables to be controlled (Meyer 1994). The model allows the X_q to overlap and the inclusion of multiple copies of the same space. This can be used to model overlapping local coordinate systems (Back et al. 1993) or hysteresis (Branicky 1995).
- *Transition Delays.* One can model situations where the autonomous jumps are not instantaneous by adding an *autonomous jump delay map*, $\Delta_a : A \times V \rightarrow \mathbf{R}_+$. Such a map associates a (possibly zero) delay to each autonomous jump. Thus, a jump begins at some time, τ , and is completed at some later time $\Gamma \geq \tau$. This concept is useful as jumps may be introduced to represent an aggregate set of (relatively) fast, transitory dynamics. Also, some commands may have a finite delay from issuance to completion, for example, closing a hydraulic valve.
- *Well-Defined Solutions.* For the solutions described to be well-defined, it is sufficient that for all q , A_q is closed, $D_q \cap A_q = \emptyset$, and f_q is Lipschitz continuous. Note, however, that in general these solutions are not continuous in the initial conditions. See Branicky (1995) for details.
- *Generalizations.* A more general notion, *GHDS* (*General HDS*), is due to Branicky (1995). It allows each ODE to be replaced by a general dynamical system. Most notably one could replace the ODEs with difference equations.

HDS with $|Q|$ finite are a coupling of finite automata and differential equations. Indeed, an HDS can be pictured as a *hybrid automaton* as in Fig. 4. Therein, each node is a constituent ODE, with the index the name of the node. Each edge represents a possible transition between constituent systems, labeled by the appropriate condition for the transition's being “enabled” and the update of the continuous state; cf. Harel (1987). The notation $![\text{condition}]$ denotes that the transition *must* be taken when enabled.

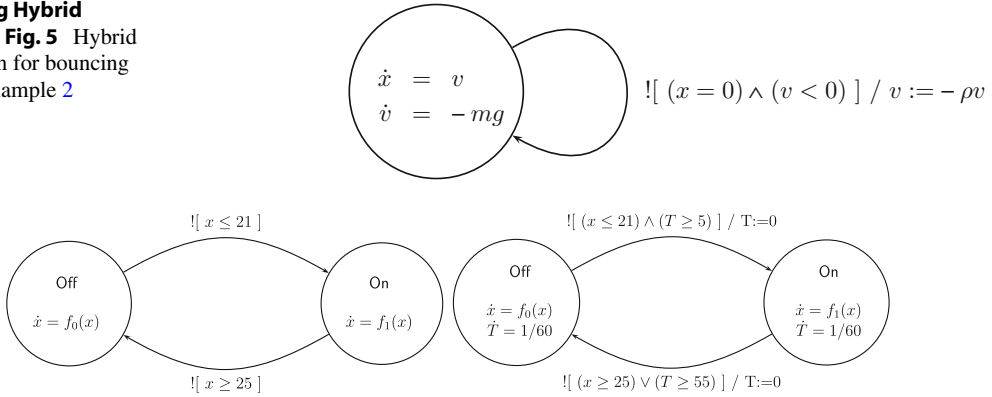
Example 5 (Bouncing Ball Revisited) The hybrid automaton associated with the bouncing ball of Example 2 appears in Fig. 5. The velocity resets are autonomous and hence must occur when the ball hits the ground.

Example 6 (HVAC Revisited) Reconsider the furnace controller of Example 1, adding the goal of keeping the temperature about 23 degrees Celsius. To avoid inordinate switching around the

Modeling Hybrid Systems, Fig. 4 Hybrid automaton representation of an HDS



Modeling Hybrid Systems, Fig. 5 Hybrid automaton for bouncing ball of Example 2



Modeling Hybrid Systems, Fig. 6 Hybrid control systems for a furnace: (l) prototypical hysteresis, (r) adding timing constraints

set point (known as *chattering*), we implement a control with hysteresis as in Fig. 6(l). This controller will (1) turn the furnace **On** when it is **Off** and the temperature falls below 21 and (2) turn the furnace **Off** when it is **On** and the temperature rises above 25. Suppose, in addition, that the furnace should never be on more than 55 min straight and must remain **Off** for at least 5 min. These objectives can be accomplished by adding a timer variable that is reset on state transitions; see Fig. 6(r).

Adding Control

One may add the ability to control an HDS by choosing among sets of possible actions at various times. Specifically, we allow (1) the possibility of continuous controls for each ODE, (2) the ability to make decisions at autonomous jump points, and (3) the ability to make controlled jumps, where one may decide to jump or not and have a choice of destination when the state satisfies certain constraints.

Formally, a *controlled hybrid dynamical system* (CHDS) is a six-tuple $H_c = (Q, \Sigma, A, G, C, F)$:

- $\Sigma = \{\Sigma_q\}_{q \in Q}$ is now a collection of *controlled* ODEs, with controlled vector fields $f_q : X_q \times U_q \rightarrow \mathbf{R}^{d_q}$ representing (controlled) *continuous dynamics*, where $U_q \subset \mathbf{R}^{m_q}$ and $m_q \in \mathbf{Z}_+$ are *continuous control spaces*.

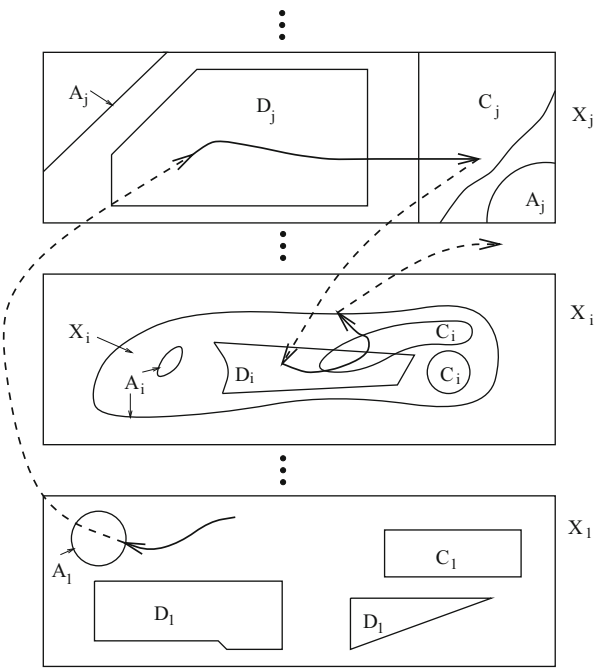
- $G = \{G_q\}_{q \in Q}$, where $G_q : A_q \times V_q \rightarrow S$ are *autonomous jump transition maps*, now modulated by *discrete decision sets* V_q .
- $C = \{C_q\}_{q \in Q}$, $C_q \subset X_q$, is a collection of *controlled jump sets*.
- $F = \{F_q\}_{q \in Q}$, where $F_q : C_q \rightarrow 2^S$, is a collection of set-valued *controlled jump destination maps*.

Again, $S = \bigcup_{q \in Q} X_q \times \{q\}$ is the hybrid state space of H_c . As before, we use some shorthand beyond that defined for HDS above. We let $C = \bigcup_{q \in Q} C_q \times \{q\}$; we let $F : C \rightarrow 2^S$ denote the set-valued map composed in the obvious way from the set-valued maps F_q . Let $\tilde{G}_q = \{G(x, v) | v \in V_q\}$

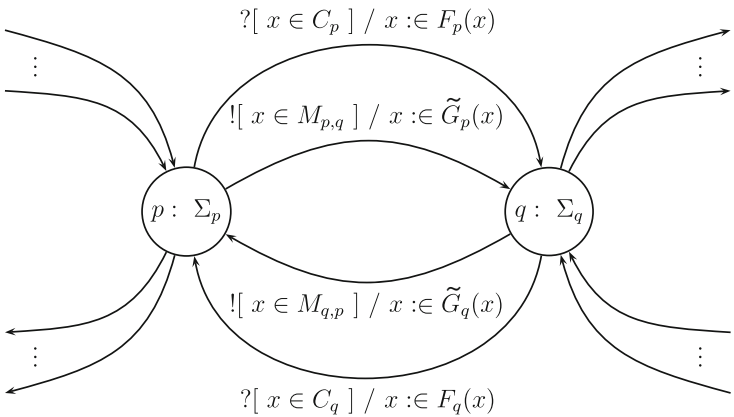
The dynamics of the CHDS H_c are as follows. The system is assumed to start in some hybrid state in $S \setminus A$, say $s_0 = (x_0, q_0)$. It evolves according to $\dot{x} = f_{q_0}(x, u)$ until the state enters – if ever – either A_{q_0} or C_{q_0} at the point $s_1^- = (x_1^-, q_0)$. If it enters A_{q_0} , then it *must* be transferred according to transition map $G_{q_0}(x_1^-, v)$ for some chosen $v \in V_{q_0}$. If it enters C_{q_0} , then we *may* choose to jump, and, if so, we may choose the destination to be any point in $F_{q_0}(x_1^-)$. In either case, we arrive at a point $s_1 = (x_1, q_1)$ from which the process continues; see Fig. 7.

A CHDS can also be pictured as an automaton, as in Fig. 8. There, each node is a constituent controlled ODE, with the index the name of the node. Again, the notation $?[\text{condition}]$ denotes

Modeling Hybrid Systems, Fig. 7
Controlled hybrid dynamical system



Modeling Hybrid Systems, Fig. 8 Hybrid automaton representation of a CHDS



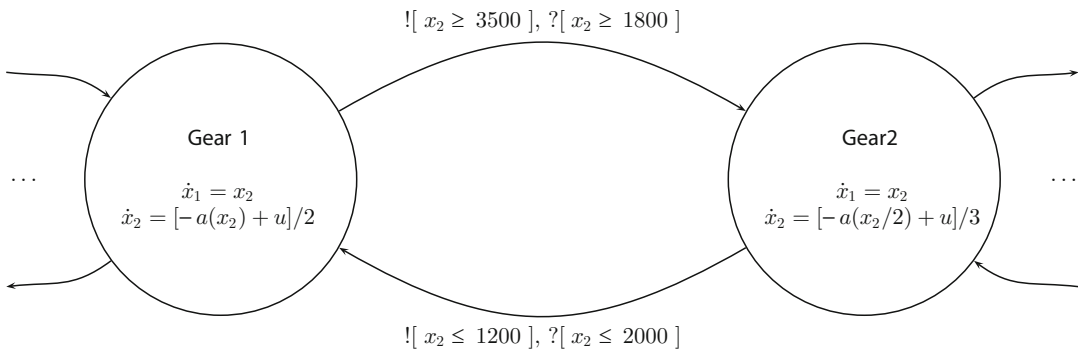
an enabled transition that *may* be taken on command; “ $\in$ ” means reassignment to some value in the given set.

Example 7 (Transmission Revisited) Some modern performance sedans offer the driver the ability to shift gears electronically. However, there are often rules in place that prevent certain shifts, or automatically perform certain shifts, to maintain safe operation and reduce wear. These rules are often a function of the engine RPM (x_2 in Example 3). A portion of a hybrid controller incorporating such logic is shown in Fig. 9. The rules pictured only allow the driver to shift from

Gear 1 to Gear 2 if RPM exceeds 1800 but automatically makes this switch if RPM exceeds 3500. Similar rules would exist for higher gears; more complicated rules might exist regarding **Neutral** and **Reverse**.

Summary and Future Directions

This entry summarized quite general models for hybrid systems, namely, hybrid dynamical systems and hybrid automata. It built these up successively from base models of completely



Modeling Hybrid Systems, Fig. 9 Portion of a hybrid control system for the transmission of Example 3

continuous and completely discrete models: ODEs and finite automata. Future directions largely revolve around using these to model new applications, such as autonomous cars and drones. Another direction involves identifying classes that are amenable to (automated) design, analysis, and verification.

Cross-References

- [Discrete Event Systems and Hybrid Systems, Connections Between](#)
- [Modeling, Analysis, and Control with Petri Nets](#)

Recommended Reading

For more details on hybrid systems modeling, see Alur et al. (1993), Branicky (1995), Henzinger et al. (1998), and Goebel et al. (2009). For equivalence of different models, see Branicky (1995) and Heemels et al. (2001). For more on modeling embedded systems, see Lee and Seshia (2016). For more on modeling networked control systems, see Zhang et al. (2001). For switching systems, see Liberzon (2003). For stochastic hybrid systems, see Cassandras and Lygeros (2006).

Bibliography

Alur R, Dill DL (1994) A theory of timed automata. *Theor Comp Sci* 126:183–235

- Alur R, Courcoubetis C, Henzinger TA, Ho P-H (1993) Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In: Grossman R, Nerode A, Ravn A, Rischel H (eds) *Hybrid systems*. Springer, New York, pp 209–229
- Antsaklis PJ, Stiver JA, Lemmon MD (1993) Hybrid system modeling and autonomous control systems. In: Grossman R, Nerode A, Ravn A, Rischel H (eds) *Hybrid systems*. Springer, New York, pp 366–392
- Back A, Guckenheimer J, Myers M (1993) A dynamical simulation facility for hybrid systems. In: Grossman R, Nerode A, Ravn A, Rischel H (eds) *Hybrid systems*. Springer, New York, pp 255–267
- Bainov DD, Simeonov PS (1989) *Systems with impulse effect*. Ellis Horwood, Chichester
- Bensoussan A, Lions J-L (1984) *Impulse control and quasi-variational inequalities*. Gauthier-Villars, Paris
- Branicky MS (1995) *Studies in hybrid systems: modeling, analysis, and control*. ScD thesis, M.I.T., Cambridge, MA
- Branicky MS (1998) Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans Autom Control* 43(4):475–482
- Branicky MS, Borkar VS, Mitter SK (1998) A unified framework for hybrid control: model and optimal control theory. *IEEE Trans Autom Control* 43(1):31–45
- Brockett RW (1993) Hybrid models for motion control systems. In: Trentelman HL, Willems JC (eds) *Essays in control*. Birkhäuser, Boston, pp 29–53
- Cassandras CG, Lafortune S (1999) *Introduction to discrete event systems*. Kluwer Academic Publishers, Boston
- Cassandras CG, Lygeros J (2006) *Stochastic hybrid systems*. CRC Press
- Ghosh R, Tomlin C (2004) Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: delta-notch protein signalling. *Syst Biol* 1(1):170–183
- Goebel R, Sanfelice RG, Teel AR (2009) Hybrid dynamical systems. *IEEE Control Syst Mag* 29(2):28–93
- Gollu A, Varaiya PP (1989) Hybrid dynamical systems. In: *Proceedings of IEEE conference on decision control*, Tampa, pp 2708–2712

- Harel D (1987) Statecharts: a visual formalism for complex systems. *Sci Comp Prog* 8:231–274
- Heemels WP, De Schutter B, Bemporad A (2001) Equivalence of hybrid dynamical models. *Automatica* 37(7):1085–1091
- Henzinger TA, Kopke PW, Puri A, Varaiya P (1998) What's decidable about hybrid automata? *J Comput Syst Sci* 57:94–124
- Hirsch MW, Smale S (1974) *Differential equations, dynamical systems, and linear algebra*. Academic, San Diego
- Hopcroft JE, Ullman JD (1979) *Introduction to automata theory, languages, and computation*. Addison-Wesley, Reading
- Lee EA, Seshia SA (2016) *Introduction to embedded systems: a cyber-physical systems approach*. MIT Press, Cambridge
- Liberzon D (2003) *Switching in systems and control*. Birkhauser, Boston
- Luenberger DG (1979) *Introduction to dynamic systems*. Wiley, New York
- Maler O, Yovine S (1996) Hardware timing verification using KRONOS. In: *Proceedings of 7th Israeli conference on computer systems and software engineering*. Herzliya, Israel
- Meyer G (1994) Design of flight vehicle management systems. In: *IEEE conference on decision control, plenary lecture*, Lake Buena Vista
- Nerode A, Kohn W (1993) Models for hybrid systems: automata, topologies, controllability, observability. In: Grossman R, Nerode A, Ravn A, Rischel H (eds) *Hybrid systems*. Springer, New York, pp 317–356
- Ramadge PJG, Wonham MW (1989) The control of discrete event systems. *Proc IEEE* 77(1):81–98
- Sontag ED (1990) *Mathematical control theory*. Springer, New York
- Tabuada P, Pappas GJ, Lima P (2001) Feasible formations of multi-agent systems. In: *Proceedings of American control conference*, Arlington
- Tavernini L (1987) Differential automata and their discrete simulators. *Nonlinear Anal Theory Methods Appl* 11(6):665–683
- Tomlin C, Pappas GJ, Sastry S (1998) Conflict resolution for air traffic management: a study in multi-agent hybrid systems. *IEEE Trans Autom Control* 43(4):509–521
- Witsenhausen HS (1966) A class of hybrid-state continuous-time dynamic systems. *IEEE Trans Autom Control AC-11*(2):161–167
- Zabczyk J (1973) Optimal control by means of switching. *Studia Mathematica* 65:161–171
- Zhang J, Johansson KH, Lygeros J, Sastry S (2000) Dynamical systems revisited: hybrid systems with Zeno executions. In: Lynch N, Krogh B (eds) *Hybrid systems: computation and control*. Springer, Berlin, pp 451–464
- Zhang W, Branicky MS, Phillips SM (2001) Stability of networked control systems. *IEEE Control Syst Mag* 21(1):84–99

Modeling of Dynamic Systems from First Principles

S. Torkel Glad

Department of Electrical Engineering,
Linköping University, Linköping, Sweden

Abstract

This paper describes how models can be formed from the basic principles of physics and the other fields of science. Use can be made of similarities between different domains which leads to the concepts of bond graphs and, more abstractly, to port-controlled Hamiltonian systems. The class of models is naturally extended to differential algebraic equations (DAE) models. The concepts described here form a natural basis for parameter identification in gray box models.

Keywords

Physical modeling · Grey box model · DAE

Introduction

The approach to the modeling of dynamic systems depends on how much is known about the system. When the internal mechanisms are known, it is natural to model them using known relationships from physics, chemistry, biology, etc. Often the result is a model of the following form

$$\frac{dx}{dt} = f(x, u; \theta), \quad y = h(x, u; \theta) \quad (1)$$

where u is the input and y the output, and the state x contains internal physical variables, while θ contains parameters. Typically all of these are vectors. The model is known as a state-space model. In many cases some elements in θ are unknown and have to be determined using parameter estimation. When used in connection with system identification, these models are some-

times referred to as *gray box* models (in contrast to black box models) to indicate that some degree of physical knowledge is assumed.

Overview of Physical Modeling

Since modeling covers such a wide variety of physical systems, there are no universal systematic principles. However a few concepts have wide application. One of them is the preservation of certain quantities like energy, leading to *balance equations*. A simple example is given by the heating of a body. If W is the energy stored as heat, P_1 an external power input, and P_2 the heat loss to the environment per time unit, energy balance gives

$$\frac{dW}{dt} = P_1 - P_2 \quad (2)$$

To get a complete model, one needs also *constitutive relations*, i.e., relations between relevant physical variables. For instance, one might know that the stored energy is proportional to the temperature T , $W = CT$ and that the energy loss is from black body radiation, $P_2 = kT^4$. The model is then

$$C \frac{dT}{dt} = P_1 - kT^4 \quad (3)$$

The model is now an ordinary differential equation with state variable T , input variable P_1 , and parameters C and k .

Physical Analogies and General structures

Physical Analogies

Physicists and engineers have noted that modeling in different areas of physics often give very similar models. The term “analogies” is often used in modeling to describe this fact. Here we will show some analogies between electrical and mechanical phenomena. Consider the electric circuit given in Fig. 1. An ideal voltage source is connected in series with an inductor, a resistor, and a capacitor. Using u and v to denote the

voltages over the voltage source and capacitor, respectively, and i to denote the current, a mathematical model is

$$\begin{aligned} C \frac{dv}{dt} &= i \\ L \frac{di}{dt} + Ri + v &= u \end{aligned} \quad (4)$$

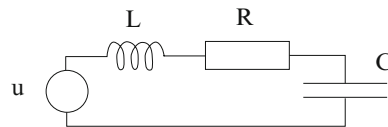
The first equation uses the definition of capacitance, and the second one uses Kirchhoff's voltage law. Compare this to the mechanical system of Fig. 2 where an external force F is applied to a mass m that is also connected to a damper b and a spring with spring constant k . If S is the elongation force of the spring and w the velocity of the mass, a system model is

$$\begin{aligned} \frac{dS}{dt} &= kw \\ m \frac{dw}{dt} + bw + S &= F \end{aligned} \quad (5)$$

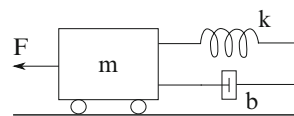
Here the first equation uses the definition of spring constant, and the second one uses Newton's second law. The models are seen to be the same with the following correspondences between time varying quantities

$$u \leftrightarrow F, \quad i \leftrightarrow w, \quad v \leftrightarrow S \quad (6)$$

and between parameters



Modeling of Dynamic Systems from First Principles, Fig. 1 Electric circuit



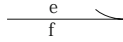
Modeling of Dynamic Systems from First Principles, Fig. 2 Mechanical system

$$C \leftrightarrow 1/k, \quad L \leftrightarrow m, \quad R \leftrightarrow b \quad (7)$$

Note that the products (voltage) $\times$ (current) and (force) $\times$ (velocity) give the power.

Bond Graphs

The bond graph is a tool to do systematic modeling based on the analogies of the previous section. The basic element is the bond



formed by a half-arrow showing the direction of positive energy flow. Two variables are associated with the bond, the *effort* variable e and the *flow* variable f . The product ef of these variables gives the power. In the electric domain, e is voltage, and f is current. For mechanical systems e is force, while f is velocity. Bond graph theory has three basic components to describe storage and dissipation of energy. The relations

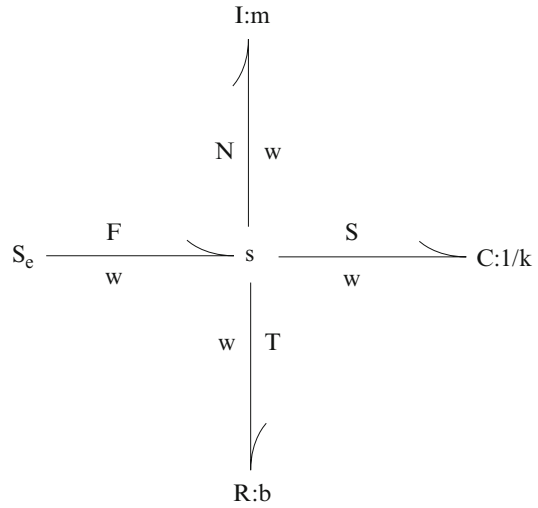
$$\alpha \frac{de}{dt} = f, \quad \beta \frac{df}{dt} = e, \quad \gamma f = e \quad (8)$$

are known as C, I, and R elements, respectively. Input signals are modeled by elements called effort sources S_e or flow sources S_f , respectively. A bond graph describes the energy flow between these elements. When the energy flow is split, it can either be at s junctions where the flows are equal and the efforts are added or at a p junction where efforts are equal and flows are additive. The model (5), for instance, can be described by the bond graph in Fig. 3. The graph shows how the energy from the external force is split into the acceleration of the mass, the elongation of the spring, and dissipation into the damper. The splitting of the energy flow is accomplished by an s element, meaning that the velocity is the same for all elements but that the forces are added

$$F = N + S + T \quad (9)$$

From (8) it follows that

$$k^{-1} \frac{dS}{dt} = w, \quad m \frac{dw}{dt} = N, \quad bw = T \quad (10)$$



Modeling of Dynamic Systems from First Principles, Fig. 3 Bond graph for mechanical or electric system

Together (9) and (10) give the same model as (5). Using the correspondences (6) and (7), it is seen that the same bond graph can also represent the electric circuit (4). An overview of bond graph modeling is given in Rosenberg and Karnopp (1983). A general overview of modeling, including bond graphs and the connection with identification, can be found in Ljung and Glad (2016).

Port-Controlled Hamiltonian Systems

Another way of viewing the mechanical system would be to introduce x_1 as the length of the spring so that $dx_1/dt = w$. If H_1 is the energy stored in the spring, then the following relations hold.

$$H_1(x_1) = \frac{kx_1^2}{2}, \quad \frac{\partial H_1}{\partial x_1} = kx_1 = S \quad (11)$$

Introducing x_2 for the momentum and H_2 as the kinetic energy, one has $dx_2/dt = N$ and

$$H_2(x_2) = \frac{x_2^2}{2m}, \quad \frac{\partial H_2}{\partial x_2} = m^{-1}x_2 = w \quad (12)$$

Let $H = H_1 + H_2$ be the total energy. Then the following relation holds

$$\frac{dx}{dt} = \left(\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & b \end{bmatrix} \right) \begin{bmatrix} \partial H / \partial x_1 \\ \partial H / \partial x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} F \quad (13)$$

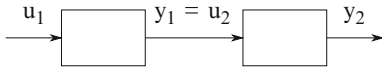
This model is a special case of

$$\frac{dx}{dt} = (M - R)\nabla H(x) + Bu \quad (14)$$

where M is a skew symmetric and R a nonnegative definite matrix, respectively. The model is a port-controlled Hamiltonian system with dissipation. Without external input ($B = 0$) and dissipation ($R = 0$), it reduces to an ordinary Hamiltonian system. For systems generated by simple bond graphs, it can be shown that the junction structure gives the skew symmetric M , while the R elements give the matrix R . The storage of energy in I and C elements is reflected in H . The reader is directed to Duindam et al. (2009) for a description of the port-Hamiltonian approach to modeling.

Component-Based Models and Modeling Languages

Since engineering systems are usually assembled from components, it is natural to treat their mathematical models in the same way. This is the idea behind block-oriented models where the output of one model is connected to the input of another one:



A nice feature of this block connection is that the state space description is preserved. Suppose the individual models are of the form (1)

$$\frac{dx_i}{dt} = f_i(x_i, u_i), \quad y_i = h_i(x_i, u_i), \quad i = 1, 2 \quad (15)$$

Then the connection $u_2 = y_1$ immediately gives the state space model

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} f_1(x_1, u_1) \\ f_2(x_2, h_1(x_1, u_1)) \end{bmatrix}, \quad y_2 = h_2(x_2, h_1(x_1, u_1)) \quad (16)$$

with input u_1 , output y_2 , and state (x_1, x_2) . This fact is the basis of block-oriented modeling and simulation tools like the MATLAB-based Simulink. Unfortunately the preservation of the state space structure does not extend to more general connections of systems. Consider, for instance, two pieces of rotating machinery described by

$$J_i \frac{d\omega_i}{dt} = -b_i \omega_i + M_i, \quad i = 1, 2 \quad (17)$$

where ω_i is the angular velocity, M_i the external torque, J_i the moment of inertia, and b_i the damping coefficient. Suppose the pieces are joined together so that they rotate with the same angular velocity. The mathematical model would then be

$$\begin{aligned} J_1 \frac{d\omega_1}{dt} &= -b_1 \omega_1 + M_1 \\ J_2 \frac{d\omega_2}{dt} &= -b_2 \omega_2 + M_2 \\ \omega_1 &= \omega_2 \\ M_1 &= -M_2 \end{aligned} \quad (18)$$

This is no longer a state space model of the form (1) but a mixture of dynamic and static relationships, usually referred to as a differential algebraic equation (DAE). The difference from the connection of blocks in block diagrams is that now the connection is not between an input and an output. Instead there are the equations $\omega_1 = \omega_2$ and $M_1 = -M_2$ that destroy the state space structure. There exist modeling languages like Modelica, Tiller (2012) and Fritzson (2015), or Simscape in MATLAB, MathWorks (2019) that accept this more general type of model. It is then possible to form model libraries of basic components that can be interconnected in very general ways to form models of complex systems. However this more general structure poses some challenges when it comes to analysis and simulation that are described in the next section.

Differential Algebraic Equations (DAE)

This model (18) is a special case of the general differential algebraic equation

$$F(dz/dt, z, u) = 0 \quad (19)$$

A good description of both theory and numerical properties of such equations is given in Kunkel and Mehrmann (2006). In many cases it is possible to split the variables and equations in such a way that the following structure is achieved.

$$F_1(dz_1/dt, z_1, z_2, u) = 0, \quad F_2(z_1, z_2, u) = 0 \quad (20)$$

If z_2 can be solved from the second equation and substituted into the first one and if dz_1/dt can then be solved from the first equation, the problem is reduced to an ordinary differential equation in z_1 . Often, however, the situation is not as simple as that. For the example (18), an addition of the first two equations gives

$$(J_1 + J_2) \frac{d\omega_1}{dt} = -(b_1 + b_2)\omega_1 \quad (21)$$

which is a standard first-order system description. Note, however, that in order to arrive at this result, the relation $\omega_1 = \omega_2$ has to be differentiated. This DAE thus includes an implicit differentiation. In the general case, one can investigate how many times (19) has to be differentiated in order to get an explicit expression for dz/dt . This number is called the (differentiation) *index*. Both theoretical analysis and practical experience show that the numerical difficulties encountered when solving a DAE increase with increasing index, see, e.g., the classical reference Brenan et al. (1987) or Campbell et al. (2019). It turns out that mechanical systems in particular give high index models when constructed by joining components, and this has been an obstacle to the use of DAE models. For linear DAE models, the role of the index can be seen more easily. A linear model is given by

$$E \frac{dz}{dt} + Fz = Gu \quad (22)$$

where the matrix E is singular (if E is invertible, multiplication with E^{-1} from the left gives an ordinary differential equation). The system can be transformed by multiplying with P from the left and changing variables with $z = Qw$ (P, Q nonsingular matrices). The transformed model is now

$$PEQ \frac{dw}{dt} + PFQw = PGu \quad (23)$$

If $\lambda E + F$ is nonsingular for some value of the scalar λ , then it can be shown that there is a choice of P and Q such that (23) takes the form

$$\begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} dw_1/dt \\ dw_2/dt \end{bmatrix} + \begin{bmatrix} -A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \quad (24)$$

where N is a nilpotent matrix, i.e., $N^k = 0$ for some positive integer k . The smallest such as k turns out to be the index of the DAE. The transformed model (24) thus contains an ordinary differential equation

$$\frac{dw_1}{dt} = Aw_1 + B_1u \quad (25)$$

Using the nilpotency of N , the equation for w_2 can be rewritten

$$w_2 = B_2u - NB_2 \frac{du}{dt} + \dots + (-N)^{k-1} B_2 \frac{d^{k-1}u}{dt^{k-1}} \quad (26)$$

This expression shows that an index $k > 1$ implies differentiation of the input (unless NB_2 happens to be zero). This in turn implies potential difficulties, e.g., if u is a measured signal.

Identification of DAE Models

The extended use of DAE models in modern modeling tools also means that there is a need to use these models in system identification. To fully use system identification theory, one needs a stochastic model of disturbances. The inclusion of such disturbances leads to a class of models described as stochastic differential algebraic equations. The treatment of such models leads to some interesting problems. In the previous section, it was seen that DAE models often contain implicit differentiations of external signals.

If a DAE model is to be well posed, this differentiation must not affect signals modeled as white noise. In Gerdin et al. (2007), conditions are given that guarantee that stochastic DAEs are well posed. There it is also described how a maximum likelihood estimate can be made for DAE models, laying the basis for parameter estimation.

Differential Algebra

For the case where models consist of polynomial equations, it is possible to manipulate them in a very systematic way. The model (19) is then generalized to

$$F(d^n z/dt^n, \dots, dz/dt, z) = 0 \quad (27)$$

where z is now a vector containing an arbitrary mix of inputs, outputs, and internal variables. There is then a theory based on Ritt (1950) that allows the transformation of (27) to a standard form where the properties of the system can be easily determined. The process is similar to the use of Gröbner bases but also includes the possibility of differentiating equations. Of particular interest to identification is the possibility of determining the identifiability of parameters with these tools. The model is then of the form

$$F(d^m y/dt^m, \dots, dy/dt, y, d^n z/dt^n, \dots, dz/dt, z; \theta) = 0 \quad (28)$$

where y contains measured signals, z contains unmeasured variables, and θ is a vector of parameters to be identified, while F is a vector of polynomials in these variables. It was shown in Ljung and Glad (1994) that there is an algorithm giving for each parameter θ_k a polynomial

$$g_k(d^m y/dt^m, \dots, dy/dt, y; \theta_k) = 0 \quad (29)$$

This relation can be regarded as a polynomial in θ_k where all coefficients are expressed in measured quantities. The local or global identifiability will then be determined by the number of solutions. If θ_k is unidentifiable, then no equation

of the form (29) will exist, and this fact will also be demonstrated by the output of the algorithm.

Summary and Future Directions

Grey box modeling is important to capture physical knowledge about models. It can, for instance, be based on bond graphs or Hamiltonian structures. With the increased capabilities of modeling languages grey box modeling will probably become even more important in the future.

Cross-References

- [Multi-domain Modeling and Simulation](#)
- [Nonlinear System Identification Using Particle Filters](#)
- [System Identification: An Overview](#)

Bibliography

- Brenan KE, Campbell SL, Petzold LR (1987) Numerical solution of initial-value problems in differential-algebraic equations. Classics in applied mathematics (Book 14). SIAM
- Campbell S, Illchmann A, Mehrmann V, Reis T (2019) Applications of differential-algebraic equations. Examples and benchmarks. Springer
- Duindam V, Macchelli A, Stramigioli S, Bruyninckx H (eds) (2009) Modeling and control of complex physical systems: the port-Hamiltonian approach. Springer
- Fritzson P (2015) Principles of object-oriented modeling and simulation with Modelica 3.3. IEEE Press/Wiley Interscience
- Gerdin M, Schön T, Glad T, Gustafsson F, Ljung L (2007) On parameter and state estimation for linear differential-algebraic equations. *Automatica* 43: 416–425
- Kunkel P, Mehrmann V (2006) Differential-algebraic equations. European Mathematical Society
- Ljung L, Glad ST (1994) On global identifiability of arbitrary model parameterizations. *Automatica* 30(2): 265–276
- Ljung L, Glad T (2016) Modeling and identification of dynamic systems. Studentlitteratur
- MathWorks T (2019) Simscape. The MathWorks. <https://se.mathworks.com/products/simscape.html>
- Ritt JF (1950) Differential algebra. American Mathematical Society, Providence
- Rosenberg RC, Karnopp D (1983) Introduction to physical system dynamics. McGraw-Hill
- Tiller MM (2012) Introduction to physical modeling with Modelica. Springer

Modeling of Pandemics and Intervention Strategies: The COVID-19 Outbreak

Giulia Giordano¹ and Fabrizio Dabbene^{2,3}

¹Department of Industrial Engineering,
University of Trento, Trento, TN, Italy

²CNR-IEIT, Politecnico di Torino, Torino, Italy

³National Research Council of Italy,
CNR-IEIT, Torino, Italy

Abstract

Since pathogen spillovers and pandemics are unfortunately recurrent in the history of humanity, mathematical approaches to describe and mitigate the spread of infectious diseases have a long tradition. Recently, the COVID-19 pandemic has challenged the scientific community to model, predict, and contain the contagion, especially given the current lack of pharmaceutical interventions such as vaccines and anti-viral drugs targeting the new SARS-CoV-2 coronavirus. The control community has quickly responded by providing new dynamic models of the epidemic outbreak, including both mean-field compartmental models and network-based models, as well as combinations of control approaches and intervention strategies to end the epidemic and recover a new normality. This entry deals with systems-and-control contributions to model epidemics and design effective intervention strategies, with a special focus on the COVID-19 outbreak in Italy.

Keywords

COVID-19 · Epidemics · Epidemiological models · Modeling and control of epidemics · SARS-CoV-2

Introduction

The history of humanity is unfortunately marked by the recurrent appearance of new

lethal pathogens and the subsequent spread of devastating epidemics. To better understand and face this threat, epidemiology studies the patterns of disease evolution in a population. The mathematical foundations of contemporary epidemiology have deep roots in the early twentieth century, but simpler models for disease transmission had been conceived even earlier; see Anderson and May (1991), Bailey (1975), Brauer and Castillo-Chavez (2012), Breda et al. (2012), Diekmann and Heesterbeek (2000), Hethcote (2000), House (2012), Keeling and Eames (2005), Kiss et al. (2017), Nowzari et al. (2016), Pastor-Satorras et al. (2015) and the references therein for a thorough survey.

Predictive mathematical models for epidemics are crucial not only to understand and forecast the evolution of epidemic phenomena but also to plan effective control strategies to contain the contagion. The epidemic models studied in the literature may be divided into two main classes: mean-field compartmental models (Anderson and May 1991; Bailey 1975; Brauer and Castillo-Chavez 2012; Diekmann and Heesterbeek 2000; Hethcote 2000) and network-based models (House 2012; Keeling and Eames 2005; Kiss et al. 2017; Nowzari et al. 2016; Pastor-Satorras et al. 2015). We discuss both in Section “[Mathematical Models and Control Approaches for Epidemics](#)”.

Although mathematical models for epidemics have been studied for a long time, the interest of the control community in the topic awakened only recently (Nowzari et al. 2016), with the exception of the pioneering work in Lee and Leitmann (1994) and Leitmann (1998) on control strategies for endemic infectious diseases. A boost of activity was stirred by the COVID-19 pandemic: in China, in the late 2019, a novel strand of Coronavirus denoted SARS-CoV-2 started to spread, causing a severe and potentially lethal respiratory syndrome labeled as COVID-19. Its high infectiousness enabled the virus to soon spread globally, causing a pandemic: on 8 July 2020, the World Health Organization reported 11,669,259 cases and 539,906 deaths worldwide. Given

the lack of a vaccine and of anti-viral drugs targeting SARS-CoV-2, sound modeling and prediction approaches were crucial to guide the implementation of non-pharmaceutical interventions: quarantine, social distancing and lockdown, testing and contact tracing, isolation, and use of personal protective equipment (Gatto et al. 2020; Giordano et al. 2020; Hellewell et al. 2020; Kucharski et al. 2020). The availability of open data resources (Alamo et al. 2020) favored the emergence of data-driven approaches and models, and the control community explored approaches to “flatten” (Stewart et al. 2020) or even “crush” the epidemic curve.

Italy was among the first non-Asian countries being severely affected by COVID-19. On 24 April 2020, the Italian Chapter of the IEEE Control Systems Society organized an online-workshop on *Modeling and Control of the COVID-19 outbreak: How dynamical models can help control the epidemic* to collect the most recent contributions of systems-and-control researchers to the problem of modeling and predicting the dynamics of contagion evolution. Models and approaches tailored to the specificities of the COVID-19 outbreak were proposed as concrete tools to support policymakers in deciding the best intervention strategies.

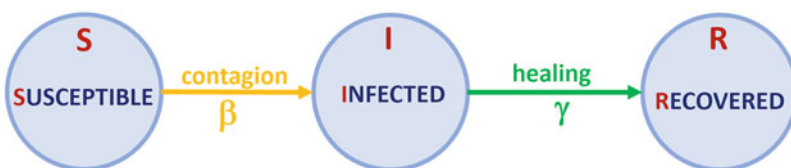
This entry illustrates classical mathematical models of epidemic spreading and recently proposed models tailored to the COVID-19 outbreak. It discusses the insight these models provide to help contain the contagion and manage the epidemic phases and surveys novel approaches for the control of epidemics with systems-and-control methodologies.

Mathematical Models and Control Approaches for Epidemics

Compartmental Models

Classical models for epidemic outbreaks are compartmental models; of these, the simplest model capturing the qualitative evolution of an epidemic phenomenon is the SIR model visualized in Fig. 1, introduced in the seminal work (Kermack and McKendrick 1927) to describe the human-to-human transmission of infectious diseases. Compartmental models are mean-field models, where all the parameters represent averaged values over the whole population and do not aim at describing the specific situation of a single individual. They rely on some simplifying assumptions:

- births, deaths (other than those due to the disease), and any other demographic processes either are absent or compensate for each other during the considered period of time, so that the total population is constant;
- the population can be partitioned into mutually exclusive classes, or compartments, depending on the stage of the disease: for the SIR model, each individual is either susceptible to infection (S), infected and infectious (I), and recovered and permanently immune (R); alternatively, R may stand for removed, namely, either recovered and immune or dead;
- the population is large and it is randomly and homogeneously mixed, so that infections occur according to the mass-action law, i.e., depending on the product between the amount of susceptible individuals and of infectious individuals, with rate constant β ;



Modeling of Pandemics and Intervention Strategies: The COVID-19 Outbreak, Fig. 1 Graph representation of the mean-field compartmental epidemiological *SIR* model: Susceptible-Infected-Recovered

- recovery (or recovery, death, and removal) from the infection occurs with rate constant γ .

To understand how the transitions between the various stages are modeled, we can adopt an analogy and represent individuals belonging to different compartments as particles or chemical species, whose concentrations evolve according to given interaction rules or reactions, each associated with a possible transition among compartments. Each “reaction” corresponds to a given “stoichiometric” law and a “reaction” rate. Then, the SIR model visualized in Fig. 1 is described by the equivalent chemical reactions $S + I \xrightarrow{\beta} 2I$ and $I \xrightarrow{\gamma} R$, which correspond to the positive dynamical system

$$\frac{ds}{dt}(t) = -\beta s(t)i(t) \quad (1)$$

$$\frac{di}{dt}(t) = \beta s(t)i(t) - \gamma i(t) \quad (2)$$

$$\frac{dr}{dt}(t) = \gamma i(t) \quad (3)$$

where $s(t)$, $i(t)$, and $r(t)$ denote, respectively, the fraction of susceptible, infected, and recovered individuals (which can be seen as the normalized concentrations of the equivalent chemical species S, I, and R) and $s(t) + i(t) + r(t) \equiv 1$ (the total population is constant, which can be seen as a conservation law). Equation (3) may be neglected, since r does not influence the other variables and its evolution can be obtained by difference: $r(t) = 1 - s(t) - i(t)$. The involved parameters are the transmission coefficient β , depending both on intrinsic features of the pathogen/infection and on the intensity of interactions among individuals, and the recovery/removal coefficient γ , depending on the average time before an individual recovers or dies from the infection; the average duration of infectiousness is $1/\gamma$. The system is positive: starting from nonnegative initial conditions, all the system variables preserve nonnegative values for their whole evolution.

We consider initial conditions with $s(0) \approx 1$, while $0 < i(0) \ll 1$ and $r(0) = 0$.

In the initial stages of the epidemic, the SIR model yields an exponential growth, or decay. In fact, since initially $s \approx 1$,

$$\frac{di}{dt}(t) \approx (\beta - \gamma)i(t), \quad (4)$$

which has the explicit exponential solution

$$i(t) \approx i(0)e^{(\beta - \gamma)t}. \quad (5)$$

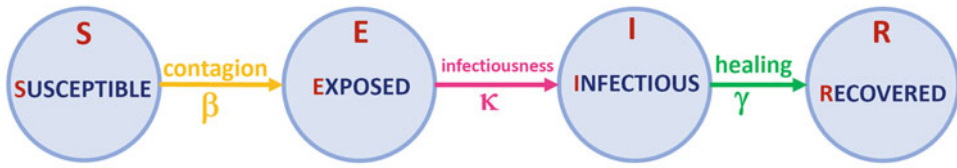
Let us define the *basic reproduction number* R_0 , corresponding to the average number of secondary infections caused by a primary case introduced in a fully susceptible population (Anderson and May 1991; Brauer and Castillo-Chavez 2012), as

$$R_0 \doteq \frac{\beta}{\gamma}. \quad (6)$$

Then, Eq. (5) represents an exponential growth if $\beta - \gamma > 0$, namely, $R_0 = \frac{\beta}{\gamma} > 1$, and an exponential decay if $\beta - \gamma < 0$, namely, $R_0 = \frac{\beta}{\gamma} < 1$.

When the complete dynamics in Eqs. (1)–(3) are considered without approximations, including later stages of infection, the value of R_0 still has a key role in determining the epidemic outcome. Denote by $\bar{s}$, $\bar{i}$, and $\bar{r}$ the asymptotic equilibrium value. Starting from an initial condition with $i(0) > 0$, it must be $\bar{s} < s(0)$, since $s(t)$ is monotonically decreasing in view of Eq. (1), while $\bar{r} > r(0)$, since $r(t)$ is monotonically increasing in view of Eq. (3). If $R_0 s(0) < 1$ (hence $R_0 s(t) < 1$ for all $t \geq 0$, being $s(t)$ decreasing), the fraction of infected population decreases, because $\frac{di}{dt} < 0$, as can be seen from (2), until $i(t)$ reaches zero; then there is no outbreak and infections spontaneously decay. Conversely, if $R_0 s(0) > 1$, the fraction of infected population is initially increasing, because $\frac{di}{dt} > 0$. However, since $s(t)$ is decreasing, $R_0 s(t)$ decreases too: it reaches from above the threshold 1 and then goes below. Therefore, the fraction of infected population reaches a peak, at the time t^* such that $R_0 s(t^*) = 1$, and eventually decreases to zero (because, for $t > t^*$, $R_0 s(t) < 1$).

Any equilibrium of the system is such that there are no infected individuals, $\bar{i} = 0$: the



Modeling of Pandemics and Intervention Strategies: The COVID-19 Outbreak, Fig. 2 Graph representation of the mean-field compartmental epidemiological *SEIR* model: Susceptible-Exposed-Infectious-Recovered

generic equilibrium has the form $(\bar{s}, 0, \bar{r})$, with $\bar{s} + \bar{r} = 1$. To ensure stability, the equilibrium value of s must be such that $R_0\bar{s} < 1$.

If we consider the sum of Eqs. (1) and (2), we obtain

$$\frac{ds}{dt}(t) + \frac{di}{dt}(t) = -\gamma i(t). \quad (7)$$

Integration of Eq. (7) yields

$$\int_0^t i(\tau) d\tau = \frac{1}{\gamma} [s(0) - s(t) + i(0) - i(t)]. \quad (8)$$

Also, integrating Eq. (1) divided by s gives

$$\log\left(\frac{s(0)}{s(t)}\right) = \beta \int_0^t i(\tau) d\tau \quad (9)$$

and substituting Eq. (8) into Eq. (9) gives

$$\log\left(\frac{s(0)}{s(t)}\right) = \underbrace{\frac{\beta}{\gamma}}_{R_0} [s(0) - s(t) + i(0) - i(t)]. \quad (10)$$

Rearranging Eq. (10) computed at time $t = t^*$, where t^* is defined above, allows us to compute the peak value $i^{\max}$ of i , which is achieved for $s^* = s(t^*) = \frac{1}{R_0}$:

$$i^{\max} = i(0) + s(0) - \frac{1}{R_0} [1 + \log(R_0 s(0))]. \quad (11)$$

If we start from an initial condition with $r(0) = 0$, hence $s(0) + i(0) = 1$, Eq. (10) allows us to compute the equilibrium value of the susceptible population as the unique positive root of the equation

$$\log\left(\frac{s(0)}{\bar{s}}\right) = R_0[1 - \bar{s}], \quad (12)$$

known as *final size relation*. The quantity $1 - \bar{s} = \bar{r}$, called *attack rate* by the epidemiologists, corresponds to the fraction of the population having experienced the disease during the epidemic.

Several extensions of the SIR model were proposed, taking into account additional infection stages: for instance, the SEIR (Susceptible, Exposed, Infectious, Recovered) model, shown in Fig. 2, considers a latency stage when the infected person is not yet infectious.

The SEIR model is described by the equivalent chemical reactions $S + I \xrightarrow{\beta} E + I$, $E \xrightarrow{\kappa} I$, $I \xrightarrow{\gamma} R$, corresponding to the positive dynamical system

$$\frac{ds}{dt}(t) = -\beta s(t)i(t) \quad (13)$$

$$\frac{de}{dt}(t) = \beta s(t)i(t) - \kappa e(t) \quad (14)$$

$$\frac{di}{dt}(t) = \kappa e(t) - \gamma i(t) \quad (15)$$

$$\frac{dr}{dt}(t) = \gamma i(t) \quad (16)$$

where $s(t)$, $e(t)$, $i(t)$, and $r(t)$ denote, respectively, the fraction of susceptible, exposed, infectious, and recovered individuals and $s(t) + e(t) + i(t) + r(t) \equiv 1$. In this case, all possible equilibria have the form $(\bar{s}, 0, 0, \bar{r})$ with $\bar{s} + \bar{r} = 1$.

More complex models were proposed to better describe the specific features of a given disease; for instance, the SEQIJR model put forth in Gumel et al. (2004) is tailored to SARS. The reader is referred to Anderson and May (1991), Bailey (1975), Brauer and Castillo-Chavez (2012), Diekmann and Heesterbeek (2000), and Hethcote (2000) for further details

on extended compartmental models for epidemic spreading and their analysis.

Network-Based Models

Network-based models embed compartmental models into a graph-theoretic framework: the interacting population is represented by a graph where the nodes are associated with individuals, whose state corresponds to one of the stages in compartmental models (e.g., susceptible, infected, recovered, and immune), and the links among them are associated with human-to-human interactions that can potentially lead to contagion. The transmission rate is no longer an averaged value for a well-mixed population, but is tailored to each specific node, depending on its connectivity degree and on the state of its neighboring nodes. Both stochastic and deterministic population models, and both stochastic and deterministic network models, can be adopted. The models can then describe how the total number of nodes at each of the infection stages evolves over time, depending on the initial conditions and the network topology, by monitoring the fractions of nodes in each stage at each time instant (in a deterministic framework), or the probability that a node belongs to a given stage (in a stochastic framework, based on the evaluation of expectations and moment closures). Network epidemics can also be seen as a percolation phenomenon (House 2012; Pastor-Satorras et al. 2015).

Network-based models also include agent-based models, typically spatially structured, which consider the discrete nature of individuals and their mobility and interaction patterns in a stochastic framework. These models are extremely complex and detailed, since they rely on the construction of a synthetic population reproducing each individual and her/his neighborhood and movements; hence, efficient data-driven computational approaches are needed.

In meta-population models, each node of the network corresponds not to a single individual in the population, but to a group of multiple individuals. We can also consider networked models where each node is associated with a compartmental dynamical system, describing local epi-

demical spreading, and each link represents mobility between local communities, which spreads the epidemic to larger geographic areas (Mei et al. 2017; Paré et al. 2018; Ye et al. 2020).

For more information on epidemics on networks, the reader is referred to House (2012), Keeling and Eames (2005), Kiss et al. (2017), Nowzari et al. (2016), and Pastor-Satorras et al. (2015).

Control Approaches

Several approaches were considered to mitigate the effects of an epidemic outbreak. In a compartmental SIR model, clearly the epidemics can be suppressed by increasing the recovery rate γ (thanks to improved treatment for infected individuals) and decreasing the transmission rate β (which can be achieved by raising awareness, use of personal protective equipment, social distancing, quarantining and isolating infected individuals, or even travel limitations and lockdown), so that $R_0 = \beta/\gamma$ becomes as small as possible, at least smaller than 1 to make the epidemic phenomenon decay.

In network-based epidemic models, with the constraint of a limited budget, one may be interested in optimally investing fixed resources to reduce at best the spreading of the disease, so as to minimize the quantity $\lambda_{\max}(B - \Gamma)$, where $\lambda_{\max}$ denotes the dominant eigenvalue, B is the matrix of infection rates β_{ij} among nodes, and Γ is the diagonal matrix of recovery rates γ_i for the individual nodes. Two possible strategies can be devised for spectral control and optimization: in order to reduce $\lambda_{\max}$, either network nodes or network links can be removed. This corresponds to isolation/quarantine for some individuals (node removal) and to social distancing (link removal).

Optimal control approaches were proposed, relying on Pontryagin's maximum principle, to minimize the cost of the epidemics, which combines the cost of infection and the cost of treatment or vaccination (Bloem et al. 2009; Forster and Gilligan 2007; Hansen and Day 2011; Morton and Wickwire 1974), so as to design an optimal treatment plan, or vaccination plan. Robust control approaches were also proposed to control

the spreading of infectious diseases, seen as an uncertain dynamical system (Lee and Leitmann 1994; Leitmann 1998). Given the complexity of the problem, numerous heuristic feedback control approaches were put forth as well. The reader is referred to Nowzari et al. (2016) for a comprehensive survey of the control of spreading processes.

Modeling and Control of the COVID-19 Outbreak

Several models of the COVID-19 pandemic started to appear in the early 2020. Stochastic transmission models were studied in Hellewell et al. (2020) and Kucharski et al. (2020) to analyze disease transmission and its control by isolation of cases. Also generalized compartmental models were considered: Lin (2020) extends a SEIR model considering perceived contagion risk and cumulative number of cases; Anastassopoulou et al. (2020) analyzes and forecasts the epidemic evolution in China based on a discrete-time SIR model explicitly accounting for dead individuals; and Wu et al. (2020) considers a meta-population SIR model, partitioning the population into age groups, to estimate the clinical severity of COVID-19 from transmission dynamics.

The compartmental SIDARTHE (Susceptible-Infected-Diagnosed-Ailing-Recognized-Threatened-Healed-Extinct; see Fig. 3) model is proposed in Giordano et al. (2020) to understand and predict the COVID-19 epidemic evolution, distinguishing between infected with different severity of illness and between detected and undetected cases of infection. The model highlights the parameters associated with the two main non-pharmaceutical interventions: social distancing and lockdown (which reduce the contagion parameters) and testing and contact tracing (which increase the diagnosis parameters, so that more infection cases are isolated). Different scenarios are explored to assess the effect of various interventions in the Italian case, and the results support the combination of social

distancing with testing and contact tracing so as to rapidly end the epidemic.

A modified SIR model including both recovered and deceased, and taking into account that only a portion of infected individuals can be detected, is proposed in Calafiore et al. (2020). The model predicts the evolution of the contagion in Italy, and in each of its regions, so as to assess the effectiveness of containment and lockdown measures.

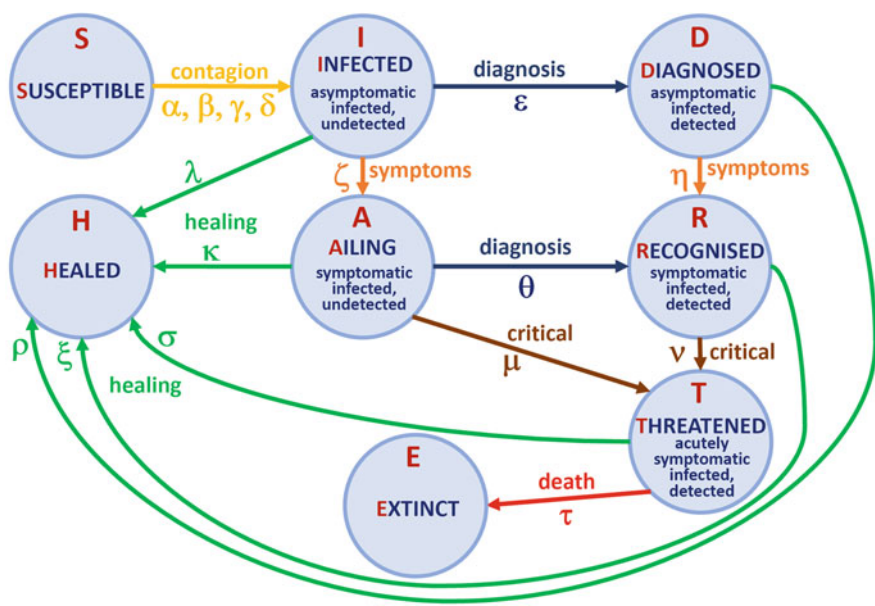
An extended SIR model that distinguishes between asymptomatic and symptomatic infected, as well as actual and confirmed cases, is used in Russo et al. (2020) to identify the estimated day zero of the COVID-19 outbreak in Lombardy (Italy).

A compartmental model that highlights the fraction of asymptomatic infectious individuals, considers the COVID-19 spread within and among regions in Italy, and assesses the impact of the adopted interventions is proposed in Di Giamberardino et al. (2020).

A feedback SIR model, with nonlinear transmission rates (Capasso and Serio 1978), is considered in Franco (2020) to study infection-based social distancing and its advantages (the infection peak is reduced, even in the presence of information delays) and disadvantages (extended duration of the epidemic).

The control-oriented SEIR model developed in Casella (2020) stresses the effect of delays and compares the outcomes of different containment policies, in China and in Lazio (Italy). It is shown that mitigation strategies (letting the epidemic run in a controlled way, typically aiming for herd immunity) are likely to fail because they aim at controlling fast unstable dynamics affected by time delays and uncertainties, while suppression strategies can be successful if they are prompt and drastic enough.

Fast multi-shot intermittent lockdown intervals with regular period are proposed in Bin et al. (2020) as an exit strategy from total lockdown to avoid second waves of infection. The suggested switching control strategy is open-loop and robust to delays and uncertainties in measurements, and the parameters of the mitigation strategy are tuned by a slow and robust



Modeling of Pandemics and Intervention Strategies: The COVID-19 Outbreak, Fig. 3 Graph representation of the mean-field compartmental epidemiolog-

ical *SIDARTHE* (Susceptible-Infected-Diagnosed-Ailing-Recognised-Threatened-Healed-Extinct) model. (Reproduced from Giordano et al. 2020)

outer supervisory feedback loop; the proposed approach is successfully tested on compartmental epidemic models ranging from the SIR to the SIDARTHE model (Giordano et al. 2020).

A robust and optimal control strategy for the COVID-19 outbreak is proposed in Köhler et al. (2020) based on model predictive control approaches that dynamically adapt social distancing measures, using the SIDARTHE model (Giordano et al. 2020).

An extended SIR model with specific compartments for socially distanced susceptible and infected (either asymptomatic or symptomatic) individuals is introduced in Gevertz et al. (2020). The analysis of time-varying social distancing strategies reveals that they are effective only if enacted quickly enough, and there is a critical intervention delay after which they have little effect; moreover, periodic relaxation strategies can be effective but are extremely fragile to small errors in timing or parameters, while gradual relaxation substantially improves the epidemic situation, but the rate of relaxation needs to be carefully chosen to prevent a second outbreak.

Economists modeled the epidemic resorting to a SIR model that incorporates the optimizing behavior of rational agents, incentives influencing the transitions, and externalities representing restrictive government interventions (Garibaldi et al. 2020): agent-based rational decision-making, leading to a decentralized epidemic equilibrium, yields outcomes consistent with the original model in Kermack and McKendrick (1927).

The work in Gatto et al. (2020) analyzes the local-global effect of containment measures by studying a meta-population SEIR-like model that interconnects the epidemics in different Italian provinces to model the spatial spreading of the outbreak; the resolution that models the disease spread at the appropriate geographical scale allows for both spatial and temporal design of containment measures and makes it possible to forecast the medical resources and infrastructures needed. The same spatially explicit approach is adopted in Bertuzzo et al. (2020) to suggest appropriate relaxations of the containment measures adopted in Italy, discussing possible

options such as tracing and testing, stop-and-go lockdown enforcement, and delayed lockdown relaxations.

A networked meta-population SIR model is proposed in Della Rossa et al. (2020) to describe the Italian epidemics by looking at the country as a network of regions, each with different epidemic parameters; this reveals the different regional effects of the adopted countermeasures, as well as the important impact of regional heterogeneity on the outbreak evolution, and suggests that differentiated (but coordinated) feedback interventions enforced at a regional level could be particularly beneficial.

A meta-population networked model is also proposed in Zino et al. (2020), focused on the role of activity and mobility in the COVID-19 outbreak in Italy: the model considers the reduction in activity (to study the effect of “stay at home” policies) and mobility within and between provinces (to study the effect of mobility limitations, travel bans, and isolation of red zones), as well as isolation (to study the effect of quarantine and testing).

A different methodology is proposed in Fanti et al. (2020), where a multi-criteria approach is adopted for COVID-19 risk assessment in different lockdown scenarios, focusing on urban district lockdowns in Puglia (Italy).

Several epidemic scenarios with different testing policies and mobility restrictions are outlined in Dahleh et al. (2020), where a network SIR-like model is used to explore control approaches based on testing, distancing, and quarantining.

The challenges of forecasting the spread of COVID-19 using different models (exponential, SIR, and Hawkes self-exciting branching process) are discussed in Bertozzi et al. (2020).

Model accuracy and validation The proposed models were tested against official epidemiological data provided by local and national governments. In spite of the large noise and uncertainty affecting the available data on the COVID-19 epidemic evolution, the proposed models, with the parameters fit based on real data, proved to

be able to accurately reproduce and predict the outbreak dynamics.

Summary and Future Directions

We provided an overview of dynamic models, both compartmental and network-based, to describe, predict, and control the evolution of epidemics, with a special focus on the models and intervention strategies tailored to the ongoing COVID-19 pandemic.

Models proved to be a precious tool not only to forecast epidemic phenomena but also to assess and predict the effectiveness of different policies and countermeasures. Non-pharmaceutical strategies are lockdown and social distancing (including the widespread use of personal protective equipment), population-scale testing, and contact tracing. Each strategy has some fragilities. The impact of lockdown on the economy, as well as on mental health (Torales et al. 2020), needs to be taken into account, while the effectiveness and correct use of personal protective equipment has been long debated. Testing strategies need to cope with a limited availability of tests; this problem is studied in Drakopoulos and Randhawa (2020) using a resource allocation approach, and it is shown that it may prove efficient to release less accurate tests when the epidemic is picking up and then more accurate tests as they become available. Contact tracing requires the adoption of individual tracking and monitoring via apps that generate privacy concerns.

In this complex situation, outlining different scenarios and predictions based on the available data and on solid mathematical modeling can inform and guide policymakers deciding how to handle the ongoing pandemic. Therefore, we are convinced that the contribution of the systems-and-control community will be valuable to contain the COVID-19 outbreak, as well as future pandemics. At the same time, the peculiar characteristics of epidemiological models pose important theoretical challenges that are pushing out

the frontiers of the methodological tools developed by our community.

Cross-References

- [Controlling Collective Behavior in Complex Systems](#)
- [Dynamical Social Networks](#)
- [Robustness Analysis of Biological Models](#)

Bibliography

References marked with an asterisk (*) denote contributions presented at the online-workshop on *Modeling and Control of the COVID-19 outbreak: How dynamical models can help control the epidemic* (24 April 2020) organized by the Italian Chapter of the IEEE Control Systems Society.

- Alamo T, Reina DG, Mammarella M, Abella A (2020) Open data resources for fighting COVID-19. *Electronics* 9:827, 1–28
- Anastassopoulou C, Russo L, Tsakris A, Siettos C (2020) Data-based analysis, modelling and forecasting of the COVID-19 outbreak. *PLoS One* 15:e0230405
- Anderson RM, May RM (1991) *Infectious diseases of humans*. Oxford University Press
- Bailey NTJ (1975) *The mathematical theory of infectious diseases and its applications*. Charles Griffin, London/High Wycombe
- Bertozzi AL, Franco E, Mohler G, Short MB, Sledge D (2020) The challenges of modeling and forecasting the spread of COVID-19. *PNAS*, 202006520
- Bertuzzo E, Mari L, Pasetto D, Miccoli S, Casagrandi R, Gatto M, Rinaldo A (2020) The geography of COVID-19 spread in Italy and implications for the relaxation of confinement measures. <https://doi.org/10.1101/2020.04.30.20083568>
- Bin M, Cheung P, Crisostomi E, Ferraro P, Lhachemi H, Murray-Smith R, Myant C, Parisini T, Shorten R, Stein S, Stone L (2020) On fast multi-shot COVID-19 interventions for post lock-down mitigation. <https://arxiv.org/abs/2003.09930> (*)
- Bloem M, Alpcan T, Basar T (2009) Optimal and robust epidemic response for multiple networks. *Control Eng Pract* 17(5):525–533
- Brauer F, Castillo-Chavez C (2012) *Mathematical models in population biology and epidemiology*, 2nd edition, Springer
- Breda D, Diekmann O, de Graaf WF, Pugliese A, Vermiglio R (2012) On the formulation of epidemic models (an appraisal of Kermack and McKendrick). *J Biol Dyn* 6(2):103–117
- Calafiore GC, Novara C, Possieri C (2020) A modified SIR model for the COVID-19 contagion in Italy. <https://arxiv.org/abs/2003.14391> (*)
- Capasso V, Serio G (1978) A generalization of the Kermack-McKendrick deterministic epidemic model. *Math Biosci* 42:43–61
- Casella F (2020) Can the COVID-19 epidemic be controlled on the basis of daily test reports? <https://arxiv.org/abs/2003.06967> (*)
- Dahleh MA (representing IDSS-COVID-19 Collaboration Group, ISOLAT) (2020) Inching back to normal after COVID-19 lockdown: quantification of interventions; see also <https://idss.mit.edu/research/idss-covid-19-collaboration-isolat> (*)
- Della Rossa F, Salzano D, Di Meglio A, De Lellis F, Coraggio M, Calabrese C, Guarino A, Cardona R, De Lellis P, Liuzza D, Lo Iudice F, Russo G, Di Bernardo M (2020) Intermittent yet coordinated regional strategies can alleviate the COVID-19 epidemic: a network model of the Italian case. <https://arxiv.org/abs/2005.07594> (*)
- Diekmann O, Heesterbeek JAP (2000) *Mathematical epidemiology of infectious diseases: model building, analysis and interpretation*. Wiley
- Di Giamberardino P, Iacoviello D, Papa F, Sinisgalli C (2020) A new mathematical model of COVID-19 spread: analysis of the impact of intervention actions and evaluation of the asymptomatic infectious subjects (*)
- Drakopoulos K, Randhawa RS (2020) Why perfect tests may not be worth waiting for: Information as a commodity. Available at SSRN. <https://doi.org/10.2139/ssrn.3565245>
- Fanti MP, Parisi F, Sangiorgio V (2020) A multicriteria approach for risk assessment of COVID-19 in urban district lockdowns (*)
- Forster GA, Gilligan CA (2007) Optimizing the control of disease infestations at the landscape level. *PNAS* 104(12):4984–4989
- Franco E (2020) A feedback SIR (fSIR) model highlights advantages and limitations of infection-based social distancing. <https://arxiv.org/abs/2004.13216> (*)
- Garibaldi P, Moen ER, Pissarides C (2020) Modelling contacts and transitions in the SIR epidemics model. *Covid Econ* 5:1–20 (*)
- Gatto M, Bertuzzo E, Mari L, Miccoli S, Carraro L, Casagrandi R, Rinaldo A (2020) Spread and dynamics of the COVID-19 epidemic in Italy: effects of emergency containment measures. *PNAS* 117(19):10484–10491
- Gevertz JL, Greene JM, Sanchez-Tapia C, Sontag ED (2020) A novel COVID-19 epidemiological model with explicit susceptible and asymptomatic isolation compartments reveals unexpected consequences of timing social distancing. <https://doi.org/10.1101/2020.05.11.20098335>
- Giordano G, Blanchini F, Bruno R, Colaneri P, Di Filippo A, Di Matteo A, Colaneri M (2020) Mod-

- elling the COVID-19 epidemic and implementation of population-wide interventions in Italy. *Nat Med* 26:855–860
- Gumel AB et al (2004) Modelling strategies for controlling SARS outbreaks. *Proc R Soc B Biol Sci* <https://doi.org/10.1098/rspb.2004.2800>
- Hansen E, Day T (2011) Optimal control of epidemics with limited resources. *J Math Biol* 62(3):423–451
- Hellewell J et al. (2020) Feasibility of controlling COVID-19 outbreaks by isolation of cases and contacts. *Lancet Global Health* 8:e488–e496
- Hethcote HW (2000) The mathematics of infectious diseases. *SIAM Rev* 42:599–653
- House T (2012) Modelling epidemics on networks. *Contemp Phys* 53(3):213–225
- Keeling MJ, Eames KTD (2005) Networks and epidemic models. *J R Soc Interface* 2:295–307
- Kermack WO, McKendrick AG (1927) A contribution to the mathematical theory of epidemics. *Proc R Soc Lond* 115:700–721
- Kiss IZ, Miller JC, Simon P (2017) *Mathematics of epidemics on networks: from exact to approximate models*. Springer
- Köhler J, Schwenkel L, Koch A, Berberich J, Pauli P, Allgöwer F (2020) Robust and optimal predictive control of the COVID-19 outbreak. <https://arxiv.org/abs/2005.03580>
- Kucharski AJ et al (2020) Early dynamics of transmission and control of COVID-19: a mathematical modelling study. *Lancet Global Health*. [https://doi.org/10.1016/S1473-3099\(20\)30144-4](https://doi.org/10.1016/S1473-3099(20)30144-4)
- Lee CS, Leitmann, G (1994) Control strategies for an endemic disease in the presence of uncertainty. In: Agarwal RP (ed) *Recent trends in optimization theory and applications*. World Scientific, Singapore
- Leitmann G (1998) The use of screening for the control of an endemic disease. *International series of numerical mathematics*, vol 124. Birkhäuser, Basel, pp 291–300
- Lin Q et al (2020) A conceptual model for the coronavirus disease 2019 (COVID-19) outbreak in Wuhan, China with individual reaction and governmental action. *Int J Inf Dis* 93:211–216
- Mei W, Mohagheghi S, Zampieri S, Bullo F (2017) On the dynamics of deterministic epidemic propagation over networks. *Ann Rev Control* 44:116–128
- Morton R, Wickwire KH (1974) On the optimal control of a deterministic epidemic. *Adv Appl Probab* 6(4):622–635
- Nowzari C, Preciado VM, Pappas GJ (2016) Analysis and control of epidemics: a survey of spreading processes on complex networks. *IEEE Control Systems Magazine*, Feb, pp 26–46
- Pastor-Satorras R, Castellano C, Van Mieghem P, Vespignani A (2015) Epidemic processes in complex networks. *Rev Mod Phys* 87(3):925–979
- Paré PE, Beck CL, Nedić A (2018) Epidemic processes over time-varying networks. *IEEE Trans Control Netw Syst* 5(3):1322–1334
- Russo L, Anastassopoulou C, Tsakris A, Bifulco GN, Campana EF, Toraldo G, Siettos C (2020) Modelling, tracing day-zero and forecasting the fade out of the COVID-19 outbreak: experiences from China and Lombardy studies. <https://doi.org/10.1101/2020.03.17.20037689> (*)
- Stewart G, van Heusden K, Dumont GA (2020) How control theory can help us control COVID-19. *IEEE Spectrum*. <https://spectrum.ieee.org/biomedical/diagnostics/how-control-theory-can-help-control-covid19>
- Torales J, O’Higgins M, Castaldelli-Maia JM, Ventriglio A (2020) The outbreak of COVID-19 coronavirus and its impact on global mental health. *Int J Soc Psychiatry* 66:317–320
- Wu J et al (2020) Estimating clinical severity of COVID-19 from the transmission dynamics in Wuhan, China. *Nat Med* 26:506–510
- Ye M, Liu J, Anderson BDO, Cao M (2020) Applications of the Poincaré–Hopf theorem: epidemic models and Lotka–Volterra systems. <https://arxiv.org/abs/1911.12985>
- Zino L, Parino F, Porfiri M, Rizzo A (2020) A metapopulation activity-driven network model for COVID-19 in Italy (*)

Modeling, Analysis, and Control with Petri Nets

M

Manuel Silva

Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza, Spain

Abstract

Petri net is a generic term used to designate a broad family of related formalisms for discrete event views of (dynamic) systems (DES), all sharing some basic relevant features, such as *minimality* in the number of primitives, *locality* of the states and actions (with consequences for model construction), or *temporal realism*. The global state of a system is obtained by the juxtaposition of the different local states. We should initially distinguish between *autonomous* formalisms and those *extended by interpretation*. Models in the latter group are obtained by restricting the underlying autonomous behaviors by means

of constraints that can be related to different kinds of external events, in particular to time. This article first describes *place/transition* nets (PT-nets), by default simply called Petri nets (PNs). Other formalisms are then mentioned. As a system theory modeling paradigm for concurrent DES, Petri nets are used in a wide variety of application fields.

Keywords

Condition/event nets (CE-nets) · Continuous petri nets (CPNs) · Diagrams · Fluidization · Grafcet · Hybrid petri nets (HPNs) · Marking petri nets · Place/transition nets (PT-nets) · High-level Petri nets (HLPNs)

Introduction

Petri nets (PNs) are able to model concurrent and distributed DES (► [“Models for Discrete Event Systems: An Overview”](#)). They constitute a powerful family of formalisms with different expressive purposes and power. They may be applied to, inter alia, modeling, logical analysis, performance evaluation, parametric optimization, dynamic control (minimum makespan, supervisory control, or other kinds), diagnosis, and implementation issues (eventually fault tolerant). Hybrid and continuous PNs are particularly useful when some parts of the system are highly populated. Being *multidisciplinary*, formalisms belonging to the Petri nets paradigm may cover several phases of the life cycle of complex DES.

A Petri net can be represented as a bipartite directed graph provided with arcs inscriptions; alternatively, this structure can be represented in algebraic form using some matrices. As in the case of differential equations, an initial condition or state should be defined in order to represent a dynamic system. This is done by means of an initial distributed state. The English translation of the Carl Adam Petri’s seminal work, presented in 1962, is Petri (1966).

Untimed Place/Transition Net Systems

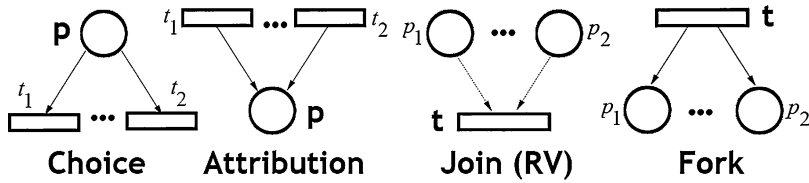
A place/transition net (PT-net) can be viewed as $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$, where:

- P and T are disjoint and finite nonempty sets of *places* and *transitions*, respectively.
- $\mathbf{Pre}$ and $\mathbf{Post}$ are $|P| \times |T|$ sized, natural-valued (zero included), and incidence matrices. The net is said to be *ordinary* if $\mathbf{Pre}$ and $\mathbf{Post}$ are valued on $\{0, 1\}$. Weighted arcs permit the abstract modeling of *bulk* services and arrivals.

A PT-net is a structure. The $\mathbf{Pre}$ ($\mathbf{Post}$) function defines the connections from places to transitions (transitions to places). Those two functions can alternatively be defined as *weighted flow* relations (nets as graphs). Thus, PT-nets can be represented as *bipartite directed graphs* with *places* (p , using circles) and *transitions* (t , using bars or rectangles) as nodes: $\mathcal{N} = \langle P, T, F, W \rangle$, where $F \subseteq (P \times T) \cup (T \times P)$ is the *flow relation* (set of directed arcs, with $\text{dom}(F) \cup \text{range}(F) = P \cup T$) and $W : F \rightarrow \mathbb{N}^+$ assigns a natural weight to each arc.

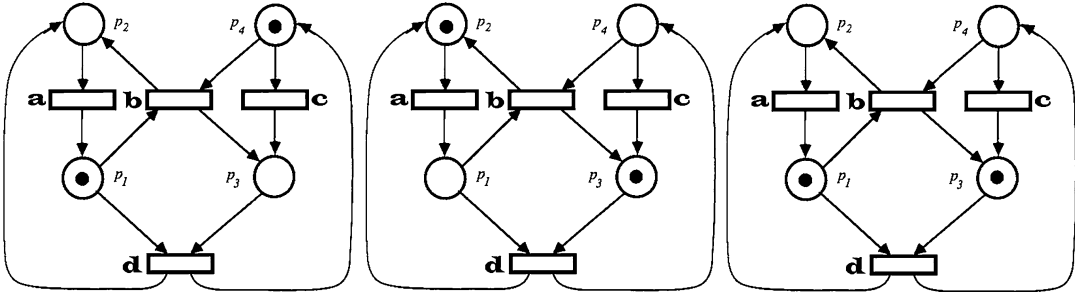
The net structure represents the *static* part of the DES model. Furthermore, a “distributed state” is defined over the set of places, known as the *marking*. This is “numerically quantified” (not in an arbitrary alphabet, as in automata), associating *natural* values to the local state variables, the places. If a place p has a value v (i.e., $\mathbf{m}(p) = v$), it is said to have v *tokens* (frequently depicted in graphic terms with v black dots or just the number inside the place). The places are “state variables,” while the markings are their “values”; the *global state* is defined through the concatenation of local states. The net structure, provided with an initial marking, to be denoted as $(\mathcal{N}, \mathbf{m}_0)$, is a *Petri net system*, or *marked Petri net*.

The last two basic PN constructions in Fig. 1 (*join* and *fork*) do not appear in finite-state machines; moreover, the arcs may be valued with natural numbers. The dynamic behavior of the net



Modeling, Analysis, and Control with Petri Nets, Fig. 1 Most basic PN constructions: The logical OR is present around places, in *choices* (or branches) and

attributions (or meets); the logical AND is formed around transitions, in *joins* (or waits or *rendezvous*) and *forks* (or splits)



Modeling, Analysis, and Control with Petri Nets, Fig. 2 Only transitions *b* and *c* are initially enabled. The results of firing *b* or *c* are shown subsequently

system (trajectories with changes in the marking) is produced by the firing of transitions, some “local operations” which follows very simple rules.

Markings in net systems evolve according to the following *firing* (or *occurrence*) rules (see Fig. 2):

- A transition is said to be *enabled* at a given marking if each input place has at least as many tokens as the weight of the arc joining them.
- The *firing* or *occurrence* of an enabled transition is an instantaneous operation that removes from (adds to) each input (output) place a number of tokens equal to the weight of the arc joining the place (transition) to the transition (place).

The precondition of a transition can be seen as the resources required for the transition to be fired. The weight of the arc from a place to a transition represents the number of resources to be *consumed*. The post-condition defines the number resources *produced* by the firing of the transition.

This is made explicit by the weights of the arcs from the transition to the places. Three important observations should be taken into account:

- The underlying logic in the firing of a transition is non-monotonic! It is a *consumption/production* logic.
- Enabled transitions are never *forced* to fire: This is a form of *non-determinism*.
- An *occurrence sequence* is a sequence of fired transitions $\sigma = t_1 \dots t_k$. In the evolution from m_0 , the reached marking m can be easily computed as:

$$m = m_0 + C \cdot \sigma, \quad (1)$$

where $C = \text{Post} - \text{Pre}$ is the *token flow* matrix (*incidence* matrix if $\mathcal{N}$ is self-loop free) and σ the firing count vector corresponding to σ . Thus m and σ are vectors of natural numbers.

The previous equation is the *state-transition* equation (frequently known as the *fundamental* or, simply, *state* equation). Nevertheless, two important remarks should be made:

- It represents a necessary but not sufficient condition for reachability; the problem is that the existence of a σ does not guarantee that a corresponding sequence σ is firable from m_0 ; thus, certain solutions – called *spurious* (Silva et al. 1998) – are not reachable. This implies that – except in certain net system subclasses – only semi-decision algorithms can usually be derived.
- All variables are natural numbers, which imply computational complexity.

It should be pointed out that in finite-state machines, the state is a single variable taking values in a symbolic unstructured set, while in PT-net systems, it is structured as a vector of nonnegative integers. This allows analysis techniques that do not require the enumeration of the state space.

At a structural level, observe that the *negation* is missing in Fig. 1; its inclusion leads to the so-called inhibitor arcs, an extension in expressive power. In its most basic form, if the place at the origin of an inhibitor arc is marked, it “inhibits” the enabling of the target transition. PT-net systems can model infinite-state systems, but not Turing machines. PT-net systems provided with inhibitor arcs (or *priorities* on the firing of transitions) can do it.

With this conceptually simple formalism, it is not difficult to express basic synchronization schemas (Fig. 3). All the illustrated examples use *joins*. When *weights* are allowed in the arcs, another kind of synchronization appears: Several copies of the same resource are needed (or produced) in a single operation. Being able to express *concurrency* and *synchronization*, when viewing the system at a higher level, it is possible to build *cooperation* and *competition* relationships.

Analysis and Control of Untimed PT Models

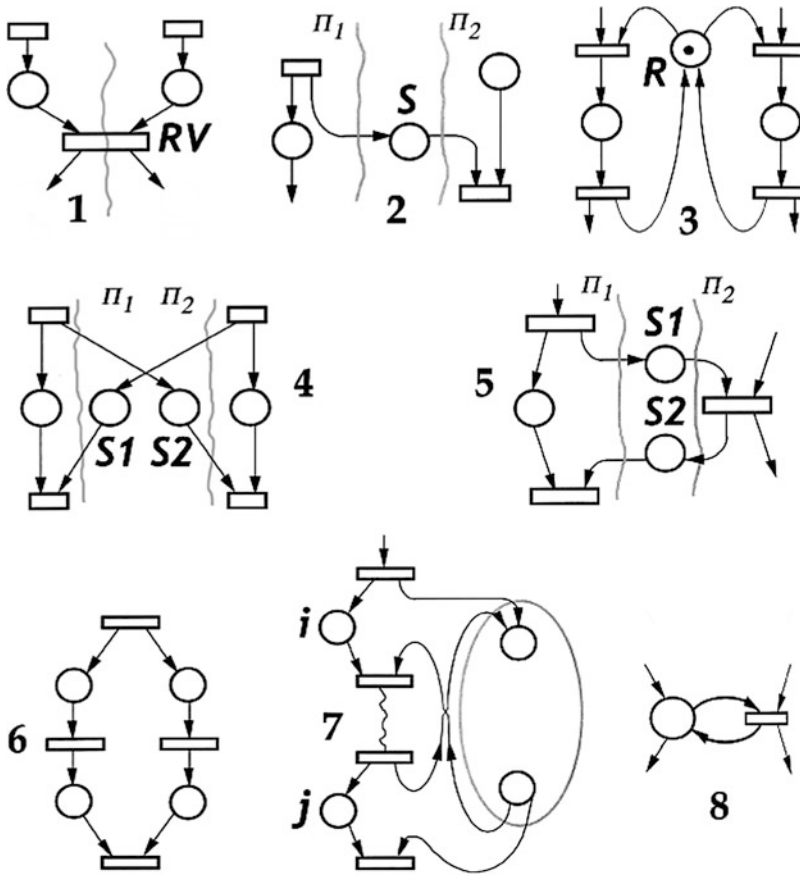
The behavior of a concurrent (eventually distributed) system is frequently difficult to understand and control. Thus, misunderstandings

and mistakes are frequent during the design cycle. A way of cutting down the cost and duration of the design process is to express in a formalized way properties that the system should enjoy and to use formal proof techniques. Errors can be eventually detected close to the moment they are introduced, reducing their propagation to subsequent stages. The goal in *verification* is to ensure that a given system is correct with respect to its specification (perhaps expressed in temporal-logic terms) or to a certain set of predetermined properties.

Among the most basic qualitative properties of “net systems” are the following: (1) *reachability* of a marking from a given one; (2) *boundedness*, characterizing finiteness of the state space; (3) *liveness*, related to potential fireability of all transitions starting on an arbitrary reachable marking (*deadlock-freeness* is a weaker condition in which only global infinite fireability of the net system model is guaranteed, even if some transitions no longer fire); (4) *reversibility*, characterizing recoverability of the initial marking from any reachable one; and (5) *mutual exclusion of two places*, dealing with the impossibility of reaching markings in which both places are simultaneously marked.

All the above are *behavioral* properties, which depend on the net system $(\mathcal{N}, m_0)$. In practice, sometimes problems with a net model are rooted in the net structure; thus, the study of the *structural* counterpart of certain behavioral properties may be of interest. For example, a “net” is *structurally bounded* if it is bounded for any initial marking; a “net” is *structurally live* if an initial marking exists that make the net system live (otherwise stated, it reflect non-liveness for arbitrary initial markings, a pathology of the net).

Basic techniques to analyze net systems include (1) *enumeration*, in its most basic form based on the construction of a *reachability graph* (a sequentialized view of the behavior). If the net system is not bounded, losing some information, a finite *coverability graph* can be constructed; (2) *transformation*, based on an iterative rewriting process in which a net system enjoys a certain property if and only if a transformed (“simpler” to analyze) one also



Modeling, Analysis, and Control with Petri Nets,
Fig. 3 Basic synchronization schemes: (1) Join or rendezvous, RV; (2) Semaphore, S; (3) Mutual exclusion semaphore (mutex), R, representing a shared resource; (4) Symmetric RV built with two semaphores; (5) Asymmetric RV built with two semaphores (master/slave); (6) Fork-join (or par-begin/par-end); (7) Non-recursive sub-program (places i and j cannot be simultaneously marked

– must be in mutex – to remember the returning point; for simplicity, it is assumed that the subprogram is single input/single output); (8) Guard (a self-loop from a place through a transition); its role is like a traffic light. If at least one token is present at the place, it allows the evolution, but it is not consumed. Synchronizations can also be modeled by the weights associated with the arcs going to transitions

does. If the new system is easier to analyze, and the transformation is computationally cheap, the process may be extremely interesting; (3) *structural*, based on graph properties, or in mathematical programming techniques rooted on the state equation; (4) *simulation*, particularly interesting for gaining certain confidence about the absence of certain pathological behaviors. Analysis strategies combining all these kinds of techniques are extremely useful in practice.

Reachability techniques provide sequentialized views for a particular initial marking.

Moreover they suffer from the so-called *state explosion* problem. The reduction of this computational issue leads to techniques such as *stubborn set* methods (smaller, equally informative reachability graphs), also to non-sequentialized views such as those based on *unfoldings*. Transformation techniques are extremely useful, but not complete (i.e., not all net systems can be reduced in a practical way). In most cases, structural techniques only provide necessary or sufficient conditions (e.g., a sufficient condition for deadlock-freeness, a

necessary condition for reversibility, etc.), but not a full characterization. As already pointed out, a limitation of methods based on the state equation for analyzing net systems is the existence of non-reachable solutions (the so-called spurious solutions). In this context, three kinds of related notions that must be differentiated are the following: (1) some natural *vectors* (left and right annullers of the token flow matrix, **C**: *P-semiflows* and *T-semiflows*), (2) some invariant *laws* (*token conservation* and *repetitive behaviors*), and (3) some peculiar *subnets* (*conservative* and *consistent* components, generated by the subsets of nodes in the P- and T-semiflows, respectively).

More than analysis, *control* leads to synthesis problems. The idea is to *enforce* the given system in order to fulfill a specification (e.g., to enforce certain mutual exclusion properties). Technically speaking, the idea is to “add” some elements in order to constrain the behavior in such a way that a correct execution is obtained. Questions related to control, observation, diagnosis, or identification are all areas of ongoing research.

With respect to classical control theory, there are two main differences: Models are DES and untimed (autonomous, fully nondeterministic, eventually labeling the transitions in order to be able to consider the PN languages). Let us remark that for control purposes, the transitions should be partitioned into *controllable* (when enabled, you can either force or block the firing) and *uncontrollable* (if enabled, the firing is nondeterministic). A natural approach to synthesize a control is to start modeling the plant dynamics (by means of a PN, **P**) and adopting a specification for the desired closed-loop system (**S**). The goal is to compute a controller (**L**) such that **S** equals the parallel-composition of **P** and **L**; in other words, controllers (called “supervisors”) are designed to ensure that only behaviors consistent with the specification may occur. The previous equality is not always possible, and the goal is usually relaxed to minimally limit the behavior within the specified legality (i.e., to compute *maximally permissive* controllers). For an approach in the framework of finite-state machines and

regular languages, see ► [“Supervisory Control of Discrete-Event Systems”](#). The synthesis in the framework of Petri nets and having goals as enforcing some generalized mutual exclusions constraints in markings or avoiding deadlocks, for example, can be efficiently approached by means of the so named *structure theory*, based on the direct exploitation of the structure of the net model (using graph or mathematical programming theories and algorithms, where the initial marking is a parameter).

Similarly, transitions (or places) can be partitioned into *observable* and *unobservable*. Many *observability* problems may be of interest, for example, observing the firing of a subset of transitions to compute the subset of markings in which the system may be. Related to observability, *diagnosis* is the process of detecting a failure (any deviation of a system from its intended behavior) and identifying the cause of the abnormality. *Diagnosability*, like observability or controllability, is a logical criterion. If a model is diagnosable with respect to a certain subset of possible faults (i.e., it is possible to detect the occurrence of those faults in finite time), a diagnoser can be constructed (see section “Diagnosis and Diagnosability Analysis of Petri Nets” in ► [“Diagnosis of Discrete Event Systems”](#)). *Identification* of DES is also a question that has required attention in recent years. In general, the starting point is a behavioral observation, the goal being to construct a PN model that generates the observed behavior, either from examples/counterexamples of its language or from the structure of a reachability graph. So the results are *derived* models, not human-made models (i.e., not made by designers).

The Petri Nets Modeling Paradigm

Along the *life cycle* of DES, designers may deal with basic modeling, analysis, and synthesis from different perspectives together with implementation and operation issues. Thus, the designer may be interested in expressing the

basic structure, understanding untimed possible behaviors, checking logic properties on the model when provided with some timing (e.g., in order to guarantee if a certain reaction is possible before 3 ms; something relevant in real-time systems), computing some performance indices on timed models (related to the throughput in the firing of a given transition, or to the length of a waiting line, expressed as the number of tokens in a place), computing a schedule or control that optimizes a certain objective function, decomposing the model in order to prepare an efficient implementation, efficiently determining redundancies in order to increase the degree of fault tolerance, etc. For these different tasks, different formalisms may be used. Nevertheless, it seems desirable to have a *family* of related formalisms rather than a collection of “unrelated” or weakly related formalisms. The expected advantages would include *coherence* among models usable in different phases, *economy* in the transformations, and *synergy* in the development of models and theories.

Other Untimed PN Formalisms: Levels of Expressive Power

PT-net systems are more powerful than *condition/event* (CE) systems, roughly speaking the basic seminal formalism of Carl Adam Petri in which places can be marked only with zero or one token (Boolean marking). CE-systems can model only finite-state systems. As already said, “extensions” of the expressive power of untimed PT-net systems to the level of Turing machines are obtained by adding inhibitor arcs or priorities to the firing of transitions.

An important idea is adding the notion of *individuals* to tokens (e.g., from anonymous to labeled or colored tokens). Information in tokens allows the objects to be named (they are no longer indistinguishable) and dynamic associations to be created. Moving from PT-nets to so-called high-level PNs (HLPNs) is something like “moving from assembler to high-level programming languages” or, at the computational level, like “moving from pure numerical to a symbolic level.” There are many proposals in this respect, the

more important being predicate/transition nets and colored PNs. Sometimes, this type of abstraction has the same theoretical expressiveness as PT-net systems (e.g., colored PNs if the number of colors is finite); in other words, high-level views may lead to more compact and structured models while keeping the same theoretical expressive power of PT-nets (i.e., we can speak of “abbreviations,” not of “extensions”). In other cases, *object-oriented* concepts from computer programming are included in certain HLPNs. The analysis techniques of HLPNs can be approached with techniques based on enumeration, transformation, or structural considerations and simulation, generalizing those developed for PT-net systems.

Extending Net Systems with External Events and Time: Nonautonomous Formalisms

When dealing with net systems that interact with some specific environment, the marking evolution rule must be slightly modified. This can be done in an enormous number of ways, considering external events and logical conditions as inputs to the net model, in particular some depending on time. The same interpretation given to a graph in order to define a *finite-state diagram* can be used to define a *marking diagram*, a formalism in which the key point is to recognize that the state is now numerical (for PT-net systems) and distributed. For example, drawing a parallel with Moore automata, the transitions should be labeled with logical conditions and events, while unconditional actions are associated with the places. If a place is marked, the associated actions are emitted.

Even if only *time-based interpretations* are considered, there are a large number of successful proposals for formalisms. For example, it should be specified if time is associated with the firing of transitions (T-timing may be atomic or in three phases), the residence of tokens in places (P-timed), and the arcs of the net, as tags to the tokens, etc. Moreover, even for T-timed models, there are many ways of defining the timing: *time intervals*, *stochastic* or *possibilistic* forms, and the *deterministic* case as a particular one. If the

firing of transitions follows *exponential* pdfs, and the conflict resolution follows the *race* policy (i.e., fire in conflicts the first that ends the task, not a *preselection* policy), the underlying Markov chain described is isomorphic to the reachability graph (due to the Markovian “memoryless” property). Moreover, the addition of *immediate* transitions (whose firing is instantaneous) enriches the practical modeling possibilities, eventually complicating the analysis techniques. Timed models are used to compute minimum and maximum time delays (when time intervals are provided, in real-time problems) or performance figures (throughput, utilization rate of resources, average number of tokens – clients – in services, etc.). For performance evaluation, there is an array of techniques to compute bounds, approximated values, or exact values, sometimes generalizing those that are used in certain queueing network classes of models. Simulation techniques are frequently very helpful in practice to produce an *educated guess* about the expected performance. Time constraints on Petri nets may change logical properties of models (e.g., mutual exclusion constraints, deadlock-freeness, etc.), calling for new analysis techniques. For example, certain timings on transitions can transform a live system into a non-live one (if to the net system in Fig. 2 are associated deterministic times to transitions and a race policy with the time associated with transition *c* smaller than that of transition *a*, transition *b* cannot be fired, after firing transition *d*; thus it is non-live, while the untimed model was live). By the addition of some time constraints, the transformation of a non-live model into a live one is also possible. So additional analysis techniques need to be considered, redefining the state, now depending also on time, more than just on the marking.

Finally, as in any DES, the optimal control of timed Petri net models (scheduling, sequencing, etc.) may be approached by techniques as dynamic programming or perturbation analysis (presented in the context of queueing networks and Markov chains, see ► [“Perturbation Analysis of Discrete Event Systems”](#)). In practice, those problems are frequently approached by means of some partially heuristic strategies. About the

diagnosis of timed Petri nets, see ► [“Diagnosis of Discrete Event Systems”](#). Of course, all these tasks can be done with HLPNs.

Fluid and Hybrid PN Models

Different ideas may lead to different kinds of *hybrid* PNs. One is to *fluidize* (here to relax the natural numbers of discrete markings into the nonnegative reals) the firing of transitions that are “most time” enabled. Then the relaxed model has *discrete* and *continuous* transitions, thus also *discrete* and *continuous* places. If all transitions are fluidized, the PN system is said to be *fluid* or *continuous*, even if technically it is a hybrid one. In this approach, the main goal is to try to overcome the *state explosion* problem inherent to enumeration techniques. Proceeding in that way, some computationally NP-hard problems may become much easier to solve, eventually in polynomial time. In other words, *fluidization* is an abstraction that tries to make tractable certain real-scale DES problems (► [“Discrete Event Systems and Hybrid Systems, Connections Between”](#)).

When transitions are timed with the so-called *infinite server* semantics, the PN system can be observed as a time differentiable *piecewise affine* system. Thus, even if the relaxation “simplifies” computations, it should be taken into account that continuous PNs with infinite server semantics are able to simulate Turing machines. From a different perspective, the steady-state throughput of a given transition may be non-monotonic with respect to the firing rates or the initial marking (e.g., if faster or more machines are used, the uncontrolled system may be slower); moreover, due to the important expressive power, *discontinuities* may even appear with respect to continuous design parameters as firing rates, for example.

An alternative way to define hybrid Petri nets is a generalization of *hybrid automata*: The net system is a DES, but sets of differential equations are associated to the marking of places. If a place is marked, the corresponding differential equations contribute to define its evolution.

Summary and Future Directions

Petri nets designate a broad family of related DES formalisms (a modeling paradigm) each one specifically tailored to approach certain problems. Conceptual simplicity coexists with powerful modeling, analysis, and synthesis capabilities. From a control theory perspective, much work remains to be done for both untimed and timed formalisms (remember, there are many different ways of timing), particularly when dealing with optimal control of timed models. In engineering practice, approaches to the latter class of problems frequently use heuristic strategies. From a broader perspective, future research directions include improvements required to deal with controllability and the design of controllers, with observability and the design of observers, with diagnosability and the design of diagnosers, and with identification. This work is not limited to the strict DES framework, but also applies to analogous problems relating to relaxations into *hybrid* or *fluid* approximations (particularly useful when high populations are considered). The distributed nature of system is more and more frequent and is introducing new constraints, a subject requiring serious attention. In all cases, different from firing languages approaches, the so named *structure theory* of Petri nets should gain more interest.

Cross-References

- [Applications of Discrete Event Systems](#)
- [Diagnosis of Discrete Event Systems](#)
- [Discrete Event Systems and Hybrid Systems, Connections Between](#)
- [Models for Discrete Event Systems: An Overview](#)
- [Perturbation Analysis of Discrete Event Systems](#)
- [Supervisory Control of Discrete-Event Systems](#)

Recommended Reading

Topics related to PNs are considered in well over a hundred thousand papers and reports.

The first generation of books concerning this field is Brauer (1980), immediately followed by Starke (1980), Peterson (1981), Brams (1983), Reisig (1985), and Silva (1985). The fact that they are written in English, French, German, and Spanish is proof of the rapid dissemination of this knowledge. Most of these books deal essentially with PT-net systems. Complementary surveys are Murata (1989), Silva (1993), and David and Alla (1994), the latter also considering some continuous and hybrid models. Concerning high-level PNs, Jensen and Rozenberg (1991) is a selection of papers covering the main developments during the 1980s. Jensen and Kristensen (2009) focus on state space methods and simulation where elements of timed models are taken into account, but performance evaluation of stochastic systems is not covered. Approaching the present days, relevant works written with complementary perspectives include inter alia, Girault and Valk (2003), Diaz (2009), David and Alla (2010), and Seatzu et al. (2013). The consideration of time in nets with an emphasis on performance and performability evaluation is addressed in monographs such as Ajmone Marsan et al. (1995), Bause and Kritzinger (1996), Balbo and Silva (1998), and Haas (2002), while timed models under different fuzzy interpretations are the subject of Cardoso and Camargo (1999). Structure-based approaches to control PN models is the main subject in Iordache and Antsaklis (2006) or Chen and Li (2013). Different kinds of hybrid PN models are studied in Di Febbraro et al. (2001), Villani et al. (2007), and David and Alla (2010), while a broad perspective about modeling, analysis, and control of fluid (or continuous) – untimed and timed – PNs is provided by Silva et al. (2011).

DiCesare et al. (1993) and Desrochers and Al-Jaar (1995) are devoted to the applications of PNs to manufacturing systems. A comprehensive updated introduction to business process systems and PNs can be found in van der Aalst and Stahl (2011). Special volumes dealing with other monographic topics are, for example, Billington et al. (1999), Agha et al. (2001), and Cortadella et al. (2002). An application domain for Petri nets emerging over the last two decades is

systems biology, a model-based approach devoted to the analysis of biological systems (Wingender 2011; Koch et al. 2011). Furthermore, it should be pointed out that Petri nets have also been employed in many other application domains (e.g., from logistics to musical systems).

For an overall perspective of the field over the five decades that have elapsed since the publication of Carl Adam Petri's PhD thesis, including historical, epistemological, and technical aspects, see Silva (2013). An overview of the historical development of the field of Petri nets (PNs) from a Systems Theory and Automatic Control perspective is provided in Giua and Silva (2018). Intentionally not meant to be comprehensive, it outlines, through selected representative topics, some of the conceptual issues studied in the literature.

Bibliography

- Agha G, de Cindio F, Rozenberg G (eds) (2001) Concurrent object-oriented programming and Petri nets, advances in Petri nets. Volume 2001 of LNCS. Springer, Berlin/Heidelberg/New York
- Balbo G, Silva M (eds) (1998) Performance models for discrete event systems with synchronizations: formalisms and analysis techniques. In: Proceedings of human capital and mobility MATCH performance advanced school, Jaca. Available online: <http://webdiis.unizar.es/GISED/?q=news/matchbook>
- Bause F, Kritzinger P (1996) Stochastic Petri nets. An introduction to the theory. Vieweg, Braunschweig
- Billington J, Diaz M, Rozenberg G (eds) (1999) Application of Petri nets to communication networks, advances in Petri nets. Volume 1605 of LNCS. Springer, Berlin/Heidelberg/New York
- Brams GW (1983) *Reseaux de Petri: Theorie et Pratique*. Masson, Paris
- Brauer W (ed) (1980) Net theory and applications. Volume 84 of LNCS. Springer, Berlin/New York
- Cardoso J, Camargo H (eds) (1999) Fuzziness in Petri nets. Volume 22 of studies in fuzziness and soft computing. Physica-Verlag, Heidelberg/New York
- Chen Y, Li Z (2013) Optimal supervisory control of automated manufacturing systems. CRC, Boca Raton
- Cortadella J, Yakovlev A, Rozenberg G (eds) (2002) Concurrency and hardware design, advances in Petri nets. Volume 2549 of LNCS. Springer, Berlin/Heidelberg/New York
- David R, Alla H (1994) Petri nets for modeling of dynamic systems – a survey. *Automatica* 30(2):175–202
- David R, Alla H (2010) Discrete, continuous and hybrid Petri nets. Springer, Berlin/Heidelberg
- Desrochers A, Al-Jaar RY (1995) Applications of Petri nets in manufacturing systems. IEEE, New York
- Diaz M (ed) (2009) Petri nets: fundamental models, verification and applications. Control systems, robotics and manufacturing series (CAM). Wiley, London
- DiCesare F, Harhalakis G, Proth JM, Silva M, Vernadat FB (1993) Practice of Petri nets in manufacturing. Chapman & Hall, London/Glasgow/New York
- Di Febraro A, Giua A, Menga G (eds) (2001) Special issue on hybrid Petri nets. *Discret Event Dyn Syst* 11(1–2):5–185
- Girault C, Valk R (2003) Petri nets for systems engineering. A guide to modeling, verification, and applications. Springer, Berlin
- Giua A, Silva M (2018) Petri nets and automatic control: a historical perspective. *Annu Rev Control* 45: 223–239
- Haas PJ (2002) Stochastic Petri nets. modelling, stability, simulation. Springer series in operations research. Springer, New York
- Iordache MV, Antsaklis PJ (2006) Supervisory control of concurrent systems: a Petri net structural approach. Birkhauser, Boston
- Jensen K, Kristensen LM (2009) Coloured Petri nets. modelling and validation of concurrent systems. Springer, Berlin
- Jensen K, Rozenberg G (eds) (1991) High-level Petri nets. Springer, Berlin
- Koch I, Reisig W, Schreiber F (eds) (2011) Modeling in systems biology. the Petri net approach. Computational biology, vol 16. Springer, Berlin
- Marsan MA, Balbo G, Conte G, Donatelli S, Franceschinis G (1995) Modelling with generalized stochastic Petri nets. Wiley, Chichester/New York
- Murata T (1989) Petri nets: properties, analysis and applications. *Proc IEEE* 77(4):541–580
- Peterson JL (1981) Petri net theory and the modeling of systems. Prentice-Hall, Upper Saddle River
- Petri CA (1966) Communication with automata. Rome Air Development Center-TR-65-377, New York
- Reisig W (1985) Petri nets. An introduction. Volume 4 of EATCS monographs on theoretical computer science. Springer, Berlin/Heidelberg/New York
- Seatzu C, Silva M, Schuppen J (eds) (2013) Control of discrete-event systems. Automata and Petri net perspectives. Number 433 in lecture notes in control and information sciences. Springer, London
- Silva M (1985) Las Redes de Petri: en la Automatica y la Informatica. Madrid AC (ed) (2nd ed, Thomson-AC, 2002)
- Silva M (1993) Introducing Petri nets. In: Practice of Petri nets in manufacturing. Chapman and Hall, London/New York, pp 1–62
- Silva M (2013) Half a century after Carl Adam Petri's Ph.D. thesis: a perspective on the field. *Annu Rev Control* 37(2):191–219
- Silva M, Teruel E, Colom JM (1998) Linear algebraic and linear programming techniques for the analysis of net

- systems. Volume 1491 of LNCS, advances in Petri nets. Springer, Berlin/Heidelberg/New York, pp 309–373
- Silva M, Julvez J, Mahulea C, Vazquez C (2011) On fluidization of discrete event models: observation and control of continuous Petri nets. *Discret Event Dyn Syst* 21:427–497
- Starke P (1980) *Petri-Netze*. Deutscher Verlag der Wissenschaften, Berlin
- van der Aalst W, Stahl C (2011) *Modeling business processes: a Petri net oriented approach*. MIT Press, Cambridge
- Villani E, Miyagi PE, Valette R (2007) *Modelling and analysis of hybrid supervisory systems. A Petri net approach*. Springer, Berlin
- Wingender E (ed) (2011) *Biological Petri nets. Studies in health technology and informatics. vol 162*. IOS Press, Lansdale

Models for Discrete Event Systems: An Overview

Christos G. Cassandras
Division of Systems Engineering, Center for
Information and Systems Engineering, Boston
University, Brookline, MA, USA

Abstract

This article provides an introduction to discrete event systems (DES) as a class of dynamic systems with characteristics significantly distinguishing them from traditional time-driven systems. It also overviews the main modeling frameworks used to formally describe the operation of DES and to study problems related to their control and optimization.

Keywords

Automata · Dioid algebras · Event-driven systems · Hybrid systems · Petri nets · Time-driven systems

Synonyms

DES

Introduction

Discrete event systems (DES) form an important class of dynamic systems. The term was introduced in the early 1980s to describe a DES in terms of its most critical feature: the fact that its behavior is governed by *discrete events* which occur asynchronously over time and which are solely responsible for generating state transitions. In between event occurrences, the state of a DES is unaffected. Examples of such behavior abound in technological environments, including computer and communication networks, manufacturing systems, transportation systems, logistics, and so forth. The operation of a DES is largely regulated by rules which are often unstructured and frequently human-made, as in initiating or terminating activities and scheduling the use of resources through controlled events (e.g., turning equipment “on”). On the other hand, their operation is also subject to uncontrolled randomly occurring events (e.g., a spontaneous equipment failure) which may or may not be observable through sensors. It is worth pointing out that the term “discrete event dynamic system” (DEDS) is also commonly used to emphasize the importance of the dynamical behavior of such systems (Ho 1991; Cassandras and Lafortune 2008).

There are two aspects of a DES that define its behavior:

1. The variables involved are both continuous and discrete, sometimes purely symbolic, i.e., nonnumeric (e.g., describing the state of a traffic light as “red” or “green”). This renders traditional mathematical models based on differential (or difference) equations inadequate and related methods based on calculus of limited use.
2. Because of the asynchronous nature of events that cause state transitions in a DES, it is neither natural nor efficient to use time as a synchronizing element driving its dynamics. It is for this reason that DES are often referred to as *event driven*, to contrast them to classical *time-driven* systems based on the laws

of physics; in the latter, as time evolves state, variables such as position, velocity, temperature, voltage, etc., also continuously evolve. In order to capture event-driven state dynamics, however, different mathematical models are necessary.

In addition, uncertainties are inherent in the technological environments where DES are encountered. Therefore, associated mathematical models and methods for analysis and control must incorporate such uncertainties. Finally, complexity is also inherent in DES of practical interest, usually manifesting itself in the form of combinatorially explosive state spaces. Although purely analytical methods for DES design, analysis, and control are limited, they have still enabled reliable approximations of their dynamic behavior and the derivation of useful structural properties and provable performance guarantees. Much of the progress made in this field, however, has relied on new paradigms characterized by a combination of mathematical techniques, computer-based tools, and effective processing of experimental data.

Event-driven and time-driven system components are often viewed as coexisting and interacting and are referred to as *hybrid systems* (separately considered in the Encyclopedia, including the article ► [Discrete Event Systems and Hybrid Systems, Connections Between](#)). Arguably, most contemporary technological systems are combinations of time-driven components (typically, the physical parts of a system) and event-driven components (usually, the computer-based controllers that collect data from and issue commands to the physical parts).

Event-Driven vs. Time-Driven Systems

In order to explain the difference between time-driven and event-driven behavior, we begin with the concept of “event.” An event should be thought of as occurring instantaneously and causing transitions from one system state value to another. It may be identified with an action

(e.g., pressing a button), a spontaneous natural occurrence (e.g., a random equipment failure), or the result of conditions met by the system state (e.g., the fluid level in a tank exceeds a given value). For the purpose of developing a model for DES, we will use the symbol e to denote an event. Since a system is generally affected by different types of events, we assume that we can define a discrete *event set* E with $e \in E$.

In a classical system model, the “clock” is what drives a typical state trajectory: with every “clock tick” (which may be thought of as an “event”), the state is expected to change, since continuous state variables continuously change with time. This leads to the term *time driven*. In the case of time-driven systems described by continuous variables, the field of systems and control has based much of its success on the use of well-known differential-equation-based models, such as

$$\mathbf{R}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (1)$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t), \quad (2)$$

where (1) is a (vector) state equation with initial conditions specified and (2) is a (vector) output equation. As is common in system theory, $\mathbf{x}(t)$ denotes the state of the system, $\mathbf{y}(t)$ is the output, and $\mathbf{u}(t)$ represents the input, often associated with controllable variables used to manipulate the state so as to attain a desired output. Common physical quantities such as position, velocity, temperature, pressure, flow, etc., define state variables in (1). The state generally changes as time changes, and, as a result, the time variable t (or some integer $k = 0, 1, 2, \dots$ in discrete time) is a natural independent variable for modeling such systems.

In contrast, in a DES, time no longer serves the purpose of driving such a system and may no longer be an appropriate independent variable. Instead, at least some of the state variables are discrete, and their values change only at certain points in time through instantaneous transitions which we associate with “events.” If a clock is used, consider two possibilities: (i) At every clock tick, an event e is selected from

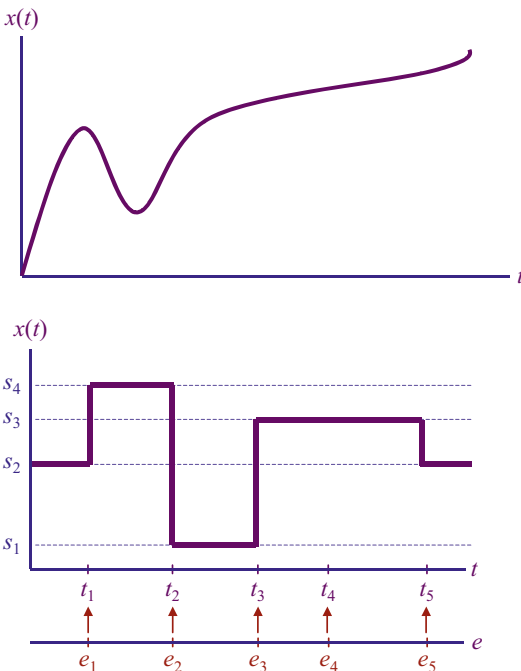
the event set E (if no event takes place, we use a “null event” as a member of E such that it causes no state change), and (ii) at various time instants (not necessarily known in advance or coinciding with clock ticks), some event e “announces” that it is occurring. Observe that in (i) state transitions are *synchronized* by the clock which is solely responsible for any possible state transition. In (ii), every event $e \in E$ defines a distinct process through which the time instants when e occurs are determined. State transitions are the result of combining these *asynchronous* concurrent event processes. Moreover, these processes need not be independent of each other. The distinction between (i) and (ii) gives rise to the terms *time-driven* and *event-driven* systems, respectively.

Comparing state trajectories of time-driven and event-driven systems is useful in understanding the differences between the two and setting the stage for DES modeling frameworks. Thus, in Fig. 1, we observe the following: (i) For the

time-driven system shown, the state space X is the set of real numbers $\mathbb{R}$, and $x(t)$ can take any value from this set. The function $x(t)$ is the solution of a differential equation of the general form $\dot{x}(t) = f(x(t), u(t), t)$, where $u(t)$ is the input. (ii) For the event-driven system, the state space is some discrete set $X = \{s_1, s_2, s_3, s_4\}$. The sample path can only jump from one state to another whenever an event occurs. Note that an event may take place, but not cause a state transition, as in the case of e_4 . There is no immediately obvious analog to $\dot{x}(t) = f(x(t), u(t), t)$, i.e., no mechanism to specify how events might interact over time or how their time of occurrence might be determined. Thus, a large part of the early developments in the DES field has been devoted to the specification of an appropriate mathematical model containing the same expressive power as (1)–(2) (Cassandras and Lafortune 2008; Baccelli et al. 1992; Glasserman and Yao 1994).

We should point out that a time-driven system with continuous state variables, usually modeled through (1)–(2), may be abstracted as a DES through some form of discretization in time and quantization in the state space. We should also point out that discrete *event* systems should not be confused with discrete *time* systems. The class of discrete time systems contains both time-driven and event-driven systems.

M



Models for Discrete Event Systems: An Overview,
Fig. 1 Comparison of time-driven and event-driven state trajectories

Timed and Untimed Models of Discrete Event Systems

Returning to Fig. 1, instead of constructing the piecewise constant function $x(t)$ as shown, it is convenient to simply write the timed sequence of events $\{(e_1, t_1), (e_2, t_2), (e_3, t_3), (e_4, t_4), (e_5, t_5)\}$ which contains the same information as the state trajectory. Assuming that the initial state of the system (s_2 in this case) is known and that the system is “deterministic” in the sense that the next state after the occurrence of an event is unique, we can recover the state of the system at any point in time and reconstruct the DES state trajectory. The set of all possible timed sequences of events that a given system

can ever execute is called the *timed language* model of the system. The word “language” comes from the fact that we can think of the event E as an “alphabet” and of (finite) sequences of events as “words” (Hopcroft and Ullman 1979). We can further refine such a model by adding statistical information regarding the set of state trajectories (sample paths) of the system. Let us assume that probability distribution functions are available about the “lifetime” of each event type $e \in E$, that is, the elapsed time between successive occurrences of this particular e . A *stochastic timed language* is a timed language together with associated probability distribution functions for the events.

Stochastic timed language modeling is the most detailed in the sense that it contains event information in the form of event occurrences and their orderings, information about the exact times at which the events occur (not only their relative ordering), and statistical information about successive occurrences of events. If we delete the timing information from a timed language, we obtain an *untimed language*, or simply *language*, which is the set of all possible orderings of events that could happen in the given system. For example, the untimed sequence corresponding to the timed sequence of events in Fig. 1 is $\{e_1, e_2, e_3, e_4, e_5\}$.

Untimed and timed languages represent different levels of abstraction at which DES are modeled and studied. The choice of the appropriate level of abstraction clearly depends on the objectives of the analysis. In many instances, we are interested in the “logical behavior” of the system, that is, in ensuring that all the event sequences it can generate satisfy a given set of specifications, e.g., maintaining a precise ordering of events. In this context, the actual timing of events is not required, and it is sufficient to model only the untimed behavior of the system. *Supervisory control* that is discussed in the article ► [Supervisory Control of Discrete-Event Systems](#) is the term established for describing the systematic means (i.e., enabling or disabling events which are controllable) by which the logical behavior of a DES is regulated to achieve

a given specification (Cassandras and Lafortune 2008; Ramadge and Wonham 1987; Moody and Antsaklis 1998).

On the other hand, we may be interested in *event timing* in order to answer questions such as the following: “How much time does the system spend at a particular state?” or “Can this sequence of events be completed by a particular deadline?” More generally, event timing is important in assessing the performance of a DES often measured through quantities such as *throughput* or *response time*. In these instances, we need to consider the timed language model of the system. Since DES frequently operate in a stochastic setting, an additional level of complexity is introduced, necessitating the development of probabilistic models and related analytical methodologies for design and performance analysis based on stochastic timed language models. *Sample path analysis* and *perturbation analysis*, discussed in the entry ► [Perturbation Analysis of Discrete Event Systems](#), refer to the study of sample paths of DES, focusing on the extraction of information for the purpose of efficiently estimating performance sensitivities of the system and, ultimately, achieving online control and optimization (Cassandras and Lafortune 2008; Ho and Cassandras 1983; Ho and Cao 1991; Glasserman 1991).

These different levels of abstraction are complementary, as they address different issues about the behavior of a DES. Although the language-based approach to DES modeling is attractive, it is by itself not convenient to address verification, controller synthesis, or performance issues. This motivates the development of *discrete event modeling formalisms* which represent languages in a manner that highlights structural information about the system behavior and can be used to address analysis and controller synthesis issues. Next, we provide an overview of three major modeling formalisms which are used by most (but not all) system and control theoretic methodologies pertaining to DES. Additional modeling formalisms encountered in the computer science literature include process algebras (Baeten and Weijland 1990) and communicating sequential processes (Hoare 1985).

Automata

A *deterministic automaton*, denoted by G , is a six-tuple

$$G = (\mathcal{X}, \mathcal{E}, f, \Gamma, x_0, \mathcal{X}_m),$$

where $\mathcal{X}$ is the set of *states*, $\mathcal{E}$ is the finite set of *events* associated with the transitions in G , and $f : \mathcal{X} \times \mathcal{E} \rightarrow \mathcal{X}$ is the *transition function*; specifically, $f(x, e) = y$ means that there is a transition labeled by event e from state x to state y and, in general, f is a *partial function* on its domain. $\Gamma : \mathcal{X} \rightarrow 2^{\mathcal{E}}$ is the *active event function* (or feasible event function); $\Gamma(x)$ is the set of all events e for which $f(x, e)$ is defined and it is called the *active event set* (or feasible event set) of G at x . Finally, x_0 is the *initial state* and $\mathcal{X}_m \subseteq \mathcal{X}$ is the set of *marked states*. The terms *state machine* and *generator* (which explains the notation G) are also used to describe the above object. Moreover, if $\mathcal{X}$ is a finite set, we call G a *deterministic finite-state automaton*. A *nondeterministic automaton* is defined by means of a relation over $\mathcal{X} \times \mathcal{E} \times \mathcal{X}$ or, equivalently, a function from $\mathcal{X} \times \mathcal{E}$ to $2^{\mathcal{X}}$.

The automaton G operates as follows. It starts in the initial state x_0 , and upon the occurrence of an event $e \in \Gamma(x_0) \subseteq \mathcal{E}$, it makes a transition to state $f(x_0, e) \in \mathcal{X}$. This process then continues based on the transitions for which f is defined. Note that an event may occur without changing the state, i.e., $f(x, e) = x$. It is also possible that two distinct events occur at a given state causing the exact same transition, i.e., for $a, b \in \mathcal{E}$, $f(x, a) = f(x, b) = y$. What is interesting about the latter fact is that we may not be able to distinguish between events a and b by simply observing a transition from state x to state y .

For the sake of convenience, f is always extended from domain $\mathcal{X} \times \mathcal{E}$ to domain $\mathcal{X} \times \mathcal{E}^*$, where $\mathcal{E}^*$ is the set of *all* finite strings of elements of $\mathcal{E}$, including the empty string (denoted by ε); the $*$ operation is called the *Kleene closure*. This is accomplished in the following recursive manner: $f(x, \varepsilon) := x$ and $f(x, se) := f(f(x, s), e)$ for $s \in \mathcal{E}^*$ and $e \in \mathcal{E}$. The (untimed) language generated by G and

denoted by $\mathcal{L}(G)$ is the set of all strings in $\mathcal{E}^*$ for which the extended function f is defined. The automaton model above is one instance of what is referred to as a *generalized semi-Markov scheme* (GSMS) in the literature of stochastic processes. A GSMS is viewed as the basis for extending automata to incorporate an event timing structure as well as nondeterministic state transition mechanisms, ultimately leading to *stochastic timed automata*, discussed in the sequel.

Let us allow for generally countable sets $\mathcal{X}$ and $\mathcal{E}$ and leave out of the definition any consideration for marked states. Thus, we begin with an automaton model $(\mathcal{X}, \mathcal{E}, f, \Gamma, x_0)$. We extend the modeling setting to *timed automata* by incorporating a “clock structure” associated with the event set $\mathcal{E}$ which now becomes the input from which a specific event sequence can be deduced. The *clock structure* (or *timing structure*) associated with an event set $\mathcal{E}$ is a set $\mathbf{V} = \{\mathbf{v}_i : i \in \mathcal{E}\}$ of clock (or lifetime) sequences

$$\mathbf{v}_i = \{v_{i,1}, v_{i,2}, \dots\}, i \in \mathcal{E}, v_{i,k} \in \mathbb{R}^+, k = 1, 2, \dots$$

M

Timed Automaton. A *timed automaton* is defined as a six-tuple

$$(\mathcal{X}, \mathcal{E}, f, \Gamma, x_0, \mathbf{V}),$$

where $\mathbf{V} = \{\mathbf{v}_i : i \in \mathcal{E}\}$ is a clock structure and $(\mathcal{X}, \mathcal{E}, f, \Gamma, x_0)$ is an automaton. The automaton generates a state sequence $x' = f(x, e')$ driven by an event sequence $\{e_1, e_2, \dots\}$ generated through

$$e' = \arg \min_{i \in \Gamma(x)} \{y_i\} \quad (3)$$

with the clock values $y_i, i \in \mathcal{E}$, defined by

$$y'_i = \begin{cases} y_i - y^* & \text{if } i \neq e' \text{ and } i \in \Gamma(x) \\ v_{i, N_i+1} & \text{if } i = e' \text{ or } i \notin \Gamma(x) \end{cases} \quad i \in \Gamma(x') \quad (4)$$

where the *interevent time* y^* is defined as

$$y^* = \min_{i \in \Gamma(x)} \{y_i\} \quad (5)$$

and the *event scores* $N_i, i \in \mathcal{E}$, are defined by

$$N'_i = \begin{cases} N_i + 1 & \text{if } i = e' \text{ or } i \notin \Gamma(x) \\ N_i & \text{otherwise} \end{cases} \quad i \in \Gamma(x'). \quad (6)$$

In addition, initial conditions are $y_i = v_{i,1}$ and $N_i = 1$ for all $i \in \Gamma(x_0)$. If $i \notin \Gamma(x_0)$, then y_i is undefined and $N_i = 0$.

A simple interpretation of this elaborate definition is as follows. Given that the system is at some state x , the next event e' is the one with the smallest clock value among all feasible events $i \in \Gamma(x)$. The corresponding clock value, y^* , is the interevent time between the occurrence of e and e' , and it provides the amount by which the time, t , moves forward: $t' = t + y^*$. Clock values for all events that remain active in state x' are decremented by y^* , except for the triggering event e' and all newly activated events, which are assigned a new lifetime v_{i,N_i+1} . Event scores are incremented whenever a new lifetime is assigned to them. It is important to note that the “system clock” t is fully controlled by the occurrence of events, which cause it to move forward; if no event occurs, the system remains at the last state observed.

Comparing $x' = f(x, e')$ to the state equation (1) for time-driven systems, we see that the former can be viewed as the event-driven analog of the latter. However, the simplicity of $x' = f(x, e')$ is deceptive: unless an event sequence is given, determining the *triggering* event e' which is required to obtain the next state x' involves the combination of (3)–(6). Therefore, the analog of (1) as a “canonical” state equation for a DES requires all Eqs. (3)–(6). Observe that this timed automaton generates a timed language, thus extending the untimed language generated by the original automaton G .

In the definition above, the clock structure $\mathbf{V}$ is assumed to be fully specified in a deterministic sense and so are state transitions dictated by $x' = f(x, e')$. The sequences $\{\mathbf{v}_i\}, i \in \mathcal{E}$, can be extended to be specified only as stochastic sequences through distribution functions denoted by $F_i, i \in \mathcal{E}$. Thus, the *stochastic clock structure* (or *stochastic timing structure*) associated with an event set $\mathcal{E}$ is a set of distribution functions

$F = \{F_i : i \in \mathcal{E}\}$ characterizing the stochastic clock sequences

$$\{V_{i,k}\} = \{V_{i,1}, V_{i,2}, \dots\}, \quad i \in \mathcal{E}, \\ V_{i,k} \in \mathbb{R}^+, \quad k = 1, 2, \dots$$

It is usually assumed that each clock sequence consists of random variables which are independent and identically distributed (iid) and that all clock sequences are mutually independent. Thus, each $\{V_{i,k}\}$ is completely characterized by a distribution function $F_i(t) = P[V_i \leq t]$. There are, however, several ways in which a clock structure can be extended to include situations where elements of a sequence $\{V_{i,k}\}$ are correlated or two clock sequences are interdependent. As for state transitions which may be nondeterministic in nature, such behavior is modeled through state transition probabilities as explained next.

Stochastic Timed Automaton. We can extend the definition of a timed automaton by viewing the state, event, and all event scores and clock values as random variables denoted respectively by capital letters X, E, N_i , and $Y_i, i \in \mathcal{E}$. Thus, a *stochastic timed automaton* is a six-tuple

$$(\mathcal{X}, \mathcal{E}, \Gamma, p, p_0, F),$$

where $\mathcal{X}$ is a countable *state space*; $\mathcal{E}$ is a countable *event set*; $\Gamma(x)$ is the *active event set* (or feasible event set); $p(x'; x, e')$ is a *state transition probability* defined for all $x, x' \in \mathcal{X}, e' \in \mathcal{E}$ and such that $p(x'; x, e') = 0$ for all $e' \notin \Gamma(x)$; p_0 is the probability mass function $P[X_0 = x], x \in \mathcal{X}$, of the initial state X_0 ; and F is a *stochastic clock structure*. The automaton generates a stochastic state sequence $\{X_0, X_1, \dots\}$ through a transition mechanism (based on observations $X = x, E' = e'$):

$$X' = x' \text{ with probability } p(x'; x, e') \quad (7)$$

and it is driven by a stochastic event sequence $\{E_1, E_2, \dots\}$ generated exactly as in (3)–(6) with random variables E, Y_i , and $N_i, i \in \mathcal{E}$, instead of deterministic quantities and with $\{V_{i,k}\} \sim$

F_i ($\sim$ denotes “with distribution”). In addition, initial conditions are $X_0 \sim p_0(x)$, $Y_i = V_{i,1}$, and $N_i = 1$ if $i \in \Gamma(X_0)$. If $i \notin \Gamma(X_0)$, then Y_i is undefined and $N_i = 0$.

It is conceivable for two events to occur at the same time, in which case we need a priority scheme to overcome a possible ambiguity in the selection of the triggering event in (3). In practice, it is common to expect that every F_i in the clock structure is absolutely continuous over $[0, \infty)$ (so that its density function exists) and has a finite mean. This implies that two events can occur at exactly the same time only with probability 0.

A stochastic process $\{X(t)\}$ with state space $\mathcal{X}$ which is generated by a stochastic timed automaton $(\mathcal{X}, \mathcal{E}, \Gamma, p, p_0, F)$ is referred to as a *generalized semi-Markov process* (GSMP). This process is used as the basis of much of the sample path analysis methods for DES (see Cassandras and LaFortune 2008; Ho and Cao 1991; Glasserman 1991).

Petri Nets

An alternative modeling formalism for DES is provided by *Petri nets*, originating in the work of C. A. Petri in the early 1960s. Like an automaton, a Petri net (Peterson 1981) is a device that manipulates events according to certain rules. One of its features is the inclusion of explicit conditions under which an event can be enabled. The Petri net modeling framework is the subject of the article ► [Modeling, Analysis, and Control with Petri Nets](#). Thus, we limit ourselves here to a brief introduction. First, we define a Petri net graph, also called the *Petri net structure*. Then, we adjoin to this graph an initial state, a set of marked states, and a transition labeling function, resulting in the complete Petri net model, its associated dynamics, and the languages that it generates and marks.

Petri Net Graph. A Petri net is a directed *bipartite graph* with two types of nodes, *places* and *transitions*, and arcs connecting them. Events are associated with transition nodes. In order

for a transition to occur, several conditions may have to be satisfied. Information related to these conditions is contained in place nodes. Some such places are viewed as the “input” to a transition; they are associated with the conditions required for this transition to occur. Other places are viewed as the output of a transition; they are associated with conditions that are affected by the occurrence of this transition. A *Petri net graph* is formally defined as a weighted directed bipartite graph (P, T, A, w) where P is the finite set of *places* (one type of node in the graph), T is the finite set of *transitions* (the other type of node in the graph), $A \subseteq (P \times T) \cup (T \times P)$ is the set of arcs with directions from places to transitions and from transitions to places in the graph, and $w : A \rightarrow \{1, 2, 3, \dots\}$ is the *weight function* on the arcs. Let $P = \{p_1, p_2, \dots, p_n\}$, and $T = \{t_1, t_2, \dots, t_m\}$. It is convenient to use $I(t_j)$ to represent the set of input places to transition t_j . Similarly, $O(t_j)$ represents the set of output places from transition t_j . Thus, we have $I(t_j) = \{p_i \in P : (p_i, t_j) \in A\}$ and $O(t_j) = \{p_i \in P : (t_j, p_i) \in A\}$.

Petri Net Dynamics. *Tokens* are assigned to places in a Petri net graph in order to indicate the fact that the condition described by that place is satisfied. The way in which tokens are assigned to a Petri net graph defines a *marking*. Formally, a *marking* x of a Petri net graph (P, T, A, w) is a function $x : P \rightarrow \mathbb{N} = \{0, 1, 2, \dots\}$. Marking x defines row vector $\mathbf{x} = [x(p_1), x(p_2), \dots, x(p_n)]$, where n is the number of places in the Petri net. The i th entry of this vector indicates the (nonnegative integer) number of tokens in place p_i , $x(p_i) \in \mathbb{N}$. In Petri net graphs, a token is indicated by a dark dot positioned in the appropriate place. The *state* of a Petri net is defined to be its marking vector $\mathbf{x}$. The state transition mechanism of a Petri net is captured by the structure of its graph and by “moving” tokens from one place to another. A transition $t_j \in T$ in a Petri net is said to be *enabled* if

$$x(p_i) \geq w(p_i, t_j) \text{ for all } p_i \in I(t_j). \quad (8)$$

In words, transition t_j in the Petri net is enabled when the number of tokens in p_i is at least as large as the weight of the arc connecting p_i to t_j , for all places p_i that are input to transition t_j . When a transition is enabled, it can occur or *fire*. The *state transition function* of a Petri net is defined through the change in the state of the Petri net due to the firing of an enabled transition. The state transition function, $f : \mathbb{N}^n \times T \rightarrow \mathbb{N}^n$, of Petri net (P, T, A, w, x) is defined for transition $t_j \in T$ if and only if (8) holds. Then, we set $\mathbf{x}' = f(\mathbf{x}, t_j)$ where

$$\begin{aligned} x'(p_i) &= x(p_i) - w(p_i, t_j) + w(t_j, p_i), \\ i &= 1, \dots, n. \end{aligned} \quad (9)$$

An “enabled transition” is therefore equivalent to a “feasible event” in an automaton. But whereas in automata the state transition function enumerates all feasible state transitions, here the state transition function is based on the structure of the Petri net. Thus, the next state defined by (9) explicitly depends on the input and output places of a transition and on the weights of the arcs connecting these places to the transition. According to (9), if p_i is an input place of t_j , it loses as many tokens as the weight of the arc from p_i to t_j ; if it is an output place of t_j , it gains as many tokens as the weight of the arc from t_j to p_i . Clearly, it is possible that p_i is both an input and output place of t_j .

In general, it is entirely possible that, after several transition firings, the resulting state is $\mathbf{x} = [0, \dots, 0]$ or that the number of tokens in one or more places grows arbitrarily large after an arbitrarily large number of transition firings. The latter phenomenon is a key difference with automata, where finite-state automata have only a finite number of states, by definition. In contrast, a finite Petri net graph may result in a Petri net with an unbounded number of states. It should be noted that a finite-state automaton can always be represented as a Petri net; on the other hand, not all Petri nets can be represented as finite-state automata.

Similar to timed automata, we can define *timed Petri nets* by introducing a clock structure, except that now a clock sequence $\mathbf{v}_j$ is associated

with a transition t_j . A positive real number, $v_{j,k}$, assigned to t_j has the following meaning: when transition t_j is enabled for the k th time, it does not fire immediately, but incurs a firing delay given by $v_{j,k}$; during this delay, tokens are kept in the input places of t_j . Not all transitions are required to have firing delays. Thus, we partition T into subsets T_0 and T_D , such that $T = T_0 \cup T_D$. T_0 is the set of transitions always incurring zero firing delay, and T_D is the set of transitions that generally incur some firing delay. The latter are called *timed transitions*. The *clock structure* (or *timing structure*) associated with a set of timed transitions $T_D \subseteq T$ of a marked Petri net (P, T, A, w, x) is a set $\mathbf{V} = \{\mathbf{v}_j : t_j \in T_D\}$ of clock (or lifetime) sequences $\mathbf{v}_j = \{v_{j,1}, v_{j,2}, \dots\}$, $t_j \in T_D$, $v_{j,k} \in \mathbb{R}^+$, $k = 1, 2, \dots$. A *timed Petri net* is a six-tuple $(P, T, A, w, x, \mathbf{V})$, where (P, T, A, w, x) is a marked Petri net and $\mathbf{V} = \{\mathbf{v}_j : t_j \in T_D\}$ is a clock structure. It is worth mentioning that this general structure allows for a variety of behaviors in a timed Petri net, including the possibility of multiple transitions being enabled at the same time or an enabled transition being preempted by the firing of another, depending on the values of the associated firing delays. The need to analyze and control such behavior in DES has motivated the development of a considerable body of analysis techniques for Petri net models which have been proven to be particularly suitable for this purpose (Peterson 1981; Moody and Antsaklis 1998).

Dioid Algebras

Another modeling framework is based on developing an algebra using two operations: $\min\{a, b\}$ (or $\max\{a, b\}$) for any real numbers a and b and addition $(a + b)$. The motivation comes from the observation that the operations “min” and “+” are the only ones required to develop the timed automaton model. Similarly, the operations “max” and “+” are the only ones used in developing the timed Petri net models described above. The operations are formally named *addition* and *multiplication* and denoted by $\oplus$ and $\otimes$ respectively. However, their actual meaning (in

terms of regular algebra) is different. For any two real numbers a and b , we define

$$\text{Addition : } a \oplus b \equiv \max\{a, b\} \quad (10)$$

$$\text{Multiplication : } a \otimes b \equiv a + b. \quad (11)$$

This dioid algebra is also called a $(\max, +)$ algebra (Cuninghame-Green 1979; Baccelli et al. 1992). If we consider a standard linear discrete time system, its state equation is of the form

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k),$$

which involves (regular) multiplication ($\times$) and addition ($+$). It turns out that we can use a $(\max, +)$ algebra with DES, replacing the $(+, \times)$ algebra of conventional time-driven systems, and come up with a representation similar to the one above, thus paralleling to a considerable extent the analysis of classical time-driven linear systems. We should emphasize, however, that this particular representation is only possible for a subset of DES. Moreover, while conceptually this offers an attractive way to capture the event timing dynamics in a DES, from a computational standpoint, one still has to confront the complexity of performing the “max” operation when numerical information is ultimately needed to analyze the system or to design controllers for its proper operation.

Control and Optimization of Discrete Event Systems

The various control and optimization methodologies developed to date for DES depend on the modeling level appropriate for the problem of interest.

Logical Behavior. Issues such as ordering events according to some specification or ensuring the reachability of a particular state are normally addressed through the use of automata and Petri nets (Ramadge and Wonham 1987; Chen and Lafortune 1991; Moody and Antsaklis 1998). Supervisory control theory provides a systematic framework for formulating and

solving problems of this type; a comprehensive coverage can be found in Cassandras and Lafortune (2008). Logical behavior issues are also encountered in the *diagnosis* of partially observed DES, a topic covered in the article [► Diagnosis of Discrete Event Systems](#).

Event Timing. When timing issues are introduced, timed automata and timed Petri nets are invoked for modeling purposes. Supervisory control in this case becomes significantly more complicated. An important class of problems, however, does not involve the ordering of individual events, but rather the requirement that selected events occur within a given “time window” or with some desired periodic characteristics. Models based on the algebraic structure of timed Petri nets or the $(\max, +)$ algebra provide convenient settings for formulating and solving such problems (Baccelli et al. 1992; Glasserman and Yao 1994).

Performance Analysis. As in classical control theory, one can define a performance (or cost) function as a means for quantifying system behavior. This approach is particularly crucial in the study of stochastic DES. Because of the complexity of DES dynamics, analytical expressions for such performance metrics in terms of controllable variables are seldom available. This has motivated the use of simulation and, more generally, the study of DES sample paths; these have proven to contain a surprising wealth of information for control purposes. The theory of *perturbation analysis* presented in the article [► Perturbation Analysis of Discrete Event Systems](#) has provided a systematic way of estimating performance sensitivities with respect to system parameters (Cassandras and Lafortune 2008; Ho and Cao 1991; Glasserman 1991; Cassandras and Panayiotou 1999).

Discrete Event Simulation. Because of the aforementioned complexity of DES dynamics, simulation becomes an essential part of DES performance analysis (Law and Kelton 1991). Discrete event simulation can be defined as a systematic way of generating sample paths of a DES by means of a timed automaton or its

stochastic counterpart. The same process can be carried out using a Petri net model or one based on the dioid algebra setting.

Optimization. Optimization problems can be formulated in the context of both untimed and timed models of DES. Moreover, such problems can be formulated in both a deterministic and a stochastic setting. In the latter case, the ability to efficiently estimate performance sensitivities with respect to controllable system parameters provides a powerful tool for stochastic gradient-based optimization (when one can define derivatives) (Vázquez-Abad et al. 1998).

A treatment of all such problems from an application-oriented standpoint, along with further details on the use of the modeling frameworks discussed in this entry, can be found in the article ► [Applications of Discrete Event Systems](#).

Cross-References

- [Applications of Discrete Event Systems](#)
- [Diagnosis of Discrete Event Systems](#)
- [Discrete Event Systems and Hybrid Systems, Connections Between](#)
- [Modeling, Analysis, and Control with Petri Nets](#)
- [Perturbation Analysis of Discrete Event Systems](#)
- [Perturbation Analysis of Steady-State Performance and Relative Optimization](#)
- [Supervisory Control of Discrete-Event Systems](#)

Bibliography

- Baccelli F, Cohen G, Olsder GJ, Quadrat JP (1992) Synchronization and linearity. Wiley, Chichester/New York
- Baeten JCM, Weijland WP (1990) Process algebra. Volume 18 of Cambridge tracts in theoretical computer science. Cambridge University Press, Cambridge/New York
- Cassandras CG, Lafortune S (2008) Introduction to discrete event systems, 2nd edn. Springer, New York
- Cassandras CG, Panayiotou CG (1999) Concurrent sample path analysis of discrete event systems. *J Discret Event Dyn Syst Theory Appl* 9:171–195
- Chen E, Lafortune S (1991) Dealing with blocking in supervisory control of discrete event systems. *IEEE Trans Autom Control* AC-36(6):724–735
- Cuninghame-Green RA (1979) Minimax algebra. Number 166 in lecture notes in economics and mathematical systems. Springer, Berlin/New York
- Glasserman P (1991) Gradient estimation via perturbation analysis. Kluwer Academic, Boston
- Glasserman P, Yao DD (1994) Monotone structure in discrete-event systems. Wiley, New York
- Ho YC (ed) (1991) Discrete event dynamic systems: analyzing complexity and performance in the modern world. IEEE, New York
- Ho YC, Cao X (1991) Perturbation analysis of discrete event dynamic systems. Kluwer Academic, Dordrecht
- Ho YC, Cassandras CG (1983) A new approach to the analysis of discrete event dynamic systems. *Automatica* 19:149–167
- Hoare CAR (1985) Communicating sequential processes. Prentice-Hall, Englewood Cliffs
- Hopcroft JE, Ullman J (1979) Introduction to automata theory, languages, and computation. Addison-Wesley, Reading
- Law AM, Kelton WD (1991) Simulation modeling and analysis. McGraw-Hill, New York
- Moody JO, Antsaklis P (1998) Supervisory control of discrete event systems using petri nets. Kluwer Academic, Boston
- Peterson JL (1981) Petri net theory and the modeling of systems. Prentice Hall, Englewood Cliffs
- Ramadge PJ, Wonham WM (1987) Supervisory control of a class of discrete event processes. *SIAM J Control Optim* 25(1):206–230
- Vázquez-Abad FJ, Cassandras CG, Julka V (1998) Centralized and decentralized asynchronous optimization of stochastic discrete event systems. *IEEE Trans Autom Control* 43(5):631–655

Monotone Systems in Biology

David Angeli

Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Dipartimento di Ingegneria dell'Informazione, University of Florence, Florence, Italy

Abstract

Mathematical models arising in biology might sometime exhibit the remarkable feature of preserving ordering of their solutions with

respect to initial data: in words, the “more” of x (the state variable) at time 0, the more of it at all subsequent times. Similar monotonicity properties are possibly exhibited also with respect to input levels. When this is the case, important features of the system’s dynamics can be inferred on the basis of purely qualitative or relatively basic quantitative knowledge of the system’s characteristics. We will discuss how monotonicity-related tools can be used to analyze and design biological systems with prescribed dynamical behaviors such as global stability, multistability, or periodic oscillations.

Keywords

Monotone systems · Systems biology · Synthetic biology · Convergence and stability · Biological networks · Multistability

Introduction

Ordinary differential equations of a scalar unknown, under suitable assumptions for unicity of solutions, trivially enjoy the property that any pair of ordered initial conditions (according to the standard $\leq$ order defined for real numbers) gives rise to ordered solutions at all positive times (as well as negative, though this is less relevant for the developments that follow). Monotone systems are a special but significant class of dynamical models, possibly evolving in high-dimensional or even infinite-dimensional state spaces that are nevertheless characterized by a similar property holding with respect to a suitably defined notion of partial order. They became the focus of considerable interest in mathematics after a series of seminal papers by Hirsch (1985, 1988) provided the basic definitions as well as deep results showing how generic convergence properties of their solutions are expected under suitable technical assumptions. Shortly before that, (Smale 1976), Smale’s construction had already highlighted

how specific solutions could instead exhibit arbitrary behavior (including periodic or chaotic). Further results along these lines provide insight into which set of extra assumptions allow one to strengthen generic convergence to global convergence, including existence of positive first integrals (Mierczynski 1987; Banaji and Angeli 2010), tridiagonal structure (Smillie 1984), or positive translation invariance (Angeli and Sontag 2008a).

While these tools were initially developed having in mind applications arising in ecology, epidemiology, chemistry, or economy, it was due to the increased importance of mathematical modeling in molecular biology and the subsequent rapid development of systems biology as an emerging independent field of investigation that they became particularly relevant to biology. The entry Angeli and Sontag (2003) first introduced the notion of control monotone systems, including input and output variables, that is of interest if one is looking at systems arising from interconnection of monotone modules. Small-gain theorems and related conditions were defined to study both positive (Angeli and Sontag 2004b) and negative (Angeli and Sontag 2003) feedback interconnections by relating their asymptotic behavior to properties of the discrete iterations of a suitable map, called the steady-state characteristic of the system.

In particular, convergence of this map is related to convergent solutions for the original continuous time system; on the other hand, specific negative feedback interconnections can instead give rise to oscillations as a result of Hopf’s bifurcations as in Angeli and Sontag (2008b) or to relaxation oscillators as highlighted in Gedeon and Sontag (2007).

A parallel line of investigation, originated in the work of Volpert et al. (1994), exploited the specific features of models arising in biochemistry by focusing on structural conditions for monotonicity of chemical reaction networks (Angeli et al. 2010; Banaji 2009). Monotonicity is only one of the possible tools adopted in the study of dynamics for such class of models in the related field of chemical reaction network theory.

Mathematical Preliminaries

To illustrate the main tools of monotone dynamics, we consider in the following systems defined on partially ordered input, state, and output spaces. Namely, along with the sets U , X , and Y (which denote input, state, and output space, respectively), we consider corresponding partial orders $\succeq_X, \succeq_U, \succeq_Y$. A typical way of defining a partial order on a set S embedded in some Euclidean space E , $S \subset E$, is to first identify a cone K of positive vectors which belong to E . A cone in this context is any closed convex set which is preserved under multiplication times nonnegative scalars and such that $K \cap -K = \{0\}$. Accordingly we may denote $s_1 \succeq_S s_2$ whenever $s_1 - s_2 \in K$. A typical choice of K in the case of finite dimensional $E = \mathbb{R}^n$ is the positive orthant, ($K = [0, +\infty)^n$), in which case $\succeq$ can be interpreted as componentwise inequalities. More general orthants are also very useful in several applications as well as more exotic cones, smooth or polyhedral, according to the specific model considered. When dealing with input signals, we let $\mathcal{U}$ denote the set of locally essentially bounded and measurable functions of time. In particular, we inherit a partial order on $\mathcal{U}$ from the partial order on U according to the following definition

$$u_1(\cdot) \succeq_{\mathcal{U}} u_2(\cdot) \Leftrightarrow u_1(t) \succeq_U u_2(t) \quad \forall t \in \mathbb{R}.$$

When obvious from the context, we do not emphasize the space to which variables belong and simply write $\succeq$. Strict order notions are also of interest and specially relevant for some of the deepest implications of the theory. We let $s_1 \succ s_2$ denote $s_1 \succeq s_2$ and $s_1 \neq s_2$. While for partial orders induced by positivity cones, we let $s_1 \gg s_2$ denote $s_1 - s_2 \in \text{int}(K)$.

A dynamical system is for us a continuous map $\varphi : \mathbb{R} \times X \rightarrow X$ which fulfills the property, $\varphi(0, x) = x$ for all $x \in X$ and $\varphi(t_2, \varphi(t_1, x)) = \varphi(t_1 + t_2, x)$ for all $t_1, t_2 \in \mathbb{R}$. Sometimes, when solutions are not globally defined (for instance, if the system is defined through a set of nonlinear differential equations), it is enough to restrict the

definitions that follow to the domain of existence of solutions.

Definition 1 A monotone system φ is one that fulfills the following:

$$\begin{aligned} \forall x_1, x_2 \in X : x_1 \succeq x_2 \\ \varphi(t, x_1) \succeq \varphi(t, x_2) \quad \forall t \geq 0. \end{aligned} \quad (1)$$

A system φ is strongly monotone when the following holds:

$$\begin{aligned} \forall x_1, x_2 \in X : x_1 \succ x_2 \\ \varphi(t, x_1) \gg \varphi(t, x_2) \quad \forall t > 0. \end{aligned} \quad (2)$$

A control system is characterized by two continuous mappings, $\varphi : \mathbb{R} \times X \times \mathcal{U} \rightarrow X$ and the readout map $h : X \times U \rightarrow Y$.

Definition 2 A control system is monotone if:

$$\begin{aligned} \forall u_1, u_2 \in \mathcal{U} : u_1 \succeq u_2, \\ \forall x_1, x_2 \in X : x_1 \succeq x_2, \quad \forall t \geq 0 \\ \varphi(t, x_1, u_1) \succeq \varphi(t, x_2, u_2) \end{aligned} \quad (3)$$

and:

$$\begin{aligned} \forall u_1, u_2 \in U : u_1 \succeq u_2, \\ \forall x_1, x_2 \in X : x_1 \succeq x_2, \\ h(x_1, u_1) \succeq h(x_2, u_2). \end{aligned} \quad (4)$$

Notice that for any ordered state and input pairs x_1, x_2, u_1, u_2 , the signals y_1 and y_2 defined as $y_1(t) := h(\varphi(t, x_1, u_1), u_1(t))$, $y_2(t) := h(\varphi(t, x_2, u_2), u_2(t))$ also fulfill, thanks to the Definition 2, $y_1(t) \succeq_Y y_2(t)$ (for all $t \geq 0$).

A system which is monotone with respect to the positive orthant is called cooperative. If a system is cooperative after reverting the direction of time, it is called competitive. Checking if a mathematical model specified by differential equations is monotone with respect to the partial order induced by some cone K is not too difficult. In particular, monotonicity, in its most basic formulation (1), simply amounts to a check of positive invariance of the set $\Gamma := \{(x_1, x_2) \in X^2 : x_1 \succeq x_2\}$ for a system formed by two copies of φ in parallel. This can be assessed without

explicit knowledge of solutions, for instance, by using the notion of tangent cones and Nagumo's Theorem (Angeli and Sontag 2003). Sufficient conditions also exist to assess strong monotonicity, for instance, in the case of orthant cones. Finding whether there exists an order (as induced, for instance, by a suitable cone K) which can make a system monotone is instead a harder task which normally entails a good deal of insight in the systems dynamics.

It is worth mentioning that for the special case of linear systems, monotonicity is just equivalent to invariance of the cone K , as incremental properties (referred to pairs of solutions) are just equivalent to their non-incremental counterparts (referred to the 0 solution). In this respect, a substantial amount of theory exists starting from classical works such as the Perron-Frobenius theory on positive and cone-preserving maps; this is however outside the scope of this entry; the interested reader may refer to Farina and Rinaldi (2000) for a recent book on the subject.

Monotone Dynamics

We divide this section in three parts; first we summarize the main tools for checking monotonicity with respect to orthant cones, then we recall some of the main consequences of monotonicity for the long-term behavior of solutions, and, finally, we study interconnections of monotone systems.

Checking Monotonicity

Orthant cones and the partial orders they induce play a major role in biology applications. In fact, for systems described by equations

$$\dot{x} = f(x) \quad (5)$$

with $X \subset \mathbb{R}^n$ open and $f : X \rightarrow \mathbb{R}^n$ of class $\mathcal{C}^1$, the following characterization holds:

Proposition 1 *The system φ induced by the set of differential equations (5) is cooperative if and only if the Jacobian $\frac{\partial f}{\partial x}$ is a Metzler matrix for all $x \in X$.*

We recall that M is Metzler if $m_{ij} \geq 0$ for all $i \neq j$. Let $\Lambda = \text{diag}[\lambda_1, \dots, \lambda_n]$ with $\lambda_i \in \{-1, 1\}$ and assume that the orthant $\mathcal{O} = \Lambda[0, +\infty)^n$. It is straightforward to see

that $x_1 \succeq_{\mathcal{O}} x_2 \Leftrightarrow \Lambda x_1 \geq \Lambda x_2$, where $\geq$ denotes the partial order induced by the positive orthant, while $\succeq_{\mathcal{O}}$ denotes the order induced by $\mathcal{O}$. This means that we may check monotonicity with respect to $\mathcal{O}$ by performing a simple change of coordinates $z = \Lambda x$. As a corollary:

Proposition 2 *The system φ induced by the set of differential equations (5) is monotone with respect to $\succeq_{\mathcal{O}}$ if and only if $\Lambda \frac{\partial f}{\partial x} \Lambda$ is a Metzler matrix for all $x \in X$.*

Notice that conditions of Propositions 1 and 2 can be expressed in terms of sign constraints on off-diagonal entries of the Jacobian; in biological terms a sign constraint in an off-diagonal entry amounts to asking that a particular species (meaning chemical compound or otherwise) consistently exhibit throughout the considered model's state space either an excitatory or inhibitory effect on some other species of interest. Qualitative diagrams showing effects of species on each other are commonly used by biologists to understand the working principles of biomolecular networks.

Remarkably, Proposition 2 has also an interesting graph theoretical interpretation if one thinks of $\text{sign}\left(\frac{\partial f}{\partial x}\right)$ as the adjacency matrix of a graph with nodes $x_1 \dots x_n$ corresponding to the state variables of the system.

Proposition 3 *The system φ induced by the set of differential equations (5) is monotone with respect to $\succeq_{\mathcal{O}}$ if and only if the directed graph of adjacency matrix $\text{sign}\left(\frac{\partial f}{\partial x}\right)$ (neglecting diagonal entries) has undirected loops with an even number of negative edges.*

This means in particular that $\frac{\partial f}{\partial x}$ must be sign-symmetric (no predator-prey-type interactions) and in addition that a similar parity property has to hold on undirected loops of arbitrary length. Sufficient conditions for strong monotonicity are also known, for instance, in terms of irreducibility of the Jacobian matrix (Kamke's condition; see Hirsch and Smith 2003).

Asymptotic Dynamics

As previously mentioned, several important implications of monotonicity are with respect

to asymptotic dynamics. Let $\mathcal{E}$ denote the set of equilibria of φ . The following result is due to Hirsch (1985).

Theorem 1 *Let φ be a strongly monotone system with bounded solutions. There exists a zero measure set Q such that each solution starting in $X \setminus Q$ converges toward $\mathcal{E}$.*

Global convergence results can be achieved for important classes of monotone dynamics. For instance, when increasing conservation laws are present, see Banaji and Angeli (2010):

Theorem 2 *Let $X \subset K \subset \mathbb{R}^n$ be any two proper cones. Let φ on X be strongly monotone with respect to the partial order induced by K and preserving a K -increasing first integral. Then every bounded solution converges.*

Dually to first integrals, positive translation invariance of the dynamics also provides grounds for global convergence (see Angeli and Sontag 2008a):

Theorem 3 *If a system is strongly monotone and fulfills $\varphi(t, x_0 + hv) = \varphi(t, x_0) + hv$ for all $h \in \mathbb{R}$ and some $v \gg 0$, then all solutions with bounded projections in $v^\perp$ converge.*

The class of tridiagonal cooperative systems has also been investigated as a significant remarkable class of global convergent dynamics (see Smillie 1984). These arise from differential equations $\dot{x} = f(x)$ when $\partial f_i / \partial x_j = 0$ for all $|i - j| > 1$.

Finally it is worth emphasizing how significant for biological systems, often subject to phenomena evolving at different time scales, are also results on singular perturbations (Wang and Sontag 2008; Gedeon and Sontag 2007) and on monotonicity (with respect to time) of transient responses, as highlighted in Angeli and Sontag (2004a) and applied in Ascensao et al. (2016).

Interconnected Monotone Systems

Results on interconnected monotone SISO systems are surveyed in Angeli and Sontag (2004a). The main tool used in this context is the notion of input-state and input-output steady-state characteristic.

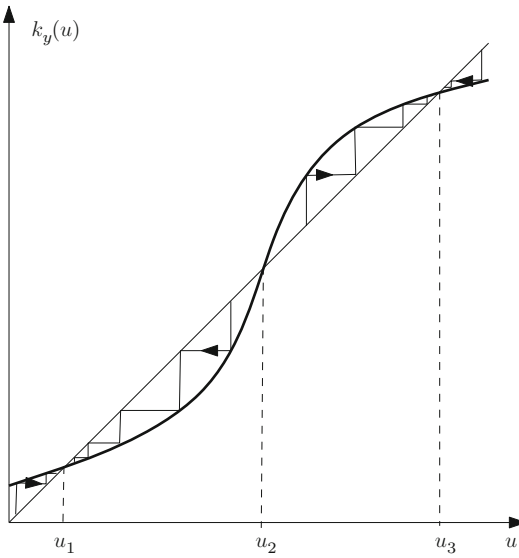
Definition 3 A control system admits a well-defined input-state characteristic if for all constant inputs u there exists a unique globally asymptotically stable equilibrium $k_x(u)$ and the map $k_x(u)$ is continuous. If moreover the equilibrium is hyperbolic, then k_x is called a nondegenerate characteristic. The input-output characteristic is defined as $k_y(u) = h(k_x(u))$.

Let φ be system with a well-defined input-output characteristic k_y ; we may define the iteration:

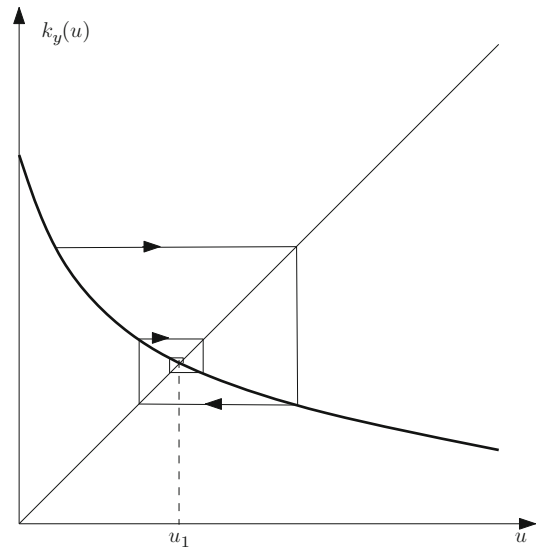
$$u_{k+1} = k_y(u_k). \quad (6)$$

It is clear that fixed points of (6) correspond to input values (and therefore to equilibria through the characteristic map k_x) of the closed-loop system derived by considering the unity feedback interconnection $u = y$. What is remarkable for monotone systems is that both in the case of positive and negative feedback and in a precise sense, stability properties of the fixed points of the discrete iteration (6) are matched by stability properties of the corresponding associated solutions of the original continuous time system. See Angeli and Sontag (2004b) for the case of positive feedback interconnections and Angeli et al. (2004) for applications of such results to synthesis and detection of multistability in molecular biology.

Multistability, in particular, is an important dynamical feature of specific cellular systems and can be achieved, with good degree of robustness with respect to different types of uncertainties, by means of positive feedback interconnections of monotone subsystems. The typical input-output characteristic k_y giving rise to such behavior is, in the SISO case, that of a sigmoidal function intersecting in 3 points the diagonal $u = y$. Two of the fixed points, namely, u_1 and u_3 (see Fig. 1), are asymptotically stable for (6), and the corresponding equilibria of the original continuous time monotone system are also asymptotically stable with a basin of attraction which covers almost all initial conditions. The fixed point u_2 is unstable and the corresponding equilibrium is also such (under suitable technical assumption on the non-degeneracy of the I-O characteristic). Extensions of similar criteria to the MIMO case are presented in Enciso and Sontag (2005). See



Monotone Systems in Biology, Fig. 1 Fixed points of a sigmoidal Input-Output characteristic



Monotone Systems in Biology, Fig. 2 Fixed point of a decreasing Input-Output characteristic

Sootla et al. (2016) for pulse design techniques finalized to driving a bistable monotone system toward a specific equilibrium state. Another recent application of bistability and monotone systems theory is quorum sensing as described in Nikolaev and Sontag (2016).

Negative feedback normally destroys monotonicity. As a result, likelihood of complex dynamical behavior is highly increased. Nevertheless, input-output characteristics still can provide useful insight in the system's dynamics at least in the case of low feedback gain or for high feedback gains in the presence of sufficiently large delays. For instance, unity negative feedback interconnection of a SISO monotone system may give rise to a unique and globally asymptotically stable fixed point of (6), thanks to the decreasingness of the input-output characteristic and as shown in Fig. 2. Under such circumstances a small-gain result applies, and global asymptotic stability of the corresponding equilibrium is guaranteed regardless of arbitrary input delays in the systems. See Angeli and Sontag (2003) for the simplest small-gain theorem developed in the context of SISO negative feedback interconnections of monotone systems and Enciso and Sontag (2006) for generalizations

to systems with multiple inputs as well as delays. A generalization of small-gain results to the case of MIMO systems which are neither in a positive nor negative feedback configuration is presented in Angeli and Sontag (2011).

When the iteration (6) has an unstable fixed point, for instance, it converges to a period-2 solution; one may expect insurgence of oscillations around the equilibrium through a Hopf's bifurcation provided sufficiently large delays in the input channels are allowed. This situation is analyzed in Angeli and Sontag (2008b) and illustrated through the study of the classical Golbeter's model for the *Drosophila*'s circadian rhythm. A related approach was recently proposed in Blanchini et al. (2015), where structural conditions for oscillations and multistationarity are studied by investigating the sign of feedback loops arising from the interconnection of monotone systems.

Summary and Future Directions

Verifying that a control system preserves some ordering of initial conditions provides important and far-reaching implications for its dynamics. Insurgence of specific behaviors can often

be inferred on the basis of purely qualitative knowledge (as in the case of the Hirsch's generic convergence theorem) as well as additional basic quantitative knowledge as in the case of positive and negative feedback interconnections of monotone systems. For the above reasons, applications in molecular biology of monotone system's theory are gradually emerging: for instance, in the study of MAPK cascades or circadian oscillations, as well as in chemical reaction network theory. Generally speaking, while monotonicity as a whole cannot be expected in large networks, experimental data shows that the number of negative feedback loops in biological regulatory networks is significantly lower than in a random signed graph of comparable size (Maayan et al. 2008).

Analyzing the properties of monotone dynamics may potentially lead to better understanding of the key regulatory mechanisms of complex networks as well as the development of bottom-up approaches for identification of meaningful submodules in biological networks. Potential research directions may include both novel computational tools as well as specific applications to systems biology, for instance:

- algorithms for detection of monotonicity with respect to exotic orders (such as arbitrary polytopic cones or even state-dependent cones);
- application of monotonicity-based ideas to control synthesis (see, for instance, Aswani and Tomlin (2009) where the special class of piecewise affine systems is considered);

Cross-References

- [Structural Properties of Biological and Ecological Systems](#)
- [Synthetic Biology](#)

Recommended Reading

For readers interested in the mathematical details of monotone systems theory, we recommend:

Smith H (1995) Monotone dynamical systems: An introduction to the theory of

competitive and cooperative systems. AMS Mathematical Surveys and Monographs, Vol 41, Providence RI, 1995

A more recent technical survey of aspects related to asymptotic dynamics of monotone systems is

Hirsch MW and Smith H (2005) Monotone Dynamical Systems, Chapter 4 in *HANDBOOK OF DIFFERENTIAL EQUATIONS Ordinary Differential Equations*, vol 2 Canada A, Drábek P and Fonda A 2005 editors, Elsevier, 2005

The following is the list of references quoted in the text.

Bibliography

- Angeli D, Sontag ED (2003) Monotone control systems. *IEEE Trans Aut Cont* 48:1684–1698
- Angeli D, Sontag ED (2004a) Interconnections of monotone systems with steady-state characteristics. In: *Optimal control, stabilization and nonsmooth analysis. Lecture notes in control and information sciences*, vol 301. Springer, Berlin, pp 135–154
- Angeli D, Sontag ED (2004b) Multi-stability in monotone input/output systems. *Sys Cont Lett* 51:185–202
- Angeli D, Sontag ED (2008a) Translation-invariant monotone systems, and a global convergence result for enzymatic futile cycles. *Nonlin Anal Ser B: Real World Appl* 9:128–140
- Angeli D, Sontag ED (2008b) Oscillations in I/O monotone systems. *IEEE Trans on Circ and Sys* 55:166–176
- Angeli D, Sontag ED (2011) A small-gain result for orthant-monotone systems in feedback: the non sign-definite case. Paper appeared in the 50th IEEE conference on decision and control, Orlando FL, 12–15 Dec 2011
- Angeli D, Sontag ED (2013) Behavior of responses of monotone and sign-definite systems. In Hüper K, Trumpf J (eds) *Mathematical system theory – Festschrift in honor of Uwe Helmke on the occasion of his sixtieth birthday*, pp 51–64
- Angeli D, Ferrell JE, Sontag ED (2004) Detection of multistability, bifurcations, and hysteresis in a large class of biological positive-feedback systems. *Proc Natl Acad Sci USA* 101:1822–1827
- Angeli D, de Leenheer P, Sontag ED (2010) Graph-theoretic characterizations of monotonicity of chemical networks in reaction coordinates. *J Math Bio* 61: 581–616
- Ascensao JA, Datta P, Hancioglu B, Sontag ED, Genaro ML, Igoshin OA (2016) Non-monotonic response dynamics of glyoxylate shunt genes in *Mycobacterium tuberculosis*. *PLoS Comput Biol* 12:e1004741
- Aswani A, Tomlin C (2009) Monotone piecewise affine systems. *IEEE Trans Aut Cont* 54:1913–1918
- Banaji M (2009) Monotonicity in chemical reaction systems. *Dyn Syst* 24:1–30

- Banaji M, Angeli D (2010) Convergence in strongly monotone systems with an increasing first integral. *SIAM J Math Anal* 42:334–353
- Banaji M, Angeli D (2012) Addendum to “Convergence in strongly monotone systems with an increasing first integral”. *SIAM J Math Anal* 44:536–537
- Blanchini F, Franco E, Giordano G (2015) Structural conditions for oscillations and multistationarity in aggregate monotone systems. In: *Proceedings of the IEEE conference on decision and control*, pp 609–614
- Enciso GA, Sontag ED (2005) Monotone systems under positive feedback: multistability and a reduction theorem. *Sys Cont Lett* 54:159–168
- Enciso GA, Sontag ED (2006) Global attractivity, I/O monotone small-gain theorems, and biological delay systems. *Disc Contin Dyn Syst* 14: 549–578
- Enciso GA, Sontag ED (2008) Monotone bifurcation graphs. *J Biol Dyn* 2:121–139
- Farina L, Rinaldi S (2000) *Positive linear systems: theory and applications*. Wiley, New York
- Gedeon T, Sontag ED (2007) Oscillations in multi-stable monotone systems with slowly varying feedback. *J Differ Equ* 239:273–295
- Hirsch MW (1985) Systems of differential equations that are competitive or cooperative II: convergence almost everywhere. *SIAM J Math Anal* 16:423–439
- Hirsch MW (1988) Stability and convergence in strongly monotone dynamical systems. *Reine Angew Math* 383:1–53
- Hirsch MW, Smith HL (2003) Competitive and cooperative systems: a mini-review. In: *Positive systems. Lecture notes in control and information science*, vol 294. Springer, pp 183–190
- Maayan A, Iyengar R, Sontag ED (2008) Intracellular regulatory networks are close to monotone systems. *IET Syst Biol* 2:103–112
- Mierczynski J (1987) Strictly cooperative systems with a first integral. *SIAM J Math Anal* 18:642–646
- Nikolaev EV, Sontag ED (2016) Quorum-sensing synchronization of synthetic toggle switches: a design based on monotone dynamical systems theory. *PLoS Comput Biol* 12(4):e1004881. <https://doi.org/10.1371/journal.pcbi.1004881>
- Smale S (1976) On the differential equations of species in competition. *J Math Biol* 3:5–7
- Smillie J (1984) Competitive and cooperative tridiagonal systems of differential equations. *SIAM J Math Anal* 15:530–534
- Sootla A, Oyarzún D, Angeli D, Stan GB (2016) Shaping pulses to control bistable systems: analysis, computation and counterexamples. *Automatica* 63: 254–264
- Volpert AI, Volpert VA, Volpert VA (1994) *Traveling wave solutions of parabolic systems*. AMS translations of mathematical monographs, vol 140. American Mathematical Society, Providence
- Wang L, Sontag ED (2008) Singularly perturbed monotone systems and an application to double phosphorylation cycles. *J Nonlin Sci* 18:527–550

Motion Description Languages and Symbolic Control

Sean B. Andersson

Mechanical Engineering and Division of Systems Engineering, Boston University, Boston, MA, USA

Abstract

The fundamental idea behind symbolic control is to mitigate the complexity of a dynamic system by limiting the set of available controls to a typically finite collection of symbols. Each symbol represents a control law that may be either open- or closed-loop. With these symbols, a simpler description of the motion of the system can be created, thereby easing the challenges of analysis and control design. In this entry, we provide a high-level description of symbolic control; discuss briefly its history, connections, and applications; and provide a few insights into where the field is going.

Keywords

Abstraction · Complex systems · Formal methods

Introduction

Systems and control theory is a powerful paradigm for analyzing, understanding, and controlling dynamic systems. Traditional tools in the field for developing and analyzing control laws, however, face significant challenges when one needs to deal with the complexity that arises in many practical, real-world settings such as the control of autonomous, mobile systems operating in uncertain, and changing physical environments. This is particularly true when the tasks to be achieved are not easily framed in terms of motion to a point in the state space. One of the primary goals of symbolic control is to mitigate this complexity by abstracting some combination of the system dynamics, the space

of control inputs, and the physical environment to a simpler, typically finite, model.

This fundamental idea, namely, that of abstracting away the complexity of the underlying dynamics and environment, is in fact a quite natural one. Consider, for example, how you give instructions to another person wanting to go to a point of interest. It would be absurd to provide details at the level of their actuators, namely, with commands to their individual muscles (or, to carry the example to an even more absurd extreme, to the dynamic components that make up those muscles). Rather, very high-level commands are given, such as “follow the road,” “turn right,” and so on. Each of these provides a description of what to do with the understanding that the person can carry out those commands in their own fashion. Similarly, the environment itself is abstracted, and only elements meaningful to the task at hand are described. Thus, continuing the example above, rather than providing metric information or a detailed map, the instructions may use environmental features to determine when particular actions should be terminated, and the next begun, such as “follow the road until the second intersection, then turn right.”

Underlying the idea of symbolic control is the notion that rich behaviors can result from simple actions. This premise was used in many early robots and can be traced back at least to the ideas of Norbert Wiener on cybernetics (see Arkin 1998). It is at the heart of the behavior-based approach to robotics (Brooks 1986). Similar ideas can also be seen in the development of a high-level language (G-codes) for computer numerically controlled (CNC) machines. The key technical ideas in the more general setting of symbolic control for dynamic systems can be traced back to Brockett (1988) which introduced ideas of formalizing a modular approach to programming motion control devices through the development of a motion description language (MDL).

The goal of the present work is to introduce the interested reader to the general ideas of symbolic control as well as to some of its application areas and research directions. While it is not a survey paper, a few select references are provided

throughout to point the reader in hopefully fruitful directions into the literature.

Models and Approaches

There are at least two related but distinct approaches to symbolic control. Both begin with a mathematical description of the system, typically given as an ordinary differential equation of the form

$$\dot{x} = f(x, u, t), \quad y = h(x, t) \quad (1)$$

where x is a vector describing the state of the system, y is the output of the sensors of the system, and u is the control input.

Under the first approach to symbolic control, the focus is on reducing the complexity of the space of possible control signals by limiting the system to a typically finite collection of control symbols. Each of these symbols represents a control law that may be open loop or may utilize output feedback. For example, *Follow the road* could be a feedback control law that uses sensor measurements to determine the position relative to the road and then applies steering commands so that the system stays on the road while simultaneously maintaining a constant speed. There are, of course, many ways to accomplish the specifics of this task, and the details will depend on the particular system. Thus, an autonomous four-wheeled vehicle equipped with a laser rangefinder, an autonomous motorcycle equipped with ultrasonic sensors, or an autonomous aerial vehicle with a camera would each carry out the command in their own way, and each would have very different trajectories. They would all, however, satisfy the notion of *Follow the road*. Description of the behavior of the system can then be given in terms of the abstract symbols rather than in terms of the details of the trajectories.

Typically, each of these symbols describes an action that at least conceptually is simple. In order to generate rich motions to carry out complex tasks, the system is switched between the available symbols. Switching conditions are often referred to as *interrupts*. Interrupts may be

purely time based (e.g., apply a given symbol for T seconds) or may be expressed in terms of symbols representing certain environmental conditions. These may be simple function of the measurements (e.g., interrupt when an intersection is detected) or may represent more complicated scenarios with history and dynamics (e.g., interrupt after the second intersection is detected). Just as the input symbols abstract away the details of the control space and of the motion of the system, the interrupt symbols abstract away the details of the environment. For example, *intersection* has a clear high-level meaning but a very different sensor “signature” for particular systems.

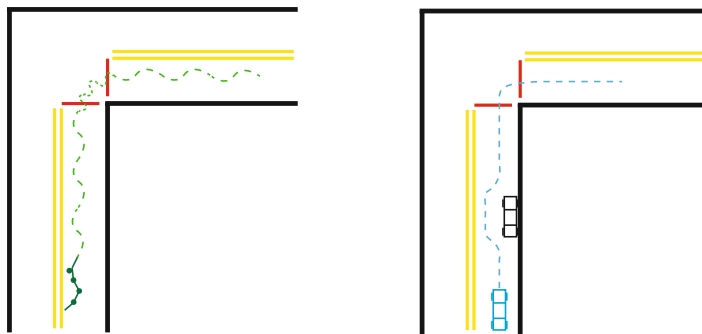
As a simple illustrative example, consider a collection of control symbols designed for moving along a system of roads, {*Follow road*, *Turn right*, *Turn left*} and a collection of interrupt symbols for triggering changes in such a setting {*In intersection*, *Clear of intersection*}. Suppose there are two vehicles that can each interpret these symbols, an autonomous car and a snake-like robot, as illustrated in Fig. 1. It is reasonable to assume that the control symbols each describe relatively complex dynamics that allow, for example, for obstacle avoidance while carrying out the action. Figure 1 illustrates a possible situation where the two systems carry out the plan defined by the symbolic sequence

(*Follow the road UNTIL In intersection*)
(*Turn right UNTIL Clear of intersection*)

The intent of this plan is for the system to navigate a right-hand turn. As shown in the figure, the actual trajectories followed by the systems can be markedly different due in part to system dynamics (the snakelike robot undulates while the car does not) as well as to different sensor responses (when the car goes through, there is a parked vehicle that it must navigate around while the snakelike robot found a clear path during its execution). Despite these differences, both systems achieve the goal of the plan.

The collection of control and interrupt symbols can be thought of as a language for describing and specifying motion and are used to write programs that can be compiled into an executable for a specific system. Different rules for doing this can be established that define different languages, analogous to different high-level programming languages such as C++, Java, or Python. Further details can be found in, for example, Manikonda et al. (1998).

Under the second approach, the focus is on representing the dynamics and state space (or environment) of the system in an abstract, symbolic way. The fundamental idea is to lump all the states in a region into a single abstract element and to then represent the entire system with a finite number of these elements. Control laws



Motion Description Languages and Symbolic Control, Fig. 1 Simple example of symbolic control with a focus on abstracting the inputs. Two systems, a snake-like robot and an autonomous car, are given a high-level plan in terms of symbols for navigating a right-hand turn.

The systems each interpret the same symbols in their own ways, leading to different trajectories due both to differences in dynamics but also to different sensors cues as caused, for example, by the parked vehicle encountered by the car in this scenario

are then defined that steer all the states in one element into some state in a different region. The symbolic control system is then the finite set of elements representing regions together with the finite set of controllers for moving between them. It can be thought of essentially as a graph (or more accurately as a transition system) in which the nodes represent regions in the state space and the edges represent achievable transitions between them. The goal of this abstraction step is for the two representations to be equivalent (or at least approximately equivalent) in that any motion that can be achieved in one can be achieved in the other (in an appropriate sense). Planning and analysis can then be done on the (simpler) symbolic model. Further details on such schemes can be found in, for example, Tabuada (2006) and Bicchi et al. (2006).

As an illustrative example, consider as before a snake-like robot moving through a right-hand turn. In a simplified view of this second approach to symbolic control, one begins by dividing the environment up into regions and then defining controllers to steer the robot from region to region as illustrated in the left image in Fig. 2. This yields the symbolic model shown in the center of Fig. 2. A plan is then developed on this model to move from the initial position to the final position. This planning step can take into account restrictions on the motion and subgoals of the task. Here, for example, one may want the robot to stay to the right of the double yellow line, which is in its lane of traffic. The plan $R_2 \rightarrow R_4 \rightarrow R_6 \rightarrow R_8 \rightarrow R_9 \rightarrow R_{10}$ is one sequence that drives the system around the turn while satisfying the lane requirement. Each transition in the sequence corresponds to a control law. The plan is then executed by applying the sequence of control laws, resulting in the actual trajectory shown in the right image in Fig. 2.

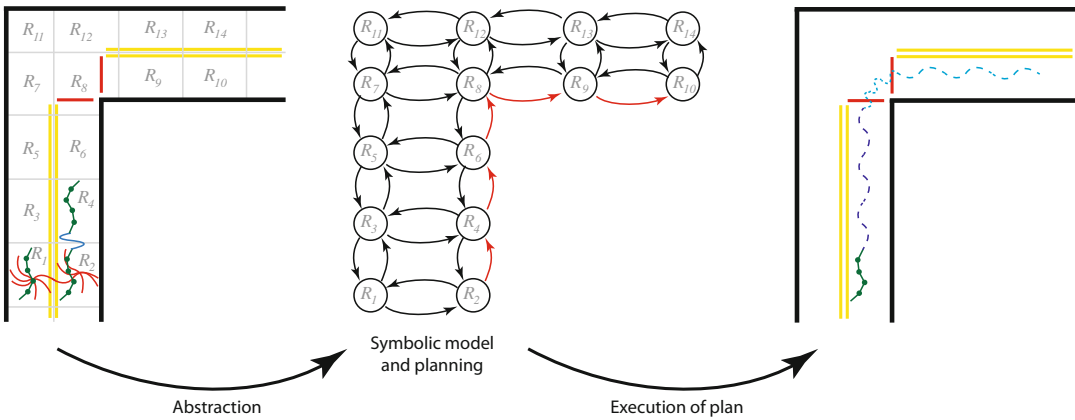
Applications and Connections

The fundamental idea behind symbolic control, namely, mitigating complexity by abstracting a system, its environment, and even the tasks to be accomplished into a simpler but (approximately)

equivalent model, is a natural and a powerful one. It has clear connections to both hybrid systems (Brockett 1993; Egerstedt 2002) and to quantized control (Bicchi et al. 2006), and the tools from those fields are often useful in describing and analyzing systems with symbolic representations of the control and of the dynamics. Symbolic control is not, however, strictly a subcategory of either field, and it provides a unique set of tools for the control and analysis of dynamic systems.

Brockett's original MDL was intended to serve as a tool for describing and planning robot motion. Inspired in part by this, languages for motion continue to be developed. Some of these extend and provide a more formal basis for motion programming (Manikonda et al. 1998) and interconnection of dynamic systems into a single whole (Murray et al. 1992), while some are designed for specialized dynamics or applications such as flight vehicles (Frazzoli et al. 2005), self-assembly (Klavins 2007), and other areas. In addition to studying standard systems and control theoretic ideas, including notions of reachability (Bicchi et al. 2002) and stability (Tarraf et al. 2008), the framework of symbolic control introduces interesting questions such as how to understand the reduction of complexity that can be achieved for a given collection of symbols (Egerstedt and Brockett 2003).

While there are many application areas of symbolic control, a particularly active one is that of motion planning for autonomous mobile robots (Belta et al. 2007). As illustrated in Figs. 1 and 2, symbolic control allows the planning problem (i.e., the determination of how to achieve a desired task) to be separated from the complexities of the dynamics. The approach has been particularly fertile when coupled with symbolic descriptions of the tasks to be achieved. While point-to-point commands are useful, and can be often thought of as symbols themselves from which to build more complicated commands, most tasks that one would want mobile robots to carry out involve combinations of spatial goals (move to a certain location), sequencing (first do this and then do that), or other temporal requirements (repeatedly visit a collection of regions), as



Motion Description Languages and Symbolic Control, Fig. 2 Simple example of symbolic control with a focus on abstracting the system dynamics and environment. The initial environment (left image) is segmented into different regions and simple controllers developed for moving from region to region. The image shows two possible controllers, one that actuates the robot through a tight slither pattern to move forward by one region and

one that twists the robot to face the cell to the left before slithering across and then reorienting. The combination of regions and actions yields a symbolic abstraction (center image) that allows for planning to achieve specific goals, such as moving through the right-hand turn. Executing this plan leads to a physical trajectory of the system (right image)

well as safety or other restrictions (avoid obstacles or regions that are dangerous for the robot to traverse). Such tasks can be described using a variety of temporal logics. These are, essentially, logic systems that include rules related to time in addition to the standard Boolean operators. These tasks can be combined with a symbolic description of a system, and then automated tools used both to check whether the system is able to perform the desired task as well as to design plans that ensure the system will do so (Fainekos et al. 2009). This symbolic approach to the design of control systems finds application in other domains as well, including systems biology and bioinformatics (Bartocci et al. 2018).

Summary and Future Directions

Symbolic control proceeds from the basic goal of mitigating the complexity of dynamic systems, especially in real-world scenarios, to yield a simplification of the problems of analysis and control design. It builds upon results from diverse fields while also contributing new ideas to those areas, including hybrid system theory, formal languages

and grammars, and motion planning. There are many open, interesting questions that are the subject of ongoing investigations as well as the genesis of future research.

One particularly fruitful direction is that of combining symbolic control with stochasticity. Systems that operate in the real world are subject to noise both with respect to their inputs (noisy actuators) and to their outputs (noisy sensors). Recent work along these lines can be found in the formal method approach to motion planning and in hybrid systems (Lahijanian et al. 2012; Abate et al. 2011). The fundamental idea is to use a Markov chain, Markov decision process, or similar model as the symbolic abstraction and then, as in all symbolic control, to do the analysis and planning on this simpler model.

Another interesting direction is to address questions of optimality with respect to the symbols and abstractions for a given dynamic system. Of course, the notion of “optimal” must be made clear, and there are several reasonable notions one could define. There is a clear trade-off between the complexity of individual symbols, the number of symbols used in the motion “alphabet,” and the complexity in terms of, say, average number of symbols required to code programs that achieve a

given set of tasks. The complexity of a necessary alphabet is also related to the variety of tasks the system might need to perform. An autonomous vacuuming robot is likely to need far fewer symbols in its library than an autonomous vehicle that must operate in everyday traffic conditions and respond to unusual events such as traffic jams. The question of the “right” set of symbols can also be of use in efficient descriptions of motion in domains such as dance (Baillieul and Ozcimder 2014).

It is intuitively clear that to handle complex scenarios and environments, a hierarchical approach is likely needed. Organizing symbols into progressively higher levels of abstraction should allow for more efficient reasoning, planning, and reaction to real-world settings. Such structures already appear in existing works, such as in the behavior-based approach of Brooks (1986), in the extended motion description language in Manikonda et al. (1998), and in the Spatial Semantic Hierarchy of Kuipers (2000). Despite these efforts, there is still a need for a rigorous approach for analyzing and designing symbolic hierarchical systems.

The final direction discussed here is that of the connection of symbolic control to emergent behavior in large groups of dynamic agents. There are a variety of intriguing examples in nature in which large numbers of agents following simple rules produce large-scale, coherent behavior, including in fish schools and termite and ant colonies (Johnson 2002). How can one predict the global behavior that will emerge from a large collection of independent agents following simple rules (symbols)? How can one design a set of symbols to produce a desired collective behavior? While there has been some work in symbolic control for self-assembling systems (Klavins 2007), this general topic remains a rich area for research.

Cross-References

- [Interaction Patterns as Units of Analysis in Hierarchical Human-in-the Loop Control](#)
- [Robot Motion Control](#)

Recommended Reading

Brockett’s original paper (Brockett 1988) is a surprisingly short but informational read. More thorough descriptions can be found in Manikonda et al. (1998) and Egerstedt (2002). An excellent description of symbolic control in robotics, particularly in the context of temporal logics and formal methods, can be found in Belta et al. (2007). There are also several related articles in a 2011 special issue of the IEEE Robotics and Automation magazine (Kress-Gazit 2011).

Bibliography

- Abate A, D’Innocenzo A, Di Benedetto MD (2011) Approximate abstractions of stochastic hybrid systems. *IEEE Trans Autom Control* 56(11):2688–2694
- Arkin RC (1998) Behavior-based robotics. MIT Press
- Baillieul J, Ozcimder K (2014) Dancing robots: the control theory of communication through movement. In: Laviers A, Egerstedt M (eds) *Controls and Art*. Springer, Switzerland, pp 51–72
- Bartocci E, Aydin Gol E, Haghghi I, Belta C (2018) A formal methods approach to pattern recognition and synthesis in reaction diffusion networks. *IEEE Trans Control Netw Syst* 5(1):308–320
- Belta C, Bicchi A, Egerstedt M, Frazzoli E, Klavins E, Pappas GJ (2007) Symbolic planning and control of robot motion (grand challenges of robotics). *IEEE Robot Autom Mag* 14(1):61–70
- Bicchi A, Marigo A, Piccoli B (2002) On the reachability of quantized control systems. *IEEE Trans Autom Control* 47(4):546–563
- Bicchi A, Marigo A, Piccoli B (2006) Feedback encoding for efficient symbolic control of dynamical systems. *IEEE Trans Autom Control* 51(6):987–1002
- Brockett RW (1988) On the computer control of movement. In: *IEEE International Conference on Robotics and Automation*, pp 534–540
- Brockett RW (1993) Hybrid models for motion control systems. In: Trentelman HL, Willems JC (eds) *Essays on control*. Birkhauser, Basel, pp 29–53
- Brooks R (1986) A robust layered control system for a mobile robot. *IEEE J Robot Autom* RA-2(1):14–23
- Egerstedt M (2002) Motion description languages for multi-modal control in robotics. In: Bicchi A, Cristensen H, Prattichizzo D (eds) *Control problems in robotics*. Springer, Berlin Heidelberg, pp 75–90
- Egerstedt M, Brockett RW (2003) Feedback can reduce the specification complexity of motor programs. *IEEE Trans Autom Control* 48(2):213–223
- Fainekos GE, Girard A, Kress-Gazit H, Pappas GJ (2009) Temporal logic motion planning for dynamic robots. *Automatica* 45(2):343–352

- Frazzoli E, Dahleh MA, Feron E (2005) Maneuver-based motion planning for nonlinear systems with symmetries. *IEEE Trans Robot* 21(6):1077–1091
- Girard A, Pappas GJ (2007) Approximation metrics for discrete and continuous systems. *IEEE Trans Autom Control* 52(5):782–798
- Johnson S (2002) *Emergence: the connected lives of ants, brains, cities, and software*. Scribner, New York
- Klavins E (2007) Programmable self-assembly. *Control Syst IEEE* 27(4):43–56
- Kress-Gazit H (2011) Robot challenges: toward development of verification and synthesis techniques [from the Guest Editors]. *IEEE Robot Autom Mag* 18(3):22–23
- Kuipers B (2000) The spatial semantic hierarchy. *Artif Intell* 119(1–2):191–233
- Lahijanian M, Andersson SB, Belta C (2012) Temporal logic motion planning and control with probabilistic satisfaction guarantees. *IEEE Trans Robot* 28(2):396–409
- Manikonda V, Krishnaprasad PS, Hendler J (1998) Languages, behaviors, hybrid architectures, and motion control. In: Baillieul J, Willems JC (eds) *Mathematical control theory*. Springer, New York, pp 199–226
- Murray RM, Deno DC, Pister KSJ, Sastry SS (1992) Control primitives for robot systems. *IEEE Trans Syst Man Cybern* 22(1):183–193
- Tabuada P (2006) Symbolic control of linear systems based on symbolic subsystems. *IEEE Trans Autom Control* 51(6):1003–1013
- Tarraf DC, Megretski A, Dahleh MA (2008) A framework for robust stability of systems over finite alphabets. *IEEE Trans Autom Control* 53(5):1133–1146

Motion Planning for Marine Control Systems

Andrea Caiti

DII – Department of Information Engineering and Centro “E. Piaggio”, ISME – Interuniversity Research Centre on Integrated Systems for the Marine Environment, University of Pisa, Pisa, Italy

Abstract

In this chapter we review motion planning algorithms for ships, rigs, and autonomous marine vehicles. Motion planning includes path and trajectory generation, and it goes from optimized route planning (off-line long-range path generation through

operating research methods) to reactive on-line trajectory reference generation, as given by the guidance system. Crucial to the marine systems case is the presence of environmental external forces (sea state, currents, winds) that drive the optimized motion generation process.

Keywords

Configuration space · Dynamic programming · Grid search · Guidance controller · Guidance system · Maneuvering · Motion plan · Optimization algorithms · Path generation · Route planning · Trajectory generation · World space

Introduction

Marine control systems include primarily ships and rigs moving on the sea surface, but also underwater systems, manned (submarines) or unmanned, and eventually can be extended to any kind of off-shore moving platform.

A *motion plan* consists in determining what motions are appropriate for the marine system to reach a goal, or a target/final state (LaValle 2006). Most often, the final state corresponds to a geographical location or destination, to be reached by the system while respecting constraints of physical and/or economical nature. Motion planning in marine systems hence starts from *route planning*, and then it covers desired *path generation* and *trajectory generation*. Path generation involves the determination of an ordered sequence of states that the system has to follow; trajectory generation requires that the states in a path are reached at a prescribed time.

Route, path, and trajectory can be generated off-line or on-line, exploiting the feedback from the system navigation and/or from external sources (weather forecast, etc.). In the feedback case, planning overlaps with the *guidance system*, i.e., the continuous computation of the reference (desired) state to be used as reference input by the motion control system (Fossen 2011).

Formal Definitions and Settings

Definitions and classifications as in Goerzen et al. (2010) and Petres et al. (2007) are followed throughout the section. The marine systems under considerations live in a physical space referred to as the *world space* (e.g., a submarine lives in a 3-D Euclidean space). A *configuration* q is a vector of variables that define position and orientation of the system in the world space. The set of all possible configurations is the *configuration space*, or *C-space*. The vector of configuration and configuration rate of changes is the *state* of the system $\mathbf{x} = [q^T \dot{q}^T]^T$, and the set of all the possible states is the *state space*. The kino-dynamic model associated to the system is represented by the system *state equations*. The regions of *C-space* free from obstacles are called *C-free*.

The *path planning* problem consists in determining a curve $\gamma: [0, 1] \rightarrow C\text{-free}, s \rightarrow \gamma(s)$, with $\gamma(0)$ corresponding to the initial configuration and $\gamma(1)$ corresponding to the goal configuration. Both initial and goal configurations are in *C-free*. The *trajectory planning* problem consists in determining a curve γ and a *time law*: $t \rightarrow s(t)$ s.t. $\gamma(s) = \gamma(s(t))$. In both cases, either the path or the trajectory must be compatible with the system state equations. In the following, definitions of motion algorithm properties are given referring to path planning, but the same definitions can be easily extended to trajectory planning.

A motion planning algorithm is *complete* if it finds a path when one exists, and returns a proper flag when no path exists. The algorithm is *optimal* when it provides the path that minimizes some cost function J . The (strictly positive) cost function J is *isotropic*, when it depends only on the system configuration ($J = J(q)$), or *anisotropic*, when it depends also on an external force field $\mathbf{f}$ (e.g., sea currents, sea state, weather perturbations) ($J = J(q, \mathbf{f})$). The cost function J induces a *pseudometric* in the configuration space; the distance d between configurations q_1 and q_2 through the path γ is the “cost-to-go” from q_1 to q_2 along γ :

$$d(q_1, q_2) = \int_0^1 J(\gamma_{q_1 q_2}(s), \mathbf{f}) ds \quad (1)$$

An optimal motion planning problem is:

- *Static* if there is perfect knowledge of the environment at any time, *dynamic* otherwise
- *Time-invariant* when the environment does not evolve (e.g., coastline that limits the *C-free* subspace), *time-variant* otherwise (e.g., other systems – ships, rigs – in navigation)
- *Differentially constrained* if the system state equations act as a constraint on the path, *differentially unconstrained* otherwise

In practice, optimal motion planning problems are solved numerically through discretization of the *C-space*. *Resolution* completeness/optimality of an algorithm implies the achievement of the solution as the discretization interval tends to zero. *Probabilistic* completeness/optimality implies that the probability of finding the solution tends to 1 as the computation time approaches infinity. *Complexity* of the algorithm refers to the computational time required to find a solution as a function of the dimension of the problem.

The scale of the motion w.r. to the scale of the system defines the specific setting of the problem. In cargo ships route planning from one port call to the next, the problem is stated first as static, time-invariant, differentially unconstrained *path* planning problem; once a large-scale route is thus determined, it can be refined on smaller scales, e.g., smoothing it, to make it compatible with ship maneuverability. Maneuvering the same cargo ship in the approaches to a harbor has to be casted as a dynamic, time-variant, differentially constrained *trajectory* planning problem.

Large-Scale, Long-Range Path Planning

Route determination is a typical long-range path planning problem for a marine system. The geographical map is discretized into a grid, and the optimal path between the approaches of the

starting and destination ports is determined as a sequence of adjacent grid nodes. The problem is taken as time-invariant and differentially unconstrained, at least in the first stages of the procedure. It is assumed that the ship will cruise at its own (constant) most economical speed to optimize bunker consumption, the major source of operating costs (Wang and Meng 2012). Navigation constraints (e.g., allowed ship traffic corridors for the given ship class) are considered, as well as weather forecasts and predicted/prevaling currents and winds. The cost-to-go Eq. (1) is built between adjacent nodes either in terms of time to travel or in terms of operating costs, both computed correcting the nominal speed with the environmental forces. Optimality is defined in terms of shortest time/minimum operating cost; the anisotropy introduced by sea/weather conditions is the driving element of the optimization, making the difference with respect to straightforward shortest route computation. The approach is iterated, starting with a coarse grid and then increasing grid resolution in the neighborhood of the previously found path.

The most widely used optimization approach for surface ships is *dynamic programming* (LaValle 2006); alternatively, since the determination of the optimal path along the grid nodes is equivalent to a search over a graph, the *A** algorithm is applied (Delling et al. 2009). As the discretization grid gets finer, system dynamics are introduced, accounting for ship maneuverability and allowing for deviation from the constant ship speed assumption. Dynamic programming allows to include system dynamics at any level of resolution desired; however, when system dynamics are considered, the problem dimension grows from 2-D to 3-D (2-D space plus time).

In the case of underwater navigation, path planning takes place in a 3-D world space, and the inclusion of system dynamics makes it a 4-D problem; moreover, bathymetry has to be included as an additional constraint to shape the *C-free* subspace. Dynamic programming may become unfeasible, due to the increase in dimensionality. Computationally feasible

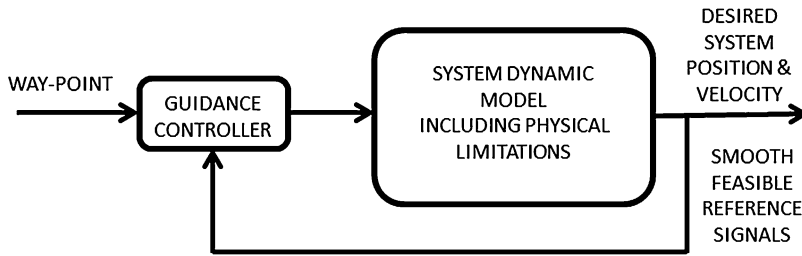
algorithms for this case include global search strategies with probabilistic optimality, as *genetic algorithms* (Alvarez et al. 2004), or improved grid-search methods with resolution optimality, as FM* (Petres et al. 2007).

Environmental force fields are intrinsically dynamic fields; moreover, the prediction of such fields at the moment of route planning may be updated as the ship is in transit along the route. The path planning algorithms can/must be rerun, over a grid in the neighborhood of the nominal path, each time new environmental information becomes available. Kino-dynamic model of the ship must be included, allowing for deviation from the established path and ship speed variation around the nominal most economical speed. The latter case is particularly important: increasing/decreasing the speed to avoid a weather perturbation keeping the same route may indeed result in a reduced operating cost with respect to path modifications keeping a constant speed. This implies that the timing over the path must be specified. Dynamic programming is well suited for this transition from path to trajectory generation, and it is still the most commonly used approach to trajectory (re)planning in reaction to environmental predictions update.

When discretizing the world space, the minimum grid size should still be large enough to allow for ship maneuvering between grid nodes. This is required for safety, to allow evasive maneuvering when other ships are at close ranges, and for the generation of smooth, dynamics-compliant trajectories between grid points. This latter aspect bridges motion planning with guidance.

Trajectory Planning, Maneuvering Generation, and Guidance Systems

Once a path has been established over a spatial grid, a continuous reference has to be generated, linking the nodes over the grid. The generation of the reference trajectory has to take into account all the relevant dynamic properties and constraints of the marine system, so that the



Motion Planning for Marine Control Systems, Fig. 1 Generation of a reference trajectory with a system model and a guidance controller (Adapted from Fossen (2011))

reference motion is *feasible*. In this scenario, the path/trajectory nodes are *way-points*, and the trajectory generation connects the way-points along the route. The approaches to trajectory generation can be divided between those that do not compute explicitly in advance the whole trajectory and those that do.

Among the approaches that do not need explicit trajectory computation between way-points, the most common is the *line-of-sight* (LOS) guidance law (Pettersen and Lefeber 2001). LOS guidance can be considered a path generation, more than a trajectory generation, since it does not impose a time law over the path; it computes directly the desired ship reference heading on the basis of the current ship position and the previous and next way-point positions. A review of other guidance approaches can be found in Breivik and Fossen (2008), where maneuvering along the path and steering around the way-points are also discussed. From such a set of different maneuvers, a library of motion primitives can be built (Greytak and Hover 2010), so that any motion can be specified as a sequence of primitives. While each primitive is feasible by construction, an arbitrary sequence of primitives may not be feasible. An optimized search algorithm (dynamic programming, A*) is again needed to determine the optimal feasible maneuvering sequence.

Path/trajectory planning explicitly computing the desired motion among two way-points may include a system dynamic model, or may not. In the latter case, a sufficiently smooth curve that connects two way-points is generated, for

instance, as splines or as Dubins paths (LaValle 2006). Curve generation parameters must be set so that the “sufficiently smooth” part is guaranteed. After curve generation, a trim velocity is imposed over the path (path planning), or a time law is imposed, e.g., smoothly varying the system reference velocity with the local curvature radius.

Planners that do use a system dynamic model are described in Fossen (2011) as part of the guidance system. In practice, the dynamic model is used in simulation, with a (simulated) feedback controller (*guidance controller*), the next way-point as input, and the (simulated) system position and velocity as output. The simulated results are feasible maneuvers by construction and can be given as reference position/velocity to the physical control system (Fig. 1).

Summary and Future Directions

Motion planning for marine control systems employs methodological tools that range from operating research to guidance, navigation, and control systems. A crucial role in marine applications is played by the anisotropy induced by the dynamically changing environmental conditions (weather, sea state, winds, currents – the external force fields). The quality of the plan will depend on the quality of environmental information and predictions.

While motion planning can be considered a mature issue for ships, rigs, and even standalone autonomous vehicles, current and future research

directions will likely focus on the following items:

- Coordinated motion planning and obstacle avoidance for *teams* of autonomous surface and underwater vehicles (Aguiar and Pascoal 2012; Casalino et al. 2009)
- Naval traffic regulation compliant maneuvering in restricted spaces and collision evasion maneuvering (Tam and Bucknall 2010)
- Underwater intervention robotics (Antonelli 2006; Sanz et al. 2010)

Cross-References

- [Control of Networks of Underwater Vehicles](#)
- [Mathematical Models of Marine Vehicle-Manipulator Systems](#)
- [Mathematical Models of Ships and Underwater Vehicles](#)
- [Underactuated Marine Control Systems](#)

Recommended Reading

Motion planning is extensively treated in LaValle (2006), while the essential reference on marine control systems is the book by Fossen (2011). Goerzen et al. (2010) reviews motion planning algorithms in terms of computational properties. The book Antonelli (2006) includes the treatment of planning and control in intervention robots. The papers Breivik and Fossen (2008) and Tam et al. (2009) provide a survey of both terminology and guidance design for both open and close space maneuvering. In particular, Tam et al. (2009) links motion planning to navigation rules.

Bibliography

Aguiar AP, Pascoal A (2012) Cooperative control of multiple autonomous marine vehicles: theoretical foundations and practical issues. In: Roberts GN, Sutton R (eds) Further advances in unmanned marine vehicles. IET, London, pp 255–282

- Alvarez A, Caiti A, Onken R (2004) Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *IEEE J Ocean Eng* 29(2): 418–429
- Antonelli G (2006) Underwater robots – motion and force control of vehicle-manipulator systems. 2nd edn. Springer, Berlin/New York
- Breivik M, Fossen TI (2008) Guidance laws for planar motion control. In: Proceedings of the 47th IEEE conference on decision & control, Cancun
- Casalino G, Simetti E, Turetta A (2009) A three-layered architecture for real time path planning and obstacle avoidance for surveillance usvs operating in harbour fields. In: Proceedings of the IEEE oceans’09-Europe, Bremen
- Delling D, Sanders P, Schultes D, Wagner D (2009) Engineering route planning algorithms. In Lerner J, Wagner D, Zweig KA (eds) Algorithmics of large and complex networks. Lecture notes in computer science, vol 5515. Springer, Berlin/New York, pp 117–139
- Fossen TI (2011) Handbook of marine crafts hydrodynamics and motion control. Wiley, Chichester
- Goerzen C, Kong Z, Mettler B (2010) A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *J Intell Robot Syst* 57: 65–100
- Greytak MB, Hover FS (2010) Robust motion planning for marine vehicle navigation. In: Proceedings of the 18th international offshore & polar engineering conference, Vancouver
- LaValle SM (2006) Planning algorithms. Cambridge University Press, Cambridge/New York. Available online at <http://planning.cs.uiuc.edu>. Accessed 6 June 2013
- Petres C, Pailhas Y, Patron P, Petillot Y, Evans J, Lane D (2007) Path planning for autonomous underwater vehicles. *IEEE Trans Robot* 23(2): 331–341
- Pettersen KY, Lefeber E (2001) Way-point tracking control of ships. In: Proceedings of the 40th IEEE conference decision & control, Orlando
- Sanz PJ, Ridao P, Oliver G, Melchiorri C, Casalino G, Silvestre C, Petillot Y, Turetta A (2010) TRIDENT: a framework for autonomous underwater intervention missions with dexterous manipulation capabilities. In: Proceedings of the 7th IFAC symposium on intelligent autonomous vehicles, Lecce
- Tam CK, Bucknall R (2010) Path-planning algorithm for ships in close-range encounters. *J Mar Sci Technol* 15:395–407
- Tam CK, Bucknall R, Greig A (2009) Review of collision avoidance and path planning methods for ships in close range encounters. *J Navig* 62:455–476
- Wang S, Meng Q (2012) Liner ship route schedule design with sea contingency time and port time uncertainty. *Transp Res B* 46:615–633

Motion Planning for PDEs

Thomas Meurer

Automatic Control Chair, Faculty of
Engineering, Kiel University, Kiel, Germany

Abstract

Motion planning or trajectory planning, respectively, refers to the design of an open-loop control to achieve desired trajectories for the system states or outputs. Given distributed parameter systems that are governed by partial differential equations (PDEs), this requires to take into account the spatial-temporal system dynamics. In this case flatness-based techniques provide a systematic motion planning approach, which is based on the parametrization of any system variable by means of a flat or basic output. With this, the motion planning problem can be solved rather intuitively as is illustrated subsequently for linear and semi-linear PDEs. In addition constraints on system input or states can be taken into account systematically by introducing a dynamic optimization problem for the computation of the flat output trajectory.

Keywords

Motion planning · Trajectory planning · Flatness · Formal integration · Trajectory assignment · Transition path · Gevrey class · Dynamic optimization · Spectral decomposition · Riesz spectral operators

Introduction

Motion planning or trajectory planning refers to the design of an open-loop control to realize prescribed desired temporal or spatial-temporal paths for the system states or outputs. Examples include smart structures with embedded actuators and sensors such as adaptive optics in telescopes; adaptive wings or smart

skins; thermal and reheating processes in steel industry and deep drawing; start-up, shutdown, or transitions between operating points in chemical engineering; as well as multi-agent deployment and formation control (see, e.g., the overview in Meurer 2013 or the applications in Schröck et al. 2013; Böhm and Meurer 2017; Kater and Meurer 2019).

For the solution of the motion planning and tracking control problem for finite-dimensional linear and nonlinear systems, differential flatness as introduced in Fliess et al. (1995) has evolved into a well-established inversion-based technique. Differential flatness implies that any system variable can be parametrized in terms of a flat output and its time derivatives up to a problem-dependent order. This yields the favorable property that the assignment of a suitable desired trajectory for the flat output directly provides the open-loop input trajectory required to realize the corresponding state trajectory and thus the prescribed motion. Moreover finite-dimensional nonlinear systems are exactly feedback linearizable by means of quasi-static feedback control (Fliess et al. 1999; Delaleau and Rudolph 1998).

The idea of parametrizing states and inputs can be adapted to systems governed by partial differential equations (PDEs). For this, different techniques have been developed utilizing operational calculus or spectral theory for linear PDEs, (formal) power series for PDEs involving polynomial nonlinearities, and formal integration for rather general semi-linear PDEs using a generalized Cauchy-Kowalevski approach. Recent extensions, e.g., address flatness-based constrained optimal control or model predictive of boundary controlled PDEs. To illustrate the principle ideas and the evolving research results starting with Fliess et al. (1997), in the following selected techniques are highlighted and are introduced based on example problems. The main part of the exposition is restricted to parabolic PDEs, but the application to motion planning for hyperbolic PDEs is briefly introduced before concluding with possible future research directions.

Linear PDEs

In the following, a scalar linear diffusion-reaction equation is considered in the state variable $x(z, t)$ with in-domain control $v(t)$ and boundary control $u(t)$ governed by

$$\partial_t x(z, t) = \partial_z^2 x(z, t) + rx(z, t) + b(z)v(t) \quad (1a)$$

$$\partial_z x(0, t) = 0, \quad x(1, t) = u(t) \quad (1b)$$

$$x(z, 0) = 0. \quad (1c)$$

This PDE describes a wide variety of thermal and fluid systems including heat conduction and tubular reactors. Herein, $r \in \mathbb{R}$ refer to the reaction coefficient, $b(z)$ denotes the in-domain input characteristics, and the initial state is without loss of generality assumed zero. To solve the motion planning problem for (1), a flatness-based feedforward control is determined to realize a finite-time transition between the initial state and a final stationary state $x_T^*(z)$ to be imposed for $t \geq T$. To illustrate two different approaches, it is subsequently distinguished between boundary and in-domain control.

Formal Integration

Given (1) with $v(t) \equiv 0$, i.e., the boundary controlled linear PDE, the application of formal integration as presented in Schörkhuber et al. (2013) is considered. By solving (1) for $\partial_z^2 x(z, t)$ and formally integrating twice with respect to z , the *implicit* expression

$$x(z, t) = x(0, t) + \int_0^z \int_0^p [\partial_t x(q, t) - rx(q, t)] dq dp \quad (2a)$$

$$u(t) = x(1, t), \quad (2b)$$

is obtained when taking into account the boundary conditions (1b). Expression (2) is the starting point to develop to a flatness-based design systematics for motion planning given boundary controlled PDEs. For this, introduce

$$y(t) = x(0, t), \quad (3)$$

and rewrite (2a) in terms of the sequence of functions $(x^{(n)}(z, t))_{n=0}^\infty$ according to

$$x^{(0)}(z, t) = y(t) \quad (4a)$$

$$x^{(n+1)}(z, t) = x^{(0)}(z, t) + \int_0^z \int_0^p [\partial_t x^{(n)}(q, t) - rx^{(n)}(q, t)] dq dp. \quad (4b)$$

This *explicit* iterative formulation enables us to deduced that $y(t)$ denotes a flat output differentially parametrizing the state variable $x(z, t) = \lim_{n \rightarrow \infty} x^{(n)}(z, t)$ and the boundary input $u(t) = x(1, t)$ provided that the limit exists as $n \rightarrow \infty$. It is worth noting that the iteration admits a solution in the form

$$\begin{aligned} x(z, t) &= \sum_{n=0}^{\infty} \frac{(\partial_t - r)^n}{(2n)!} \circ y(t) \\ &= \sum_{n=0}^{\infty} \frac{1}{(2n)!} \sum_{j=0}^n \binom{n}{j} (-r)^{n-j} \partial_t^j y(t). \end{aligned} \quad (5)$$

The respective boundary input $u(t)$ follows immediately from (2b). In particular, by prescribing a suitable trajectory $t \mapsto y^*(t) \in C^\infty(\mathbb{R})$ for $y(t)$, the evaluation of (4) or (5), respectively, yields the spatial-temporal path $x^*(z, t)$ that is realized by the application of the feedforward control $u^*(t) = x^*(1, t)$. This relies on the uniform convergence of (4) or (5), respectively, with at least a unit radius of convergence in z . For this, the notion of a Gevrey class function is needed (Rodino 1993).

Definition 1 (Gevrey class) The function $y(t)$ is in $G_{D,\alpha}(\Lambda)$, the Gevrey class of order α in $\Lambda \subseteq \mathbb{R}$, if $y(t) \in C^\infty(\Lambda)$, and for every closed subset Λ' of Λ , there exists a $D > 0$ such that $\sup_{t \in \Lambda'} |\partial_t^n y(t)| \leq D^{n+1} (n!)^\alpha$.

The set $G_{D,\alpha}(\Lambda)$ forms a linear vector space and a ring with respect to the arithmetic product of functions which is closed under the standard rules of differentiation. Gevrey class functions of order $\alpha < 1$ are entire and are analytic if $\alpha = 1$.

Theorem 1 *Let $y(t) \in G_{D,\alpha}(\mathbb{R})$ for $\alpha < 2$, then the sequence $(x^{(n)}(z, t))_{n=0}^{\infty}$ with coefficients (4) converges uniformly for any $z \geq 0$.*

The proof of this result is a special case of the general treatise in Schörkhuber et al. (2013) and relies on the analysis of the iteration (4) taking into account the assumptions on the function $y(t)$.

Spectral Approach

For (1) with $u(t) \equiv 0$, i.e., in-domain control only, the spectral design approach presented in Meurer (2011, 2013) is illustrated. It should be pointed out that this approach is applicable also to the boundary controlled or the mixed boundary and in-domain controlled PDE and linear PDEs with higher-dimensional spatial domain. By introducing the state space $X = L^2([0, 1])$ and the state variable $x(t) = x(\cdot, t) \in X$, system (1) with $u(t) \equiv 0$ can be rewritten in abstract form as

$$\partial_t x(t) = \mathfrak{A}x(t) + \mathfrak{B}v(t), \quad t > 0 \quad (6a)$$

$$\begin{aligned} x(0) &= x_0 \in \mathcal{D}(\mathfrak{A}) \\ &= \{x \in X \mid \partial_z^2 x \in X, \partial_z x(0) = x(1) = 0\}. \end{aligned} \quad (6b)$$

Herein, $\mathfrak{A}x = \partial_z^2 x + rx$ is the system operator with domain $\mathcal{D}(\mathfrak{A})$, and $\mathfrak{B} = b(z)$ is the input operator. In particular, $\mathfrak{A}$ denotes a so-called Riesz spectral operator, whose eigenfunctions $\phi(z) \in \mathcal{D}(\mathfrak{A})$ and adjoint eigenfunctions $\psi(z) \in \mathcal{D}(\mathfrak{A})$ form an orthonormal Riesz basis for X . Moreover, $\mathfrak{A}$ generates a C_0 -semigroup, whose Laplace transform, i.e., the resolvent operator $R(s, \mathfrak{A}) = (s\mathfrak{I} - \mathfrak{A})^{-1} = \sum_{k \in \mathbb{N}} (s - \lambda_k)^{-1} \langle \cdot, \psi_k(z) \rangle \phi_k(z)$ with identity operator $\mathfrak{I}$ and $L^2([0, 1])$ inner product $\langle \cdot, \cdot \rangle$, exists for $s > \sup_{k \in \mathbb{N}} \Re\{\lambda_k\}$ with $\{\lambda_k\}_{k \in \mathbb{N}}$ the eigenvalues of $\mathfrak{A}$. Applying the Laplace transform to (6) provides

$$\begin{aligned} X(s) &= R(s, \mathfrak{A})\mathfrak{B}V(s) \\ &= \sum_{k \in \mathbb{N}} \frac{1}{s - \lambda_k} \langle \mathfrak{B}V(s), \psi_k(z) \rangle \phi_k(z) \end{aligned} \quad (7)$$

For the considered case, the eigenproblem reads $\mathfrak{A}\phi(z) = \lambda\phi(z)$, $\phi(z) \in \mathcal{D}(\mathfrak{A})$ and yields the solution $\lambda_k = r - \mu_k^2$, $\phi_k(z) = \sqrt{2} \cos(\mu_k z)$ for $\mu_k = (2k - 1)\pi/2$ and $\psi_k(z) = \phi_k(z)$ since $\mathfrak{A}$ is self-adjoint.

The formulation (7) admits the determination of a flatness-based state and input parametrization by introducing a flat output. For this, (7) is rewritten according to

$$\begin{aligned} X(s) &= - \sum_{k \in \mathbb{N}} \frac{1}{\lambda_k} \frac{1}{1 - \frac{s}{\lambda_k}} \langle \mathfrak{B}, \psi_k(z) \rangle \phi_k(z) V(s) \\ &= - \sum_{k \in \mathbb{N}} \frac{1}{\lambda_k} \frac{\prod_{j \in \mathbb{N}, j \neq k} (1 - \frac{s}{\lambda_j})}{\prod_{j \in \mathbb{N}} (1 - \frac{s}{\lambda_j})} \langle \mathfrak{B}, \psi_k(z) \rangle \phi_k(z) V(s). \end{aligned} \quad (8)$$

Formally introducing the variable $Y(s)$ by

$$\begin{aligned} V(s) &= \mathcal{D}_v(s)Y(s) \Rightarrow X(s) \\ &= - \sum_{k \in \mathbb{N}} \frac{1}{\lambda_k} \langle \mathfrak{B}, \psi_k(z) \rangle \phi_k(z) \mathcal{D}_k^x(s) Y(s) \end{aligned} \quad (9)$$

with $\mathcal{D}^v(s) = \prod_{j \in \mathbb{N}} (1 - \frac{s}{\lambda_j})$ and $\mathcal{D}_k^x(s) = \prod_{j \in \mathbb{N}, j \neq k} (1 - \frac{s}{\lambda_j})$ shows that $Y(s)$ is a flat output parametrizing state and input in the operational domain. Herein, $\mathcal{D}^v(s)$ and $\mathcal{D}_k^x(s)$ can

be interpreted as (Hadamard) factorizations of entire functions in the complex domain $s \in \mathbb{C}$ (Boas 1954). The time-domain representation follows immediately by taking into account that s corresponds to time differentiation and the fact that an entire function admits a MacLaurin series expansion (Expressions are for the sake of simplicity in the following only written for the input parametrization. The state parametrization follows accordingly.) $\mathcal{D}_v(s) = \sum_{j \in \mathbb{N}} c_j s^{j-1}$ so that

$$v(t) = \mathcal{D}_v(\partial_t) \circ y(t) = \sum_{j \in \mathbb{N}} c_j \partial_t^{j-1} y(t). \quad (10a)$$

These results rely on the uniform convergence of (10), which can be guaranteed under certain conditions for $\mathcal{D}_v(s)$ and the Gevrey regularity of $y(t)$.

Theorem 2 *Let $(\lambda_k)_{k \in \mathbb{N}}$ be the sequence of disjoint eigenvalues of the Riesz spectral operator $\mathfrak{A}$, and let this sequence be of convergence exponent γ . Then $\mathcal{D}_v(s)$ is an entire function of order $\varrho = \gamma$. If in addition $\mathcal{D}_v(s)$ is of finite type, then (10) converges uniformly for $y(t) \in G_{D,\alpha}(\mathbb{R})$ with $\alpha < 1/\varrho$.*

For the proof of this result and the required notions from complex analysis, the reader is referred to Meurer (2013, Theorem 2.4). Given (1) or (6), respectively, with the eigenvalues discussed before, the convergence exponent of $(\lambda_k)_{k \in \mathbb{N}} = (r - ((2k-1)\pi/2)^2)_{k \in \mathbb{N}}$ is $\gamma = 1/2$, and $\mathcal{D}_v(s) = \cos(\sqrt{r-s})/\cos(\sqrt{r})$ is an entire function of order $\varrho = 1/2$ and finite type so that convergence is ensured for Gevrey class functions $y(t)$ of order $\alpha < 2$. The favorable property of this design approach is – besides its applicability to rather general linear PDE problems involving coupled systems and higher-dimensional spatial domains – the possibility to directly include numerical approximation techniques for the determination for the eigenvalue distribution to obtain a very efficient flatness-based design methodology.

Trajectory Assignment

To apply these results for the solution of the motion planning problem to achieve finite-time transitions between stationary profiles, it is crucial to properly assign the desired trajectory $y^*(t)$ for the basic output $y(t)$. For this, observe that stationary profiles $x^s(z) = x^s(z; y^s)$ are due to the flatness property (Classically stationary solutions are to be defined in terms of stationary input values $x^s(1) = u^s$.) for the boundary controlled setup with $v(t) \equiv 0$ governed by

$$0 = \partial_z^2 x^s(z) + r x^s(z) \quad (11a)$$

$$\partial_z x^s(0) = 0, \quad x^s(0) = y^s. \quad (11b)$$

Given the in-domain control for $u(t) \equiv 0$, stationary profiles are with $v^s = y^s$ (apply the final value theorem to (9) or directly consider (10)) obtained by solving

$$0 = \partial_z^2 x^s(z) + r x^s(z) + b(z) y^s \quad (12a)$$

$$\partial_z x^s(0) = 0, \quad x^s(0) = 0. \quad (12b)$$

For both scenarios the assignment of different y^s results in different stationary profiles $x^s(z; y^s)$. The connection between an initial stationary profile $x_0^s(z; y_0^s)$ and a final stationary profile $x_T^s(z; y_T^s)$ is achieved by assigning $y^*(t)$ such that

$$\begin{aligned} y^*(0) &= y_0^s, & y^*(T) &= y_T^s \\ \partial_t^n y^*(0) &= 0, & \partial_t^n y^*(T) &= 0, \quad n \geq 1. \end{aligned}$$

This implies that $y^*(t)$ has to be locally nonanalytic at $t \in \{0, T\}$ and in view of the previous discussion a Gevrey class function of order $\alpha \in (1, 2)$. For specific problems different functions have been suggested fulfilling these properties. In the following, the ansatz

$$y^*(t) = y_0^s + (y_T^s - y_0^s) \Phi_{T,\gamma}(t) \quad (13a)$$

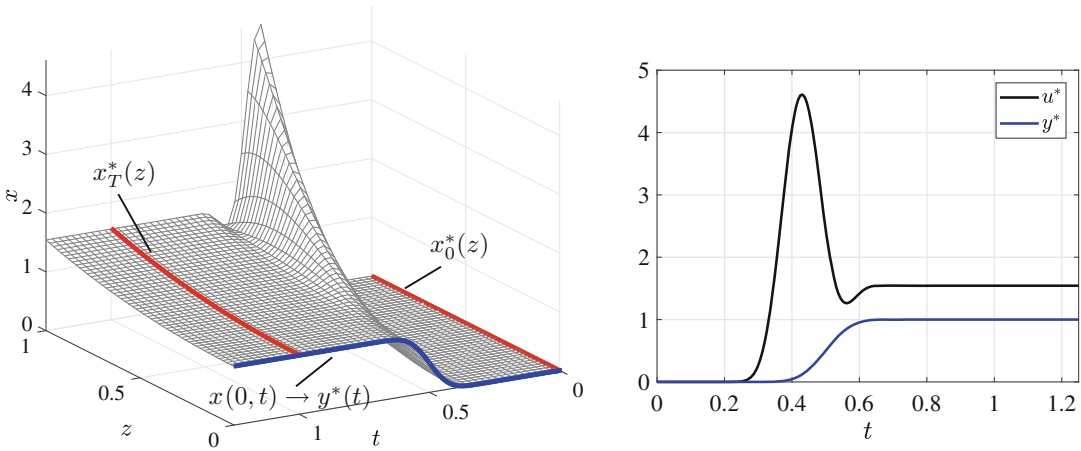
is used with

$$\Phi_{T,\gamma}(t) = \begin{cases} 0, & t \leq 0 \\ \frac{\int_0^t h_{T,\gamma}(\tau) d\tau}{\int_0^T h_{T,\gamma}(\tau) d\tau} & t \in (0, T) \\ 1, & t \geq T \end{cases} \quad (13b)$$

for $h_{T,\gamma}(t) = \exp(-[t/T(1-t/T)]^{-\gamma})$ if $t \in (0, T)$ and $h_{T,\gamma}(t) = 0$ else. It can be shown that (13b) is a Gevrey class function of order $\alpha = 1 + 1/\gamma$ (Fliess et al. 1997). Alternative function are presented, e.g., in Rudolph (2003b).

Simulation Example

In order to illustrate the results of the motion planning procedure described above, let $r = -1$ in (1).



Motion Planning for PDEs, Fig. 1 Simulated spatial-temporal transition path (left) and applied flatness-based feedforward control $u^*(t)$ and desired trajectory $y^*(t)$ (right) for (1) with boundary control

Boundary Control Using Formal Integration

The differential parametrization (4) is evaluated for the desired trajectory $y^*(t)$ defined in (13) for $y_0^s = 0$ and $y_T^s = 1$ with the transition time $T = 1$ and the parameter $\gamma = 2$. With this, the finite-time transition between the zero initial stationary profile $x_0^*(z) = 0$ and the final stationary profile $x_T^*(z) = x_T^s(z) = y_T^s \cosh(z)$ is realized along the trajectory $x(0,t) = y^*(t)$. The corresponding feedforward control and spatial-temporal transition path are shown in Fig. 1.

In-Domain Control Using Spectral Approach

The determined eigenvalues are taken into account to evaluate the differential parametrization (10) for the desired trajectory $y^*(t)$ defined in (13) for $y_0^s = 0$ and $y_T^s = 20$ with the transition time $T = 0.5$ and the parameter $\gamma = 2$. The spatial input characteristics is chosen asymmetrically with $b(z) = \sigma(z - 0.3) - \sigma(z - 0.5)$ and $\sigma(\cdot)$ the Heaviside function. The resulting finite-time transition from the zero initial stationary profile $x_0^*(z) = 0$ to the final stationary profile $x_T^*(z) = x_T^s(z)$ and the computed feedforward control are depicted in Fig. 2.

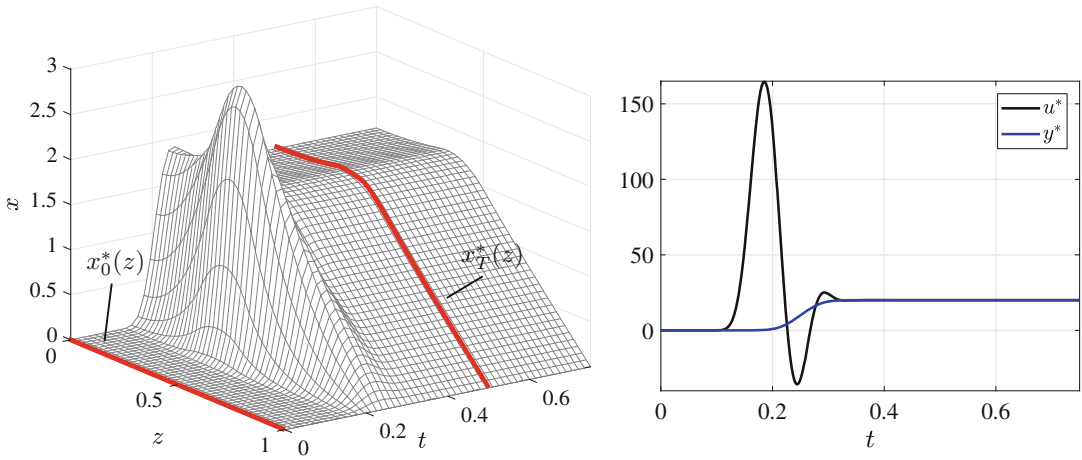
Extensions and Generalizations

The previous considerations provide a first introduction to systematic approaches for the

solution of the motion planning problem for systems governed by PDEs. The techniques can be further generalized to address coupled systems of PDEs, certain classes of nonlinear PDEs (see also section “Semi-linear PDEs”), joint boundary and in-domain control, or PDEs with higher-dimensional spatial domain (Rudolph 2003a; Meurer 2011, 2013).

Constrained motion planning and optimal control have been recently addressed by introducing a dynamic optimization problem based on an integrator chain for computation of the flat output trajectory (Andrej and Meurer 2018). Herein, the highest derivative serves as decision variable so that in view of the differential state and input parametrization in terms of the flat output or the states of the integrator chain, respectively, the constrained optimal control problem can be completely expressed in terms of the new decision variable. An extension to flatness-based model predictive control for PDEs is available in Meurer and Andrej (2018).

Besides the many theoretical contributions experimental results for flexible beam and plate structures with embedded piezoelectric actuators (Schröck et al. 2013; Kater and Meurer 2019) and for thermal problems in deep drawing (Böhm and Meurer 2017), confirm the applicability of this design approach and the achievable high tracking accuracy when transiently shaping the deflection



Motion Planning for PDEs, Fig. 2 Simulated spatial-temporal transition path (left) and applied flatness-based feedforward control $u^*(t)$ and desired trajectory $y^*(t)$ (right) for (1) with in-domain control

profile or controlling the spatial-temporal temperature distribution during operation.

Semi-linear PDEs

Flatness can be extended to semi-linear PDEs. This is subsequently illustrated for the diffusion-reaction system

$$\partial_t x(z, t) = \partial_z^2 x(z, t) + r(x(z, t)) \quad (14a)$$

$$\partial_z x(0, t) = 0, \quad x(1, t) = u(t) \quad (14b)$$

$$x(z, 0) = 0 \quad (14c)$$

with boundary input $u(t)$. Similar to the previous section, the motion planning problem refers to the determination of a feedforward control $t \mapsto u^*(t)$ to realize finite-time transitions starting at the initial profile $x_0^*(z) = x(z, 0) = 0$ to achieve a final stationary profile $x_T^*(z)$ for $t \geq T$.

Formal Integration

The generalization of the formal integration approach to general semi-linear PDEs is addressed in Schörkhuber et al. (2013) by making use of an abstract Cauchy-Kowalevski theorem in Gevrey classes. Proceeding as for the determination of (2) yields the *implicit* solution

$$x(z, t) = x(0, t)$$

$$+ \int_0^z \int_0^p [\partial_t x(q, t) - r(x(q, t))] dq dp \quad (15a)$$

$$u(t) = x(1, t), \quad (15b)$$

which can be used to develop a flatness-based design systematics for motion planning given semi-linear PDEs. For this, introduce

$$y(t) = x(0, t), \quad (16)$$

and rewrite (15b) in *explicit* form by means of the sequence of functions $(x^{(n)}(z, t))_{n=0}^\infty$ fulfilling

$$x^{(0)}(z, t) = y(t) \quad (17a)$$

$$x^{(n+1)}(z, t) = x^{(n)}(z, t) + \int_0^z \int_0^p [\partial_t x^{(n)}(q, t) - r(x^{(n)}(q, t))] dq dp. \quad (17b)$$

Obviously, $y(t)$ denotes a flat output, which differentially parametrizes $x(z, t) = \lim_{n \rightarrow \infty} x^{(n)}(z, t)$ and $u(t) = x(1, t)$ provided that the limit exists as $n \rightarrow \infty$. As is shown in Schörkhuber et al. (2013), by making use of scales of Banach spaces in Gevrey classes

and abstract Cauchy-Kowalevski theory, the convergence of the parametrized sequence of functions $(x^{(n)}(z, t))_{n=0}^{\infty}$ can be ensured in some compact subset of the domain $z \in [0, 1]$. Besides its general set-up, this approach provides an iteration scheme, which can be directly utilized for a numerically efficient solution of the motion planning problem.

Simulation Example

Let the reaction be subsequently described by

$$r(x(z, t)) = \sin(2\pi x(z, t)). \quad (18)$$

The iterative scheme (17) is evaluated for the desired trajectory $y^*(t)$ defined in (13) for $y_0^s = 0$ and $y_T^s = 1$ with the transition time $T = 1$ and the parameter $\gamma = 1$, i.e., the desired trajectory is of Gevrey order $\alpha = 2$. The resulting feedforward control $u^*(t)$ and the spatial-temporal transition path resulting from the numerical solution of the PDE are depicted in Fig. 3. The desired finite-time transition between the zero initial stationary profile $x_0^*(z) = 0$ and the final stationary profile $x_T^*(z) = x_T^s(z)$ determined by

$$0 = \partial_z^2 x^s(z) + r(x^s(z)) \quad (19a)$$

$$\partial_z x^s(0) = 0, \quad x^s(0) = y^s. \quad (19b)$$

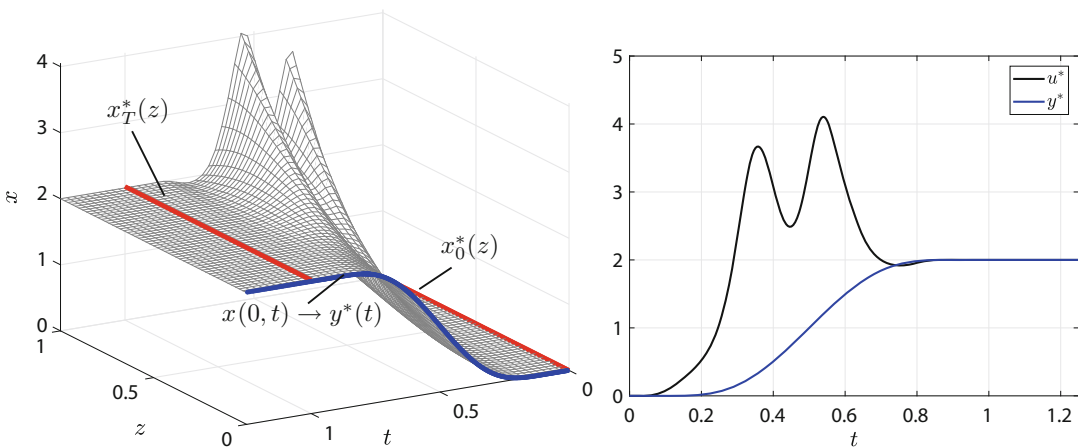
is clearly achieved along the prescribed path $y^*(t)$.

Extensions and Generalizations

Generalizations of the introduced formal integration approach to solve motion planning problems for systems of coupled PDEs are, e.g., provided in Schörkhuber et al. (2013). Moreover, linear diffusion-convection-reaction systems with spatially and time-varying coefficients defined on a higher-dimensional parallelepipedon are addressed, e.g., in Meurer and Kugi (2009) and Meurer (2013). Novel applications to address motion planning for the deployment and formation control of multi-agent continua using linear, semi- and quasi-linear PDEs are, e.g., considered in Meurer and Krstic (2011), Qi et al (2015), and Freudenthaler and Meurer (2016).

Hyperbolic PDEs

Hyperbolic PDEs exhibiting wavelike dynamics require the development of a design systematics explicitly taking into account the finite speed of wave propagation. For linear hyperbolic PDEs, operational calculus has been successfully applied to determine the state and input



Motion Planning for PDEs, Fig. 3 Simulated spatial-temporal transition path (left) and applied flatness-based feedforward control $u^*(t)$ and desired trajectory $y^*(t)$ (right) for (14) with (18)

parametrizations in terms of the flat output and its advanced and delayed arguments (Rouchon 2001; Petit and Rouchon 2001, 2002; Woittennek and Rudolph 2003; Rudolph and Woittennek 2008). In addition, the method of characteristics can be utilized to address both linear as well as quasi-linear hyperbolic PDEs. Herein, a suitable change of coordinates enables to reformulate the PDE in a normal form, which can be (formally) integrated in terms of a basic output. With this, also an efficient numerical procedure can be developed to solve motion planning problems for hyperbolic PDEs (Woittennek and Mounier 2010; Knüppel et al 2010).

Summary and Future Directions

Motion planning constitutes an important design step when solving control problems for systems governed by PDEs. This is particularly due to the increasing demands on quality, accuracy, and efficiency, which require to turn away from the pure stabilization of an operating point towards the realization of specific start-up, transition, or tracking tasks. It is a favorable property of the above introduced techniques that once the parametrization of states and inputs in terms of the flat output is computed, the numerical evaluation can be performed on the fly, which is particularly useful in online evaluations under real-time constraints. In view of these aspects, future research directions might deepen and further evolve

- semi-analytic design techniques taking into account suitable approximation and model order reduction schemes for complex-shaped spatial domains,
- nonlinear PDEs and coupled systems of nonlinear PDEs with boundary and in-domain control, and
- applications arising, e.g., in multi-agent systems, swarm dynamics, aeroelasticity, fluid flow, and fluid-structure interaction.

Cross-References

- Backstepping for PDEs
- Boundary Control of 1-D Hyperbolic Systems
- Boundary Control of Korteweg-de Vries and Kuramoto-Sivashinsky PDEs
- Control of Fluid Flows and Fluid-Structure Models
- Regulation and Tracking of Nonlinear Systems
- Tracking and Regulation in Linear Systems

Bibliography

- Andrej J, Meurer T (2018) Flatness-based constrained optimal control of reaction-diffusion systems. In: American control conference (ACC), Milwaukee, pp 2539–2544
- Boas R (1954) Entire functions. Academic, New York
- Böhm T, Meurer T (2017) Trajectory planning and tracking control for the temperature distribution in a deep drawing tool. *Control Eng Pract* 64: 127–139
- Delaleau E, Rudolph J (1998) Control of flat systems by quasi-static feedback of generalized states. *Int J Control* 71(5):745–765 <https://doi.org/10.1080/002071798221551>
- Fliess M, Lévine J, Martin P, Rouchon P (1995) Flatness and defect of non-linear systems: introductory theory and examples. *Int J Control* 61:1327–1361
- Fliess M, Mounier H, Rouchon P, Rudolph J (1997) Systèmes linéaires sur les opérateurs de Mikusiński et commande d'une poutre flexible. *ESAIM Proceedings* 2:183–193
- Fliess M, Lévine J, Martin P, Rouchon P (1999) A Lie-Bäcklund approach to equivalence and flatness of nonlinear systems. *IEEE Trans Autom Control* 44(5): 922–937
- Freudenthaler G, Meurer T (2016) PDE-based tracking control for multi-agent deployment. *IFAC-PapersOnLine* 49(18):582–587; 10th IFAC symposium on nonlinear control systems (NOLCOS 2016), Monterey, 23–25 Aug 2016
- Kater A, Meurer T (2019) Motion planning and tracking control for coupled flexible beam structures. *Control Eng Pract* 84:389–398
- Knüppel T, Woittennek F, Rudolph J (2010) Flatness-based trajectory planning for the shallow water equations. In: Proceedings of 49th IEEE conference on decision and control (CDC), Atlanta, pp 2960–2965

- Meurer T (2011) Flatness-based trajectory planning for diffusion-reaction systems in a Parallelepipedon – a spectral approach. *Automatica* 47(5): 935–949
- Meurer T (2013) Control of higher-dimensional PDEs: flatness and backstepping designs. Communications and control engineering series. Springer
- Meurer T, Andrej J (2018) Flatness-based model predictive control of linear diffusion-convection-reaction processes. In: IEEE conference on decision and control (CDC), Miami Beach, pp 527–532
- Meurer T, Krstic M (2011) Finite-time multi-agent deployment: a nonlinear PDE motion planning approach. *Automatica* 47(11):2534–2542
- Meurer T, Kugi A (2009) Trajectory planning for boundary controlled parabolic PDEs with varying parameters on higher-dimensional spatial domains. *IEEE T Automat Control* 54(8):1854–1868
- Petit N, Rouchon P (2001) Flatness of heavy chain systems. *SIAM J Control Optim* 40(2):475–495
- Petit N, Rouchon P (2002) Dynamics and solutions to some control problems for water-tank systems. *IEEE Trans Autom Control* 47(4):594–609
- Qi J, Vazquez R, Krstic M (2015) Multi-agent deployment in 3-D via PDE control. *IEEE Trans Autom Control* 60(4):891–906
- Rodino L (1993) Linear partial differential operators in Gevrey spaces. World Scientific Publishing Co. Pte. Ltd., Singapore
- Rouchon P (2001) Motion planning, equivalence, and infinite dimensional systems. *Int J Appl Math Comp Sci* 11:165–188
- Rudolph J (2003a) Beiträge zur flachheitsbasierten Folgeregung linearer und nichtlinearer Systeme endlicher und unendlicher Dimension. Berichte aus der Steuerungs- und Regelungstechnik, Shaker-Verlag, Aachen
- Rudolph J (2003b) Flatness Based Control of Distributed Parameter Systems. Berichte aus der Steuerungs- und Regelungstechnik, Shaker-Verlag, Aachen
- Rudolph J, Woittennek F (2008) Motion planning and open loop control design for linear distributed parameter systems with lumped controls. *Int J Control* 81(3):457–474
- Schörkhuber B, Meurer T, Jüngel A (2013) Flatness of semilinear parabolic PDEs – a generalized Cauchy-Kowalevski approach. *IEEE T Automat Control* 58(9):2277–2291
- Schröck J, Meurer T, Kugi A (2013) Motion planning for Piezo-actuated flexible structures: modeling, design, and experiment. *IEEE T Control Sys Tech* 21(3): 807–819
- Woittennek F, Mounier H (2010) Controllability of networks of spatially one-dimensional second order p.d.e. – an algebraic approach. *SIAM J Control Optim* 48(6):3882–3902
- Woittennek F, Rudolph J (2003) Motion planning for a class of boundary controlled linear hyperbolic PDE's involving finite distributed delays. *ESAIM: Control Optim Calcul Var* 9:419–435

Motorcycle Dynamics and Control

Martin Corless

School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, USA

Abstract

A basic model due to Sharp which is useful in the analysis of motorcycle behavior and control is developed. This model is based on linearization of a bicycle model introduced by Whipple, but is augmented with a tire model in which the lateral tire force depends in a dynamic fashion on tire behavior. This model is used to explain some of the important characteristics of motorcycle behavior. The significant dynamic modes exhibited by this model are capsize, weave, and wobble.

Keywords

Motorcycle · Tire model · Single-track vehicle · Bicycle · Weave · Wobble · Capsize · Counter-steering

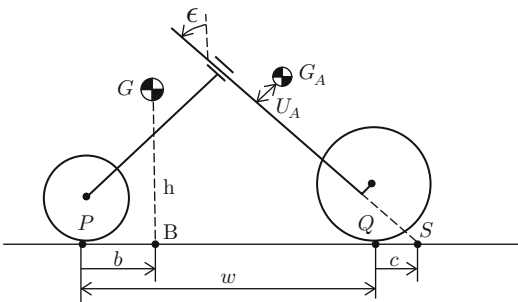
Introduction

The bicycle is mankind's ultimate solution to the quest for a human-powered vehicle (Herlihy 2006; Limebeer and Sharp 2006). The motorcycle just makes riding more fun. Bicycles, motorcycles, scooters, and mopeds are all examples of single-track vehicles and have similar dynamics. The dynamics of a motorcycle are considerably more complicated than that of a four-wheel vehicle such as a car. The first obvious difference in behavior is stability. An unattended upright stationary motorcycle is basically an inverted pendulum and is unstable about its normal upright position, whereas a car has no stability issues in the same configuration. Another difference is that a motorcycle must lean when cornering. Although a car leans a little due to suspension travel, there is no necessity for it to lean in

cornering. A perfectly rigid car would not lean. Furthermore, beyond low speeds, the steering behavior of a motorcycle is not intuitive like that of a car. To turn a car right, the driver simply turns the steering wheel right; on a motorcycle the rider initially turns the handlebars to the left. This is called counter-steering and is not intuitive.

A Basic Model

To obtain a basic motorcycle model, we start with four rigid bodies: the **rear frame** (which includes a rigidly attached rigid rider), the **front frame** (includes handlebars and front forks), the **rear wheel**, and the **front wheel**; see Fig. 1. We assume that both frames and wheels have a plane of symmetry which is vertical when the bike is in its nominal upright configuration. The front frame can rotate relative to the rear frame about the **steering axis**; the steering axis is in the plane of symmetry of each frame, and, in the nominal upright configuration of the bike, the angle it makes with the vertical is called the **rake angle** or **caster angle** and is denoted by ϵ . The rear wheel rotates relative to the rear frame about an axis perpendicular to the rear plane of symmetry and is symmetrical with respect to this axis. The same relationship holds between the front wheel and the front frame. Although each wheel can be three-dimensional, we model the wheels as terminating in a knife edge at their boundaries and contact the ground at a single point. Points



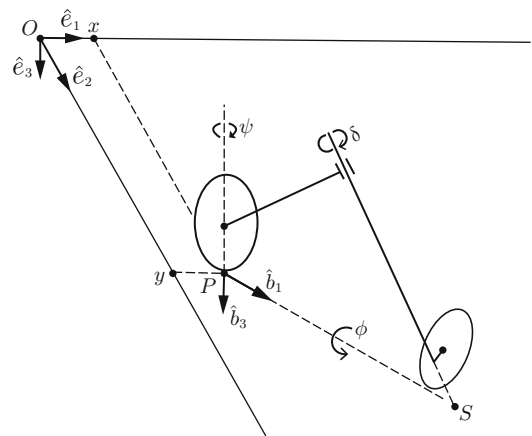
Motorcycle Dynamics and Control, Fig. 1 Basic model

P and Q are the points on the ground in contact with the rear and front wheels, respectively.

Each of the above four bodies is described by their mass, mass center location, and a 3×3 inertia matrix. Two other important parameters are the **wheelbase** w and the **trail** c . The wheelbase is the distance between the contact points of the two wheels in the nominal configuration, and the trail is the distance from the front wheel contact point Q to the intersection S of the steering axis with the ground. The trail is normally positive, that is, Q is behind S . The point G locates the mass center of the complete bike in its nominal configuration, whereas G_A is the location of the mass center of the **front assembly** (front frame and wheel).

Description of Motion

Considering a right-handed reference frame $e = (\hat{e}_1, \hat{e}_2, \hat{e}_3)$ with origin O fixed in the ground, the bike motion can be described by the location of the rear wheel contact point P relative to O , the orientation of the rear frame relative to e , and the orientation of the front frame relative to the rear frame; see Fig. 2. Assuming the bike is moving along a horizontal plane, the location of P is usually described by Cartesian coordinates x and y . Let reference frame b be fixed in the rear frame



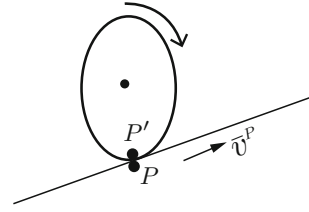
Motorcycle Dynamics and Control, Fig. 2 Description of motion

with $\hat{b}_1$ and $\hat{b}_3$ in the plane of symmetry with $\hat{b}_1$ along the $P - S$ line; see Fig. 2. Using this reference frame, the orientation of the rear frame is described by a 3-1-2 Euler angle sequence which consists of a **yaw** rotation by ψ about the 3-axis followed by a **lean** (roll) rotation by ϕ about the 1-axis and finally by a **pitch** rotation by θ about the 2-axis. The orientation of the front frame relative to the rear frame can be described by the **steer angle** δ . Assuming both wheels remain in contact with the ground, the pitch angle θ is not independent; it is uniquely determined by δ and ϕ . In considering small perturbations from the upright nominal configuration, the variation in pitch is usually ignored. Here we consider it to be zero. Also the dynamic behavior of the bike is independent of the coordinates x , y , and ψ . These coordinates can be obtained by integrating the velocity of P and $\dot{\psi}$.

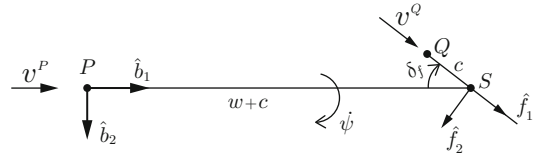
The Whipple Bicycle Model

The “simplest” model which captures all the salient features of a single-track vehicle for a basic understanding of low-speed dynamics and control is that originally due to Whipple (Whipple 1899). We consider the linearized version of this model which is further expounded on in Meijaard et al. (2007). The salient feature of this model is that there is no **slip** at each wheel. This means that the velocity of the point on the wheel which is instantaneously in contact with the ground is zero; this is point P' in Fig. 3. No slip implies that there is no **sideslip** which means that the velocity of the wheel contact point ($\bar{v}^P$ in Fig. 3) is parallel to the intersection of the wheel plane with the ground plane; the wheel contact point is the point moving along the ground which is in contact with the wheel.

The rest of this article is based on linearization of motorcycle dynamics about an equilibrium configuration corresponding to the bike traveling upright in a straight line at constant forward speed $v := v^P$, the speed of the rear wheel contact point P . In the linearized system, the longitudinal dynamics are independent of the



Motorcycle Dynamics and Control, Fig. 3 No slip: $v^{P'} = 0$



Motorcycle Dynamics and Control, Fig. 4 Some kinematics

lateral dynamics, and, in the absence of driving or braking forces, the speed v is constant.

With no sideslip at both wheels, kinematical considerations (see Fig. 4) show that the yaw rate $\dot{\psi}$ is determined by δ ; specifically for small angles we have the following linearized relationship:

$$\dot{\psi} = v\delta + \mu\dot{\delta} \quad (1)$$

where

$$v = c_\epsilon/w, \quad \mu = cc_\epsilon/w, \quad c_\epsilon = \cos \epsilon. \quad (2)$$

In Fig. 4, δ_f is the effective steer angle, the angle between the intersections of the front wheel plane and the rear frame plane with the ground. It is approximately given by $\delta_f = \delta c_\epsilon$. Thus, we can completely describe the lateral bike dynamics with the roll angle ϕ and the steer angle δ . To obtain the above relationship, first note that, as a consequence of no sideslip, $\bar{v}^P = v\hat{b}_1$ and $\bar{v}^Q$ is perpendicular to $\hat{f}_2$. Taking the dot product of the expression,

$$\bar{v}^Q = \bar{v}^P + (w+c)\dot{\psi}\hat{b}_2 - c(\dot{\psi} + \dot{\delta}_f)\hat{f}_2$$

with $\hat{f}_2$ while noting that $\hat{b}_1 \cdot \hat{f}_2 = -\sin \delta_f$ and $\hat{b}_2 \cdot \hat{f}_2 = \cos \delta_f$ results in

$$0 = -v \sin \delta_f + (w+c)\dot{\psi} \cos \delta_f - c(\dot{\psi} + \dot{\delta}_f).$$

Linearization about $\delta = 0$ yields the desired result.

The relationship in (1) also holds for four-wheel vehicles. There one can achieve a desired constant yaw rate $\dot{\psi}_d$ by simply letting the steer angle $\delta = \dot{\psi}_d/vv$. However, as we shall see, a motorcycle with its steering fixed at a constant steer angle is unstable. Neglecting gyroscopic terms, it is simply an inverted pendulum. With its steering free, a motorcycle can be stable over a certain speed range and, if unstable, can be easily stabilized above a very low speed by most people. Trials riders can stabilize a motorcycle at any speed including zero.

In developing our mathematical model, we initially ignore the mass and inertia of the front assembly along with gyroscopic effects, and we assume that the $\hat{b}_1$ and $\hat{b}_3$ axes are the principal axes of inertia of the bike/rider with moments of inertia I_{xx} and I_{zz} , respectively. Angular momentum considerations yield that

$$\frac{d\tilde{H}^P}{dt} + m\tilde{r}^{PG} \times \tilde{a}^P = \Sigma \tilde{M}^P \quad (3)$$

where $\tilde{H}^P \approx I_{xx}\ddot{\phi}\hat{b}_1 + I_{zz}\ddot{\psi}\hat{b}_3$ is the angular momentum of the rider/bike about point P , $\tilde{a}^P$ is the acceleration of P , $\tilde{r}^{PG} = b\hat{b}_1 - h\hat{b}_3$, and $\Sigma \tilde{M}^P$ is the sum of the external moments on rider/bike about P . Considering the roll component ($\hat{b}_1$ component) of (3) and linearization results in

$$I_{xx}\ddot{\phi} + mha = mgh\phi + (N_f c_\epsilon c)\delta \quad (4)$$

where N_f is the normal force (vertical and upward) on the front wheel and a is the lateral acceleration of point P . Considering the pitch component ($\hat{b}_2$ component) of (3), one can obtain that

$$N_f \approx mgb/w \quad (5)$$

which along with (4) yields

$$I_{xx}\ddot{\phi} - mgh\phi = -mha + \mu mgb\delta. \quad (6)$$

Cornering at Constant Speed

If the speed v and the steer angle δ are constant, then (1) implies that $\dot{\psi}$ is constant and equal to $v\delta$. Since $\dot{\psi}$ and v are constant, point P moves on a circle of radius $R = v/\dot{\psi}$. Using these relationships, one obtains that

$$\delta = 1/vR = w/c_\epsilon R.$$

Since P is moving in a circle of radius R at speed v , $a = v^2/R$. The behavior of (6) is that of an inverted pendulum subject to a constant torque equal to $-mha + \mu mgb\delta$. It is unstable about its equilibrium angle

$$\phi = \frac{a}{g} - \frac{\mu b\delta}{h} = \frac{v^2}{gR} - \frac{bc}{hR}.$$

Hence, except at small speeds, to corner with a lateral acceleration a , the rider/bike must lean into the corner at an angle of approximately a/g radians.

The Lean Equation

Using basic kinematics and recalling relationship (1),

$$a = v\dot{\psi} = v^2\delta + \mu v\dot{\delta}. \quad (7)$$

Equation (6) now yields the lean equation:

$$I_{xx}\ddot{\phi} + c_{\phi\delta}v\dot{\delta} - mgh\phi + (k_{\phi\delta} + k_{\phi\delta_2}v^2)\delta = 0, \quad (8)$$

where

$$\begin{aligned} c_{\phi\delta} &= \mu mh, & k_{\phi\delta} &= -S_A g, \\ k_{\phi\delta_2} &= vmh, & S_A &= \mu mb. \end{aligned}$$

Note that v is a constant parameter corresponding to the nominal speed of the rear wheel contact point.

With $\delta = 0$, we have a system whose behavior is characterized by two real eigenvalues: $\pm\sqrt{mgh/I_{xx}}$. This system is unstable due to the positive eigenvalue $\sqrt{mgh/I_{xx}}$. For v sufficiently large, the coefficient of δ in (8) is positive and one can readily show that the above system can be stabilized with positive feedback $\delta = K\phi$ provided $K > mgh/(k_{\phi\delta} + k_{\phi\delta_2}v^2)$. This helps

explain the stabilizing effect of a rider turning the handlebars in the direction the bike is falling. Actually, the rider input is a **steer torque** T_δ about the steer axis.

Taking into account the mass and inertia of the front assembly, gyroscopic effects, and crossproducts of inertia of the rear frame, one can show (see Meijaard et al. 2007) that the lean equation (8) still holds with the addition of a $m_{\phi\delta}\ddot{\delta}$ term, that is,

$$\boxed{I_{xx}\ddot{\phi} + m_{\phi\delta}\ddot{\delta} + c_{\phi\delta}v\dot{\delta} - mgh\phi + (k_{\phi\delta} + k_{\phi\delta_2}v^2)\delta = 0} \quad (9)$$

where

$$\begin{aligned} m_{\phi\delta} &= \mu I_{xz} + I_{A\epsilon x}, \\ c_{\phi\delta} &= \mu mh + v I_{xz} + \mu S_T + c_\epsilon S_F, \\ k_{\phi\delta} &= -S_A g, \\ k_{\phi\delta_2} &= v(mh + S_T), \\ S_A &= \mu mb + m_A u_A. \end{aligned}$$

Here I_{xx} is the moment of inertia of the total motorcycle and rider in the nominal configuration about the $\hat{b}_1$ axis and I_{xz} is the inertia crossproduct w.r.t the $\hat{b}_1$ and $\hat{b}_3$ axes. The term $I_{A\epsilon x}$ is the front assembly inertia crossproduct with respect to the steering axis and the $\hat{b}_1$ axis; see Meijaard et al. (2007) for a further description of this parameter. m_A is the mass of the front assembly (front wheel and front frame), and u_A is the offset of the mass center of the front assembly from the steering axis, that is, the distance of this mass center from the steering axis; see Fig. 1. The terms $S_F = I_{Fyy}/r_F$ and $S_T = I_{Ryy}/r_R + I_{Fyy}/r_F$ are **gyroscopic** terms due to the rotation of the front and rear wheels where r_F and r_R are the radii of the front and rear wheels, while I_{Fyy} and I_{Ryy} are the moments of inertias of the front and rear wheels about their axles. It is assumed that the mass center of each wheel is located at its geometric center.

The Steer Equation

Considering the yaw component ($\hat{b}_3$ component) of (3), one can obtain that

$$I_{zz}\ddot{\psi} + mba \approx wF^f \quad (10)$$

where F^f is the lateral force on the front wheel. Considering the sum of moments about the steering axis for the front assembly and linearization results in

$$T_\delta + \mu s_\epsilon w N_f \delta + \mu w N_f \phi - \mu w F^f = 0. \quad (11)$$

Combining (11), (10), (5), and (7) yields the steer equation:

$$m_{\delta\delta}\ddot{\delta} + c_{\delta\delta}v\dot{\delta} + k_{\delta\phi}\phi + (k_{\delta\delta} + k_{\delta\delta_2}v^2)\delta = T_\delta \quad (12)$$

where

$$\begin{aligned} m_{\delta\delta} &= \mu^2 I_{zz}, \quad c_{\delta\delta} = \mu (S_A + v I_{zz}), \\ k_{\delta\phi} &= -S_A g, \quad k_{\delta\delta} = -s_\epsilon S_A g, \\ k_{\delta\delta_2} &= v S_A, \quad S_A = \mu mb. \end{aligned}$$

Taking into account the mass and inertia of the front assembly, gyroscopic effects, and crossproducts of inertia of the rear frame, one can show (see Meijaard et al. 2007) that the steer equation (12) still holds with the addition of $m_{\delta\phi}\ddot{\phi}$ and $c_{\delta\phi}v\dot{\phi}$ terms, that is,

$$\boxed{m_{\delta\phi}\ddot{\phi} + m_{\delta\delta}\ddot{\delta} + v c_{\delta\phi}\dot{\phi} + v c_{\delta\delta}\dot{\delta} + k_{\delta\phi}\phi + (k_{\delta\delta} + k_{\delta\delta_2}v^2)\delta = T_\delta} \quad (13)$$

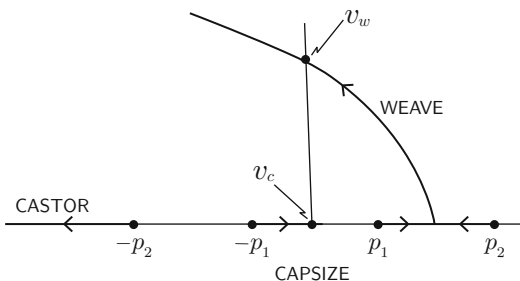
where

$$\begin{aligned} m_{\delta\phi} &= m_{\phi\delta}, \quad m_{\delta\delta} = \mu^2 I_{zz} + 2\mu I_{A\epsilon z} + I_{A\epsilon\epsilon}, \\ c_{\delta\phi} &= -(\mu S_T + c_\lambda S_F), \\ c_{\delta\delta} &= \mu (S_A + v I_{zz}) + v I_{A\epsilon z}, \\ S_A &= \mu mb + m_A u_A, \quad k_{\delta\phi} = k_{\phi\delta}, \\ k_{\delta\delta} &= -s_\epsilon S_A g, \quad k_{\delta\delta_2} = v(S_A + s_\epsilon S_F). \end{aligned}$$

Here, I_{zz} is the moment of inertia of the total motorcycle and rider in the nominal configuration about the $\hat{b}_3$ axis, $I_{A\epsilon\epsilon}$ is the moment of inertia of the front assembly about the steering axis, and $I_{A\epsilon z}$ is the front assembly inertia crossproduct with respect to the steering axis and the vertical axis through P . The lean equation (9) combined with the steer equation (12) provides an initial model for motorcycle dynamics. This is a linear model with the nominal speed v as a constant parameter and the rider's steering torque T_δ as an input.

Modes of the Whipple Model

With $v = 0$, the linearized Whipple model (9), (10), (11), and (12) has two pairs of real eigenvalues: $\pm p_1, \pm p_2$ with $p_2 > p_1 > 0$; see Fig. 5. The pair $\pm p_1$ roughly describes the inverted pendulum behavior of the whole bike with fixed steering. As v increases the real eigenvalues corresponding to p_1 and p_2 meet and from there on form a complex conjugate pair of eigenvalues which result in a single oscillatory mode called the **weave mode**. Initially the weave mode is unstable, but is stable above a certain speed v_w , and for large speeds, its eigenvalues are roughly a linear function of v ; thus, it becomes more damped and its frequency increases with speed. The eigenvalue corresponding to $-p_2$ remains real and becomes more negative with speed; this is called the **castor mode**, because it roughly corresponds to the front assembly castoring about the steer axis. The eigenvalue corresponding to $-p_1$ also remains real but increases, eventually becoming slightly positive above some speed v_c ,



Motorcycle Dynamics and Control, Fig. 5 Variation of eigenvalues of Whipple model with speed v

resulting in an unstable system; the corresponding mode is called the **capsize mode**. Thus, the bike is stable in the autostable speed range (v_w, v_c) and unstable outside this speed range. However, above v_c , the unstable capsize mode is easily stabilized by a rider and usually without conscious effort. This is because the time-constant of the unstable capsize mode is very small (Astrom et al. 2005).

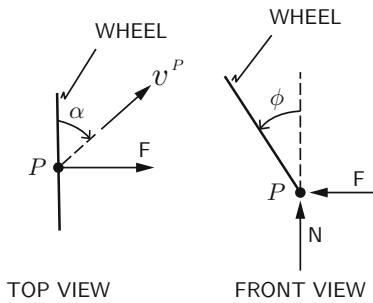
Sharp71 Model (Sharp 1971)

The Whipple bicycle model is not applicable at higher speeds. In particular, it does not contain a **wobble mode** which is common to bicycle and motorcycle behavior at higher speeds. A wobble mode is characterized mainly by oscillation of the front assembly about the steering axis and can sometimes be unstable. Also, in a real motorcycle, the damping and frequency of the weave mode do not continually increase with speed; the damping usually starts to decrease after a certain speed; sometimes this mode even becomes unstable. At higher speeds, one must depart from the simple non-slipping wheel model. In the Whipple model, the lateral force F on a wheel is simply that force which is necessary to maintain the non-holonomic constraint which requires the velocity of the wheel contact point to be parallel to the wheel plane, that is, no sideslip. Actual tires on wheels slip in the longitudinal and lateral direction and the lateral force depends on slip in the lateral direction, that is, sideslip. This lateral slip is defined by the **slip angle** α which is the angle between the contact point velocity and the intersection of the wheel plane and the ground; see Fig. 6.

The lateral force also depends on the tire **camber angle** which is the roll angle of the tire; motorcycle tires can achieve large camber angles in cornering; modern MotoGP racing motorcycles can achieve camber angles of nearly 65° . An initial linear model of a tire lateral force is given by

$$F = N(-k_\alpha \alpha + k_\phi \phi) \quad (14)$$

where N is the normal force on the tire, $k_\alpha > 0$ is called the tire **cornering stiffness**, and $k_\phi > 0$ is



Motorcycle Dynamics and Control, Fig. 6 Lateral force, slip angle, and camber angle

called the **camber stiffness**. Since lateral forces do not instantaneously respond to changes in slip angle and camber, the dynamic model,

$$\frac{\sigma}{v} \dot{F} + F = N(-k_{\alpha}\alpha + k_{\phi}\phi), \quad (15)$$

is usually used where $\sigma > 0$ is called the **relaxation length** of the tire. This yields more realistic behavior (Sharp 1971) and results in the appearance of the wobble mode. In this model the weave mode damping eventually decreases at higher speeds and the frequency does not continually increase. The frequency of the wobble mode is higher than that of the weave mode and its damping decreases at higher speeds.

Further Models

To obtain nonlinear models, resort is usually made to multi-body simulation codes. In recent years, several researchers have used such codes to make nonlinear models which take into account other features such as frame flexibility, rider models, and aerodynamics; see Cossalter (2002), Cossalter and Lot (2002), Sharp and Limebeer (2001), and Sharp et al. (2004). The nonlinear behavior of the tires is usually modeled with a version of the **Magic formula**; see Pacejka (2006), Sharp et al. (2004), and Cossalter et al. (2003). Another line of research is to use some of these models to obtain optimal trajectories for high performance; see Saccon et al. (2012).

Cross-References

► [Vehicle Dynamics Control](#)

Bibliography

- Astrom KJ, Klein RE, Lennartsson A (2005) Bicycle dynamics and control. *IEEE Control Syst Mag* 25(4):26–47
- Cossalter V (2002) *Motorcycle dynamics*. Race Dynamics, Greendale
- Cossalter V, Lot R (2002) A motorcycle multi-body model for real time simulations based on the natural coordinates approach. *Veh Syst Dyn* 37(6):423–447
- Cossalter V, Doria A, Lot R, Ruffo N, Salvador M (2003) Dynamic properties of motorcycle and scooter tires: measurement and comparison. *Veh Syst Dyn* 39(5):329–352
- Herlihy DV (2006) *Bicycle: the history*. Yale University Press, New Haven
- Limebeer DJN, Sharp RS (2006) Bicycles, motorcycles, and models. *IEEE Control Syst Mag* 26(5):34–61
- Meijaard JP, Papadopoulos JM, Ruina A, Schwab AL (2007) Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review – including appendix. *Proc R Soc A* 463(2084):1955–1982
- Pacejka HB (2006) *Tire and vehicle dynamics*, 2nd edn. SAE, Warrendale
- Saccon A, Hauser J, Beghi A (2012) Trajectory exploration of a rigid motorcycle model. *IEEE Trans Control Syst Technol* 20(2):424–437
- Sharp RS (1971) The stability and control of motorcycles. *J Mech Eng Sci* 13(5):316–329
- Sharp RS, Limebeer DJN (2001) A motorcycle model for stability and control analysis. *Multibody Syst Dyn* 6:123–142
- Sharp RS, Evangelou S, Limebeer DJN (2004) Advances in the modelling of motorcycle dynamics. *Multibody Syst Dyn* 12(3):251–283
- Whipple FJW (1899) The stability of the motion of a bicycle. *Q J Pure Appl Math* 30:312–348

Moving Horizon Estimation

James B. Rawlings and Douglas A. Allan
Department of Chemical Engineering, University of California, Santa Barbara, CA, USA

Abstract

Moving horizon estimation (MHE) is a state estimation method that is particularly useful for nonlinear or constrained dynamical systems for which few general methods with established properties are available. This article explains the concept of full information

estimation and introduces moving horizon estimation as a computable approximation of full information. To obtain a tractable approximation, the moving horizon method considers a fixed-sized window of only the most recent measurements, and therefore the horizon window *moves* in time with the data. The basic design methods for ensuring stability of MHE are presented. Relationships of full information and MHE to other state estimation methods such as Kalman filtering and statistical sampling are discussed.

Keywords

State estimation · Kalman filtering · Particle filtering

Introduction

In state estimation, we consider a dynamical system from which measurements are available. In discrete time, the system description is

$$x^+ = f(x, w) \quad y = h(x) + v \quad (1)$$

The state of the systems is $x \in \mathbb{R}^n$, the measurement is $y \in \mathbb{R}^p$, and the notation x^+ means x at the next sample time. A control input u may be included in the model, but it is considered a known variable, and its inclusion is irrelevant to state estimation, so we suppress it in the model under consideration here. We receive measurement y from the sensor, but the process disturbance, $w \in \mathbb{R}^g$; measurement disturbance, $v \in \mathbb{R}^p$; and system initial state, $x(0)$, are considered unknown variables.

The goal of state estimation is to construct or estimate the trajectory of x from only the measurements y . Note that for control purposes, we are usually interested in the estimate of the state at the current time, T , rather than the entire trajectory over the time interval $[0, T]$. In the moving horizon estimation (MHE) method, we use optimization to achieve this goal. We have two sources of error, the state transition is affected by an unknown process disturbance (or noise),

w , and the measurement process is affected by another disturbance, v . In the MHE approach, we formulate the optimization objective to minimize the size of these errors, thus finding a trajectory of the state that comes close to satisfying the (error-free) model while still fitting the measurements.

First, we define some notation necessary to distinguish the system variables from the estimator variables. We have already introduced the system variables (x, w, y, v) . In the estimator optimization problem, these have corresponding decision variables, which we denote by the Greek letters $(\chi, \omega, \eta, \nu)$. The relationships between these variables are

$$\chi^+ = f(\chi, \omega) \quad y = h(\chi) + \nu \quad (2)$$

and they are depicted in Fig. 1. Notice that ν measures the gap between the model prediction $\eta = h(\chi)$ and the measurement y . The *optimal* decision variables are denoted $(\hat{x}, \hat{w}, \hat{y}, \hat{v})$, and these optimal decisions are the estimates provided by the state estimator.

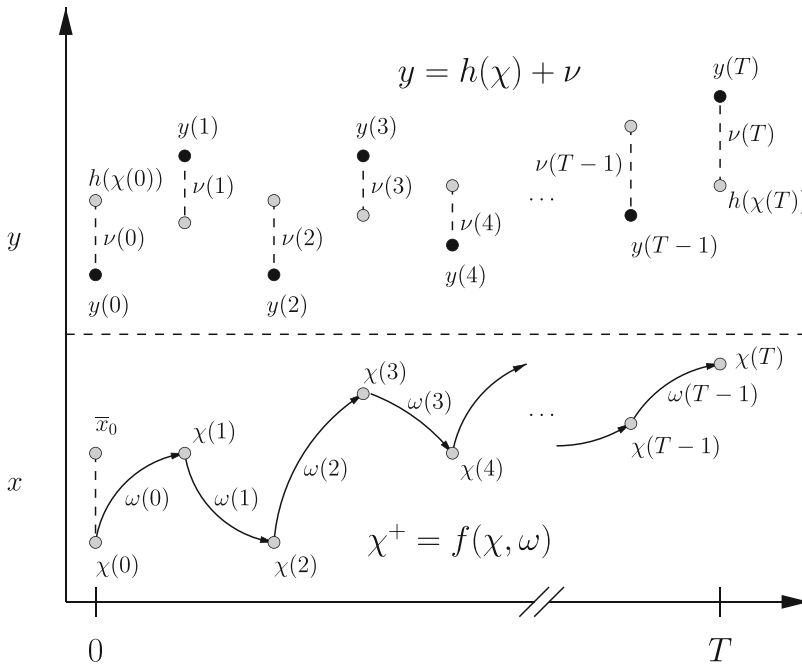
Full Information Estimation

The objective function for full information estimation (FIE) is

$$V_T(\chi(0), \omega) = \ell_x(\chi(0) - \bar{x}_0) + \sum_{i=0}^{T-1} \ell_d(\omega(i), \nu(i)) \quad (3)$$

subject to (2), in which T is the current time, ω is the estimated sequence of process disturbances, $(\omega(0), \dots, \omega(T-1))$, $y(i)$ is the measurement at time i , and $\bar{x}_0$ is the prior, i.e., available, value of the initial state. *Full information* here means that we use *all* the data on time interval $[0, T]$ to estimate the state (or state trajectory) at time T . The stage cost $\ell_d(\omega, \nu)$ costs the model disturbance and the fitting error, the two error sources that we reconcile in all state estimation problems.

The full information estimator is then defined as the solution to



Moving Horizon Estimation, Fig. 1 The state, measured output, and disturbance variables appearing in the state estimation optimization problem. The state trajectory

(gray circles in lower half) is to be reconstructed given the measurements (black circles in upper half)

$$\min_{\chi(0), \omega} V_T(\chi(0), \omega) \quad (4)$$

The solution to the optimization exists for all $T \in \mathbb{I}_{\geq 0}$ under mild continuity assumptions and choice of stage cost. Many choices of (positive, continuous) prior cost $\ell_x(\cdot)$ and stage cost $\ell_d(\cdot)$ are possible, providing a rich class of estimation problems that can be tailored to different applications. Because the system model (1) and cost function (3) are so general, it is perhaps best to start off by specializing them to see the connection to some classical results.

Related Problem: The Kalman Filter

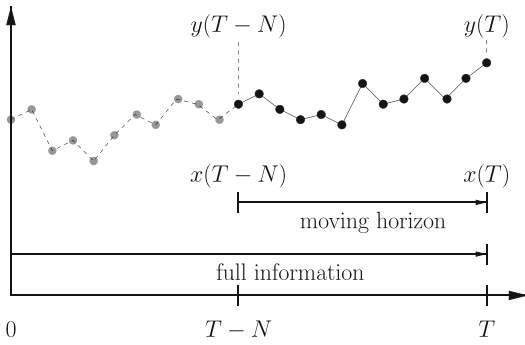
If we specialize to the linear dynamic model $f(x, w) = Ax + Gw$ and $h(x) = Cx$ and let $x(0)$, w , and v be independent, normally distributed random variables, the classic Kalman filter is known to be the statistically optimal estimator, i.e., the Kalman filter produces the state estimate that maximizes the conditional probability of

$x(T)$ given $y(0), \dots, y(T)$. The full information estimator is *equivalent* to the Kalman filter given the linear model assumption and the following choice of quadratic prior and stage costs

$$\begin{aligned} \ell_x(\chi(0), \bar{x}_0) &= (1/2) \|\chi(0) - \bar{x}_0\|_{P_0^{-1}}^2 \\ \ell_d(\omega, v) &= (1/2) \left(\|\omega\|_{Q^{-1}}^2 + \|v\|_{R^{-1}}^2 \right) \end{aligned}$$

where random variable $x(0)$ is assumed to have mean $\bar{x}_0$ and variance P_0 and random variables w and v are assumed zero mean with variances Q and R , respectively. The Kalman filter is also a *recursive* solution to the state estimation problem so that only the current mean $\hat{x}$ and variance P of the conditional density are required to be stored, instead of the entire history of measurements $y(i), i = 0, \dots, T$. This computational efficiency is critical for success in online application for processes with short time scales requiring fast processing.

But if we consider nonlinear models, the maximization of conditional density of $x(T)$ is usually



Moving Horizon Estimation, Fig. 2 Schematic of the moving horizon estimation problem

an intractable problem, especially in online applications. So MHE becomes a natural alternative for nonlinear models or if an application calls for hard constraints to be imposed on the estimated variables.

Moving the Horizon

An obvious problem with solving the full information optimization problem is that the number of decision variables grows linearly with time T , which quickly renders the problem intractable for continuous processes that have no final time. A natural alternative to full information is to consider instead a finite moving horizon of the most recent N measurements. Figure 2 displays this idea. The initial condition $\chi(0)$ is now replaced by the initial state in the horizon, $\chi(T-N)$, and the decision variable sequence of process disturbances is now just the last N variables $\omega = (\omega(T-N), \dots, \omega(T-1))$. Now, the big question remaining is what to do about the neglected, past data. This question is strongly related to what penalty to use on the initial state in the horizon $\chi(T-N)$. If we make this initial state a free variable, that is equivalent to completely discounting the past data. If we wish to retain some of the influence of the past data and keep the moving horizon estimation problem close to the full information problem, then we must choose an appropriate penalty for the initial state. We discuss this problem next.

Arrival cost. When time is less than or equal to the horizon length, $T \leq N$, we can simply do full information estimation. So we assume throughout that $T > N$. For $T > N$, we express the MHE objective function as

$$\hat{V}_T(\chi(T-N), \omega) = \Gamma_{T-N}(\chi(T-N)) + \sum_{i=T-N}^{T-1} \ell_d(\omega(i), v(i))$$

subject to (2). The MHE problem is defined to be

$$\min_{\chi(T-N), \omega} \hat{V}_T(\chi(T-N), \omega) \quad (5)$$

in which $\omega = \{\omega(T-N), \dots, \omega(T-1)\}$ and the hat on $V(\cdot)$ distinguishes the MHE objective function from full information. The designer must now choose this prior weighting $\Gamma_k(\cdot)$ for $k > N$.

To think about how to choose this prior weighting, it is helpful to first think about solving the full information problem by breaking it into *two* nonoverlapping sequences of decision variables: the decision variables in the time interval corresponding to the neglected data ($\omega(0), \omega(1), \dots, \omega(T-N-1)$) and those in the time interval corresponding to the considered data in the horizon ($\omega(T-N), \dots, \omega(T-1)$). If we optimize over the first sequence of variables and store the solution as a function of the terminal state $\chi(T-N)$, we have defined what is known as the arrival cost. This is the optimal cost to arrive at a given state value.

Definition 1 (Arrival cost) The (full information) arrival cost is defined for $k \geq 1$ as

$$Z_k(x) = \min_{\chi(0), \omega} V_k(\chi(0), \omega) \quad (6)$$

subject to (2) and $\chi(k; \chi(0), \omega) = x$.

Notice the terminal constraint that χ at time k ends at value x . Given this arrival cost function, we can then solve the full information problem by optimizing over the remaining decision variables. What we have described is simply the *dynamic*

programming strategy for optimizing over a sum of stage costs with a dynamic model (Bertsekas 1995).

We have the following important equivalence.

Lemma 1 (MHE and full information estimation) *The MHE problem (5) is equivalent to the full information problem (4) for the choice $\Gamma_k(\cdot) = Z_k(\cdot)$ for all $k > N$ and $N \geq 1$.*

Using dynamic programming to decompose the full information problem into an MHE problem with an arrival cost penalty is conceptually important to understand the structure of the problem, but it doesn't yet provide us with an implementable estimation strategy because we cannot compute and store the arrival cost when the model is nonlinear or other constraints are present in the problem. But if we are not too worried about the optimality of the estimator and are mainly interested in other properties, such as stability of the estimator, we can find simpler design methods for choosing the weighting $\Gamma_k(\cdot)$. We address this issue next.

Estimator Properties: Stability

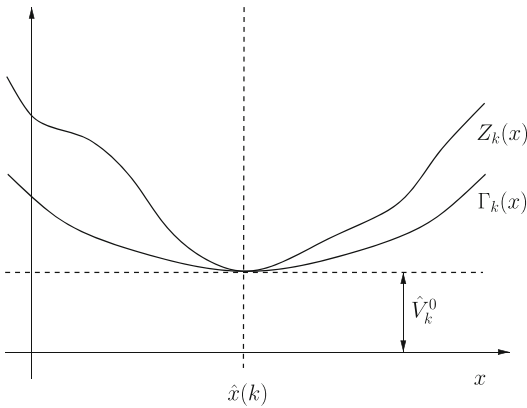
An estimator is termed *robustly stable* if small prior errors $x(0) - \bar{x}_0$ and small disturbances (w, v) lead to small estimate errors proportional to the size of (w, v) . Even in the case of the Kalman filter for linear systems, estimator optimality is insufficient to guarantee stability. For linear systems, there exists a stable estimator only if the system is *detectable*. The Kalman filter is one such estimator, but statistical optimality is not required for (robust) stability, and there are many Luenberger observers that are robustly stable. For nonlinear systems, one popular form of detectability is called *incremental input/output-to-state stability (i-IOSS)* (Sontag and Wang 1997). A property stronger than detectability is *observability*. A system is observable if its initial state can be reconstructed from a finite number of measurements. Precise definitions of these terms are available elsewhere (Rawlings et al. 2017, Ch. 4), but these basic notions are sufficient for this overview.

Since there are suboptimal estimators that nevertheless are robustly stable for detectable linear systems, we can hope that there are also such estimators for nonlinear systems. Because even the more modest goal of making MHE equivalent to FIE appears out of reach, we examine the circumstances in which MHE is robustly stable. The simplest case is when the system is observable. Leaving aside certain technicalities, MHE using the flat prior $\Gamma_k(\cdot) = 0$ is robustly stable if a system is observable. However, we may significantly compromise estimation accuracy if we ignore past data completely.

Here, we are faced with a situation embarrassing to the current state of theory: there is no general proof that FIE (and, as a result, MHE with the arrival cost as prior penalty) is robustly stable in the presence of bounded disturbances, i.e., persistent small disturbances. It is known that FIE is both robustly stable and convergent when the disturbances converge to zero quickly enough (Rawlings and Ji 2012) and in the special case of exponential detectability and a restrictive choice of stage cost (excluding least-squares objectives) in the presence of bounded disturbances (Knüfer and Müller 2018). Even more perplexing is that there are conditions that guarantee the robust stability of MHE in the case of exponential detectability while (potentially) permitting a least-squares objective (Müller 2017; Allan and Rawlings 2019). Unfortunately, these conditions are not well understood and difficult to verify, and thus do not yet shed much light on practical design considerations. In the case of convergent disturbances, we can achieve robust stability of MHE by choosing $\Gamma_k(\cdot)$ both positive-definite and *smaller* than the arrival cost, as shown in Fig. 3 (Rawlings et al. 2017, Theorem 4.25).

Related Problem: Statistical Sampling

MHE is based on optimizing an objective function that bears some relationship to the conditional probability of the state trajectory given the measurements. As discussed in the section on the Kalman filter, if the system is linear with



Moving Horizon Estimation, Fig. 3 Arrival cost $Z_k(x)$, underbounding prior weighting $\Gamma_k(x)$, and MHE optimal value $\hat{V}_k^0$, for all x and $k > N$, $Z_k(x) \geq \Gamma_k(x) \geq \hat{V}_k^0$, and $Z_k(\hat{x}(k)) = \Gamma_k(\hat{x}(k)) = \hat{V}_k^0$

normally distributed noise, this relationship can be made exact, and MHE is therefore an optimal statistical estimator. But in the nonlinear case, the objective function is chosen with engineering judgment and is usually only a surrogate for the conditional probability. By contrast, sampling methods such as particle filtering are designed to sample the conditional density also in the nonlinear case. The mean and variance of the samples then provide estimates of the mean and variance of the conditional density of interest. In the limit of infinitely many samples, these methods are exact. The efficiency of the sampling methods depends strongly on the model and the dimension of the state vector n , however. The efficiency of the sampling strategy is particularly important for online use of state estimators. Rawlings and Bakshi (2006) and Rawlings and Mayne (2009, pp. 329–355) provide some comparisons of particle filtering with MHE.

Another downside of the particle filtering methods covered in those sources is that, as time goes on, particles can drift out of areas with high probability density. These low-weight particles contribute little to the state estimate and variance, leading to a small number of particles with large weights to determine these quantities. Resampling can mitigate this problem, but it does not solve it. A better method is the

feedback particle filter proposed in Yang et al. (2014). There, a control law is used to move particles as part of the Bayesian update step of the filter, which helps keep them in regions of high probability density.

Numerical Implementation

MHE is implemented by solving a series of nonlinear programs. In principle, any nonlinear programming method can be used, but the problem has a sparse structure that can be exploited by suitable solvers. Several open-source packages that exist, together, permit users to quickly implement MHE without detailed knowledge of the underlying numerical methods. IPOPT (Wächter and Biegler 2006) is an interior-point solver, CasADi (Available online at casadi.org) is a symbolic framework that provides algorithmic (automatic) differentiation and interfaces to optimization packages, and MPCTools (Available online at www.bitbucket.org/rawlings-group/octave-mpctools) provides a high-level interface to CasADi and IPOPT. Several examples of MHE problems implemented with these tools are available online (www.engineering.ucsb.edu/~jbrow/mpc/figures.html).

Summary and Future Directions

MHE is one of the few state estimation methods that can be applied to general nonlinear models for which properties such as estimator stability can be established (Rawlings et al. 2017; Rao et al. 2003). The required online solution of an optimization problem is computationally demanding in some applications but can provide significant benefits in estimator accuracy and rate of convergence (Patwardhan et al. 2012). Current topics for MHE theoretical research include proving robust stability of FIE in the presence of bounded disturbances, seeking better understanding about the conditions under which MHE is robustly stable, and establishing properties of suboptimal MHE. For example, Alessandri et al. (2008) analyze a simplified form of MHE where

the ω decision variables of (4) are neglected and show that terminating within a specified tolerance of optimality preserves the boundedness and stability of the estimate error. Although encouraging that strict optimality is not required, the challenge with using this style of suboptimal MHE is that because the tolerance is given from the *global* solution, a global solver must still be used online to satisfy this requirement. The current main focus for MHE applied research involves reducing the online computational complexity to reliably handle challenging large-dimensional, nonlinear applications (Zavala et al. 2008; Zavala and Biegler 2009; Kühl et al. 2011; López-Negrete and Biegler 2012; Wynn et al. 2014).

Cross-References

- [Bounds on Estimation](#)
- [Extended Kalman Filters](#)
- [Nonlinear Filters](#)
- [Particle Filters](#)

Recommended Reading

Moving horizon estimation has by this point a fairly extensive literature; a recent summary is provided in Rawlings et al. (2017, pp. 325–326). The following references provide either (i) general background required to understand MHE theory and its relationship to other methods or (ii) computational methods for solving the real-time MHE optimization problem or (iii) challenging nonlinear applications that demonstrate benefits and probe the current limits of MHE implementations.

Bibliography

- Alessandri A, Baglietto M, Battistelli G (2008) Moving-horizon state estimation for nonlinear discrete-time systems: new stability results and approximation schemes. *Automatica* 44(7):1753–1765
- Allan DA, Rawlings JB (2019) Moving horizon estimation. In: Raković S, Levine WS (eds) *Handbook of model predictive control*. Springer, Cham, pp 99–124
- Bertsekas DP (1995) *Dynamic programming and optimal control*, vol 1. Athena Scientific, Belmont
- Knüfer S, Müller MA (2018) Robust global exponential stability for moving horizon estimation. In: 2018 IEEE conference on decision and control (CDC), pp 3477–3482
- Kühl P, Diehl M, Kraus T, Schlöder JP, Bock HG (2011) A real-time algorithm for moving horizon state and parameter estimation. *Comput Chem Eng* 35: 71–83
- López-Negrete R, Biegler LT (2012) A moving horizon estimator for processes with multi-rate measurements: a nonlinear programming sensitivity approach. *J Process Control* 22(4):677–688
- Müller MA (2017) Nonlinear moving horizon estimation in the presence of bounded disturbances. *Automatica* 79:306–314
- Patwardhan SC, Narasimhan S, Jagadeesan P, Gopaluni B, Shah SL (2012) Nonlinear Bayesian state estimation: a review of recent developments. *Control Eng Pract* 20: 933–953
- Rao CV, Rawlings JB, Mayne DQ (2003) Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations. *IEEE Trans Autom Control* 48(2):246–258
- Rawlings JB, Bakshi BR (2006) Particle filtering and moving horizon estimation. *Comput Chem Eng* 30: 1529–1541
- Rawlings JB, Ji L (2012) Optimization-based state estimation: current status and some new results. *J Process Control* 22:1439–1444
- Rawlings JB, Mayne DQ (2009) *Model predictive control: theory and design*. Nob Hill Publishing, Madison, 668pp, ISBN 978-0-9759377-0-9
- Rawlings JB, Mayne DQ, Diehl MM (2017) *Model predictive control: theory, design, and computation*, 2nd edn. Nob Hill Publishing, Madison, 770pp, ISBN 978-0-9759377-3-0
- Sontag ED, Wang Y (1997) Output-to-state stability and detectability of nonlinear systems. *Syst Control Lett* 29:279–290
- Wächter A, Biegler LT (2006) On the implementation of a primal-dual interior-point filter line-search algorithm for large-scale nonlinear programming. *Math Prog* 106(1):25–57
- Wynn A, Vukov M, Diehl M (2014) Convergence guarantees for moving horizon estimation based on the real-time iteration scheme. *IEEE Trans Autom Control* 59(8):2215–2221
- Yang T, Blom HAP, Mehta PG (2014) The continuous-discrete time feedback particle filter. In: 2014 American control conference, pp 648–653
- Zavala VM, Biegler LT (2009) Optimization-based strategies for the operation of low-density polyethylene tubular reactors: moving horizon estimation. *Comput Chem Eng* 33(1):379–390
- Zavala VM, Laird CD, Biegler LT (2008) A fast moving horizon estimation algorithm based on nonlinear programming sensitivity. *J Process Control* 18: 876–884

MRAC

► [Model Reference Adaptive Control](#)

Keywords

Game theory · Non-stationarity ·
Equilibrium · Communication ·
Coordination · Competition

MSPCA

► [Multiscale Multivariate Statistical Process Control](#)

Multiagent Reinforcement Learning

Jonathan P. How, Dong-Ki Kim, and
Samir Wadhwania

Department of Aeronautics and Astronautics,
Aerospace Controls Laboratory, Massachusetts
Institute of Technology, Cambridge, MA, USA

Abstract

The area of multiagent reinforcement learning (MARL) provides a promising approach to learning collaborative policies for multiagent systems. However, MARL is inherently more difficult than single-agent learning problems because agents interact with both the environment and other agents. Specifically, learning in multiagent settings involves significant issues of the nonstationary, equilibrium selection, credit assignment, and curse of dimensionality. Despite these difficulties, there have been important recent developments in MARL. This entry provides a background in multiagent reinforcement learning as well as an overview of recent work on topics of game theory, communication, coordination, knowledge sharing, and agent modeling. This entry also summarizes several useful multiagent simulation platforms.

Introduction

The area of multiagent reinforcement learning (MARL) investigates how a group of artificial intelligence (AI) agents can learn to cooperate and coordinate at the level of human experts. MARL presents several benefits over a single-agent learning approach, including (1) improved robustness because the failure of a single agent could be addressed by another agent; (2) improved efficiency, since a complicated task can be divided into simpler sub-tasks, distributed across agents, and then solved simultaneously; and (3) improved scalability, because new agents can easily be added to the decentralized collaboration in MARL. These advantages increase the appeal of MARL in numerous applications, including robot manufacturing, traffic control, and network routing.

Learning in a multiagent setting is inherently more difficult than single-agent learning problems, in particular because agents interact with both the environment and the other agents (Buşoniu et al. 2010). One crucial challenge is the difficulty of learning optimal policies in the presence of changing policies of the other learning agents whose concurrent actions affect the team-wide performance (Omidshafiei et al. 2017; Tuyls and Weiss 2012; Hernandez-Leal et al. 2017). Therefore, a small change in one agent's policy may result in large, unpredictable changes in the policies of fellow agents. The difficulty of learning in MARL is further complicated by the challenges of multiagent credit assignments (han Chang et al. 2004; Foerster et al. 2017), the high dimensionality of the problems (Yang et al. 2018), and the lack of convergence guarantees (Bowling 2005; Nowe et al. 2012). There have been several important recent developments in MARL, from integration of game-theoretic concepts to new deep learning methods.

Background

Framework

There are several frameworks introduced in this section which can be used to represent learning in MARL settings.

Normal Form Games

Interactions between multiple agents can be represented by a normal form game. Formally, a normal form game is defined as a tuple $\langle \mathcal{I}, \mathcal{A}, \mathcal{R} \rangle$ (Leyton-Brown and Shoham 2008); $\mathcal{I} = \{1, \dots, n\}$ is the set of n agents (also referred to as players); $\mathcal{A} = \{\mathcal{A}^1, \dots, \mathcal{A}^n\}$ is the set of available actions for each agent; and $\mathcal{R} = \{\mathcal{R}^1, \dots, \mathcal{R}^n\}$ is the set of reward functions, where $i \in \mathcal{I}$ and $\mathcal{R}^i : \mathcal{A}^1 \times \dots \times \mathcal{A}^n \rightarrow \mathbb{R}$ is a reward function for agent i according to the payoff table (e.g., Table 1). Joint sets are typeset in bold throughout the paper for clarity.

In a normal form game, each agent i selects an action a^i from its strategy $\sigma^i : \mathcal{A}^i \rightarrow [0, 1]$, where σ^i outputs the probability of taking any given action in the action set. Then, agent i receives its utility based on a joint action $\mathbf{a} = \{a^1, \dots, a^n\}$ and its reward function (i.e., $r^i = \mathcal{R}^i(\mathbf{a})$). For instance, with the payoff table shown in Table 1, if agent 1's strategy chooses action x (i.e., $a^1 = x$) and agent 2's strategy chooses action y (i.e., $a^2 = y$), then the joint action is $\mathbf{a} = \{x, y\}$, and agents 1 and 2 receive a reward of 0 and 10, respectively. Each agent i has an objective to maximize its *expected* utility: $\sum_{\mathbf{a} \in \mathbb{A}} \prod_{j=1}^n \sigma^j(a^j) R^i(\mathbf{a})$. Note that the utility depends on the other agents' actions, and thus the expected utility is affected by other agents' strategies. Therefore, agents must consider the strategies of others in order to maximize their expected utilities.

Markov Games

The normal form game framework assumes the environment is static, so it is not suitable for problems in which agents are making sequential decisions in an environment where there is a state transition function. A Markov game (Littman 1994) provides a more general framework than a normal form game based on the Markov

Multiagent Reinforcement Learning, Table 1

Example payoff table of the prisoner's dilemma. Each agent chooses to either cooperate and stay silent (x) or betray and confess their crime (y)

		Agent 2	
Agent 1		x	y
	x	(5, 5)	(0, 10)
	y	(10, 0)	(1, 1)

decision process (MDP). Formally, a Markov game is defined as a tuple $\langle \mathcal{I}, \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma \rangle$; $\mathcal{I} = \{1, \dots, n\}$ is the set of n agents; $\mathcal{S}$ is the set of states; $\mathcal{A} = \{\mathcal{A}^1, \dots, \mathcal{A}^n\}$ is the set of action spaces for all n agents; $\mathcal{R} = \{\mathcal{R}^1, \dots, \mathcal{R}^n\}$ is the set of reward functions for all n agents; $\mathcal{T} : \mathcal{S} \times \mathcal{A}^1 \times \dots \times \mathcal{A}^n \mapsto \mathcal{S}$ is the transition function; and $\gamma \in [0, 1]$ is the discount factor. At each timestep t , each agent i executes an action according to its stochastic policy $a_t^i \sim \pi^i(s_t; \theta^i)$ parameterized by θ^i , where $s_t \in \mathcal{S}$. A joint action $\mathbf{a}_t = \{a_t^1, \dots, a_t^n\}$ yields a transition from current state s_t to next state $s_{t+1} \in \mathcal{S}$ with probability $\mathcal{T}(s_{t+1} | s_t)$. Agent i then obtains a reward according to the function $r_t^i = \mathcal{R}^i(s_t, \mathbf{a}_t, s_{t+1})$. Each agent has an objective to maximize its expected cumulative reward or value function. Specifically, given an initial state s_0 and current joint policy $\{\pi^1, \dots, \pi^n\}$, a value function at the initial state for agent i is given by $V^i(s_0) = \mathbb{E}[\sum_t \gamma^t r_t^i | s_0, \pi^1, \dots, \pi^n]$. As in normal form games, the value function depends on other agents' current policies. Because agents are independently updating their policies in MARL, the dependency in the value function leads to the challenge of non-stationarity (see Section “Challenges”). Note that a Markov game reduces to an MDP (or a normal form game) if there is only a single agent (or a single state) (Nowe et al. 2012).

Decentralized Partially Observable MDP (Dec-POMDP)

Dec-POMDPs Oliehoek and Amato (2016) are a special case of Markov games in which agents have equivalent reward functions (i.e., shared team reward), but do not have direct access to the state. At each time step t , each agent can

partially observe the state by receiving observation $o_t^i \in \Omega^i$ with joint observation probability $\mathcal{O}(\mathbf{o}_t, s_{t+1}, \mathbf{a}_t) = P(\mathbf{o}_t | s_{t+1}, \mathbf{a}_t)$, where Ω^i is a finite set of observations available to agent i , $\mathbf{o}_t = \{o_t^1, \dots, o_t^n\}$, and $\mathcal{O}$ is the observation function representing the probability of observing $\mathbf{o}_t$ conditioned on agents executing $\mathbf{a}_t$ and the state transitioning to s_{t+1} . In this case, agent i selects an action based on its observation history $h_t^i: a_t^i \sim \pi^i(h_t^i; \theta^i)$, where $h_t^i = \{o_0^i, \dots, o_t^i\}$.

Meta Games

A meta game (also referred to as an empirical game) (Tuyls et al. 2018; Ponsen et al. 2009) is a simplified model of a multiagent system. In a meta game, only relevant meta-strategies, or styles of play, that are played empirically are considered, instead of the entire primitive strategies. For example, in a game of poker, possible meta-strategies are {"aggressive," "tight," "passive"}.

Challenges

Non-stationarity

In MARL, multiple agents concurrently learn and interact in a shared environment (Buşoniu et al. 2010). From the perspective of agent i , the value of a state s estimated at a learning time t_1 may not match the value of the same state estimated at a different learning time t_2 (i.e., $V_{t_1}^i(s) \neq V_{t_2}^i(s)$) due to the possibility that other agents could have different policies as a result of learning (i.e., $\{\pi_{t_1}^1, \dots, \pi_{t_1}^n\} \neq \{\pi_{t_2}^1, \dots, \pi_{t_2}^n\}$). This leads to the environment being perceived as nonstationary by individual agents, rendering the Markov property invalid (Tesauro 2004). This problem is also referred to as the *moving-target* problem (Tuyls and Weiss 2012).

Equilibrium Selection

The concept of an optimal solution may not exist in some multiagent settings. An alternative solution concept is an equilibrium among agent strategies, such as a Nash equilibrium (Leyton-Brown and Shoham 2008; Nowe et al. 2012). Furthermore, multiple equilibria can exist in MARL (Nowe et al. 2012) (e.g., two Nash equilibria exist in Table 1: $\{x, x\}$ and $\{y, y\}$).

The challenge then is to converge to a better equilibrium with higher expected utility or return. This equilibrium selection, however, is often intractable (e.g., when the agents' action spaces are large or continuous) (Omidshafiei et al. 2019; Goldberg et al. 2010; Avis et al. 2010).

Credit Assignment

In cooperative MARL settings, joint actions often generate shared rewards, such that $\mathcal{R}^1(s_t, \mathbf{a}_t, s_{t+1}) = \dots = \mathcal{R}^n(s_t, \mathbf{a}_t, s_{t+1})$ for all agents. As they all receive the same reward, it is difficult for each agent to identify its own contribution to the team's performance. This makes coordination difficult, even with small numbers of agents (han Chang et al. 2004; Foerster et al. 2017).

Curse of Dimensionality

As the number of agents increases, the most straightforward issue with learning is the exponentially increasing state/action space. The credit assignment problem also becomes increasingly more difficult with more agents due to greater noise from the increased number of agent interactions and exploratory actions being taken at any given time (Yang et al. 2018). To illustrate this idea, consider four agents learning in a discrete, 4×4 gridworld. While this domain might seem small, it has the equivalent state space complexity to a 256×256 gridworld for a single-agent problem (a 6×6 multiagent problem has the equivalent scaling to a 1296×1296 single-agent problem).

Current Research Topics in MARL

Game Theory

One important solution concept in game theory is the Nash equilibrium in which each agent acts with its best response to the current strategies of the other agents (Leyton-Brown and Shoham 2008). A number of approaches use the Nash equilibrium as a solution concept for MARL. For example, *policy-space response oracles* (PSRO) (Lanctot et al. 2017) incrementally build a policy by finding the best response to

the meta-strategies of the other players. An extension, called *deep cognitive hierarchies* (DCH), approximates PSRO by searching for a fixed number of epochs (in contrast to the unbounded number of epochs in PSRO). PSRO/DCH are evaluated on 2D gridworld games and Leduc poker, (Leduc poker is a toy poker game often used as a benchmark in Poker AI research. See Southey et al. (2012) for more information.) which show convergence to an approximate Nash equilibrium.

Dynamical multiagent systems, however, are not guaranteed to converge to a Nash equilibrium (Omidshafiei et al. 2019). To address this issue, *Markov-Conley Chains* (MCCs) have been introduced as a new game-theoretic solution concept that is capable of capturing the evolutionary dynamics of multiagent interactions. The α -Rank method utilizes MCCs to evaluate and rank agents in large-scale dynamical systems and has been empirically validated in several domains, such as AlphaZero Chess and Leduc Poker (Omidshafiei et al. 2019).

Learning Communication

Communication between agents has an essential role in accomplishing multiagent tasks, but it is often difficult to specify an appropriate communication protocol. Recent work has shown that agents can learn and use a communication protocol to solve a coordinated task. For instance, *reinforced inter-agent learning* (RIAL) (Foerster et al. 2016) learns a discrete and finite set of communication messages. Each agent outputs Q-values for the environment and communication actions, and they then use their observations and received messages to make their decisions. Because agents in RIAL are assumed to be homogeneous, learning is simplified via parameter sharing in which all agents simultaneously train a single policy. Despite learning a single policy, agents can still learn specialized behaviors that are based on different initial conditions, observations, or relative formations (Panait and Luke 2005). *Differentiable inter-agent learning* (DIAL) (Foerster et al. 2016) introduces the idea of agents providing feedback about received messages by sending gradients across

the communication channel. Experiments on coordinated tasks show that the integration of this feedback in DIAL leads to a better performance than both RIAL and a no-communication baseline. An alternative, called *communication neural nets* (CommNets) (Sukhbaatar et al. 2016), sends continuous messages between agents. Each agent takes the form of either multilayer neural networks or long short-term memory networks (LSTMs) (Hochreiter and Schmidhuber 1997), (long short-term memory (LSTM) networks (Hochreiter and Schmidhuber 1997) are one type of recurrent neural network (RNN). LSTMs are well-suited for identifying long-term temporal dependencies by augmenting information in the cell state.) and the message is constructed by taking the average of the outputs among internal states in neural networks.

Learning Coordination

The learning of cooperative behaviors can be achieved without the explicit involvement of communication. For example, *multiagent deep deterministic policy gradient* (MADDPG) (Lowe et al. 2017) proposes actor-critic policy gradient methods with decentralized actors and centralized critics. During training, centralized critics receive additional information about the other agents (i.e., agents' policies, observations, and actions) and compute *joint* Q-values. As a result, the environment is stationary from the perspective of the centralized critics, which has been shown to significantly improve the stability of the multiagent learning. The critic is only necessary during the learning process, and since the actors only use local information during both training and execution, the system is considered to be decentralized during execution. MADDPG has demonstrated success in both collaborative and competitive tasks, and it has been evaluated in particle environments at various levels of complexity (see Section “Domains of MARL”).

Another related research direction is based on population-based training (PBT). PBT Jaderberg et al. (2017) is an online evolutionary process that optimizes hyperparameters by replacing underperforming agents with well-performing agents and adding noise (called mutation)

between the optimization steps. PBT is extended to multiagent settings in the *for the win* (FTW) framework (Jaderberg et al. 2018) in which a population of 30 agents is used to learn collaborative behaviors. FTW agents use a two-tiered hierarchical structure of recurrent neural networks (RNNs) (RNNs are a type of network that incorporates feedback, allowing the network to process sequences of data.) that operate on different time scales, improving their ability to utilize memory and learn temporally consistent strategies. FTW recently demonstrated “above human-level performance” in a 2 versus 2 capture the flag challenge in the video game, Quake III Arena (id Software).

Knowledge Sharing

Agents typically develop individualized knowledge of a domain as a result of executing some type of stochastic exploration of the environment. Similar to human social groups, the collective intelligence of MARL agents may be improved if the agents share what they have individually learned. Several frameworks allow a wide variety of methods to transfer different types of knowledge (Taylor and Stone 2009; Wadhwanian et al. 2019). One approach to transferring knowledge is via action advising in which an experienced “teacher” agent helps a less experienced “student” agent by suggesting which action to take next. For example, the recent *Learning to Coordinate and Teach Reinforcement* (LeCTR) framework (Omidshafiei et al. 2018) learns teacher policies that indicate both when and which actions to advise. LeCTR shows improved team-wide learning performance compared to prior teaching methods that advise using a variety of teaching heuristics (Amir et al. 2016; Clouse 1997; da Silva et al. 2017; Torrey and Taylor 2013). *Hierarchical multiagent teaching* (HMAT) (Kim et al. 2019) extends LeCTR to learn teacher policies with deep hierarchical reinforcement learning (RL). HMAT shows improved learning performance in difficult tasks involving high-dimension state-action spaces for problems with long time horizons.

The *distillation with value matching* (DVM) (Wadhwanian et al. 2019) approach accomplished

knowledge sharing between homogeneous agents by exploiting the fact that teams of homogeneous agents have the inherent property of being interchangeable within the same environment (Zinkevich and Balch 2001). In particular, DVM creates a new distilled policy that combines knowledge learned from all agents by minimizing the Kullback-Leibler divergence of the policy with respect to all agents’ policies. Additionally, the centralized critics in DVM are trained such that they output an identical Q-value invariant to the ordering of inputs (a process referred to as value matching). Value matching is shown to be a necessary step to reflect the new Q-value with the distilled policy and enables continued learning without forgetting of the information that has been distilled.

Agent Modeling

It is typically necessary for an agent in a multiagent system to be able to predict the intent of the other agents and then respond accordingly. However, if, in contrast to the centralized training approaches, agents cannot share full policies with one another (e.g., due to privacy concerns), agents can instead develop models of the other agents to predict their strategies. *Deep reinforcement opponent network* (DRON) (He et al. 2016) learns representations of other agents’ policies and incorporates them into a joint Q-function. A second approach by Grover et al. (Grover et al. 2018) aims to learn representations of agent policies using an unsupervised encoder-decoder framework. An encoder is used to learn a mapping from episodes (i.e., a sequence of observation and action pairs) to continuous embeddings, which are then mapped to a distribution over actions of the agent being modeled using a learned decoder. Experiments show that the learned embeddings are generalizable and can even be used to represent unseen agents. Inferring other agents’ policies is often useful, but for some domains/tasks, other information might also need to be inferred. For example, *Theory of Mind neural network* (ToMnet) (Rabinowitz et al. 2018) learns to predict an agent’s characteristics and internal states, such as action probabilities,

probabilities of certain objects being used, and successor representations (Dayan 1993).

Domains of MARL

Evaluating new algorithms is an essential part of research on MARL. This section highlights numerous multiagent evaluation platforms and provides links to code if they are open source.

Multiagent Particle Environment

This environment provides a simple multiagent environment with basic simulated physics, such as collision, speed, and acceleration. The observation space is continuous, and the action space can be either discrete or continuous. Several scenarios are provided within the environment. These scenarios include cooperative navigation, where agents learn to cover all landmarks; predator–prey, where multiple predators learn to collaborate to catch prey; and cooperative communication, where a speaker agent communicates to guide a listener agent. MADDPG (Lowe et al. 2017) uses the environment to show both collaborative and competitive behaviors. Researchers can create their own environments, and open-sourced code is available at <https://github.com/openai/multiagent-particle-envs>.

MAgent

MAgent provides a large-scale platform that can support many (e.g., hundreds of) agents. Three environments, pursuit, gathering, and battle, are provided, but researchers can use the platform to design customized environments/agents. Open-sourced code is available at <https://github.com/geek-ai/MAgent>.

MuJoCo

MuJoCo is a physics engine that provides accurate simulations for robotic applications Todorov (2012). Single-agent MuJoCo environments have gained popularity, and they are often used for benchmarking and comparing RL algorithms. Several competitive multiagent domains have recently been introduced, including the 2 vs. 2

soccer game (Liu et al. 2019). Open-sourced code for single-agent environments is provided at <https://gym.openai.com/envs/#mujoco>. Open-sourced code for multiagent environments is provided at <https://github.com/openai/multiagent-competition> and https://github.com/deepmind/dm_control/tree/master/dm_control/locotion/soccer.

StarCraft II

StarCraft II is a strategy video game in which a player must collect resources, build structures, and gather combat units to win against an opponent player (for the case of a 1 vs. 1 tournament). StarCraft II is one of the “grand challenges” for AI research because algorithms must overcome several difficult obstacles, such as long-term planning, large action spaces, and imperfect information. Recently, AlphaStar (Vinyals et al. 2019) developed by DeepMind has beaten professional human players. A basic StarCraft II learning environment is provided at <https://github.com/deepmind/pysc2>.

Dota 2

The video game of Dota 2 is another grand challenge for AI research. Dota 2 is a strategy game played between two teams of five players. The five players must collaborate and win against an opponent team. Similar to StarCraft II, Dota 2 has difficult challenges, including long time horizons, high-dimensional observation/action spaces, and partially observed states. The algorithm developed by the OpenAI Five project has defeated professional human teams (OpenAI 2018).

Conclusion and Future Directions

Multiagent reinforcement learning is a promising approach to learning collaborative policies for multiagent systems. However, learning in multiagent settings is difficult due to the issues of non-stationarity (Omidshafiei et al. 2017; Tuyls and Weiss 2012; Hernandez-Leal et al. 2017), equilibrium selection (Omidshafiei et al. 2019; Goldberg et al. 2010; Avis et al. 2010), credit

assignment (han Chang et al. 2004; Foerster et al. 2017), and high-dimensional state spaces (Yang et al. 2018). Despite these difficulties, there have been important recent developments in MARL. This entry has provided an overview of recent work on topics of game theory, communication, coordination, knowledge sharing, and agent modeling (see Section “Current Research Topics in MARL”). This entry has also introduced several extremely useful multiagent simulation platforms (see Section “Domains of MARL”).

There are many open research questions in MARL. For example, it is unclear how existing learning methods can be extended to environments with heterogeneous agents and/or problems in which agents have different observation/action spaces and capabilities. Another important question is fast adaptation in MARL. An ideal agent should be able to adapt quickly to changing teammates, opponents, and environments. However, the existing MARL algorithms are sample inefficient and require thousands of experiences to adapt to the changes. Combining meta-learning approaches (Meta-learning trains a model on a variety of tasks, such that it can learn new skills or adapt to new environments quickly using only a small number of training samples (e.g., learning of initial model parameters (Finn et al. 2017)).) with MARL is one promising future direction to pursue.

Cross-References

- [Model-Free Reinforcement Learning-Based Control for Continuous-Time Systems](#)
- [Reinforcement Learning and Adaptive Control](#)
- [Reinforcement Learning for Control Using Value Function Approximation](#)
- [Robot Learning](#)
- [Stochastic Games and Learning](#)

Recommended Readings

There are several books and surveys to consider (Hernandez-Leal et al. 2018; Tuyls and

Weiss 2012; Hernandez-Leal et al. 2017; Nowe et al. 2012) for more detailed information about MARL.

Acknowledgments This work was supported by IBM (as part of the MIT-IBM Watson AI Lab initiative), Boeing, AWS Machine Learning Research Awards program, and by ARL DCIST under Cooperative Agreement Number W911NF-17-2-0181.

Bibliography

- Amir O, Kamar E, Kolobov A, Grosz BJ (2016) Interactive teaching strategies for agent training. In: International joint conferences on artificial intelligence (IJCAI)
- Avis D, Rosenberg GD, Savani R, von Stengel B (2010) Enumeration of nash equilibria for two-player games. *Econ Theory* 42(1):9–37. [Online]. Available: <https://doi.org/10.1007/s00199-009-0449-x>
- Bowling M (2005) Convergence and no-regret in multiagent learning. In: Saul LK, Weiss Y, Bottou L (eds) *Advances in neural information processing systems* 17. MIT Press, pp 209–216. [Online]. Available: <http://papers.nips.cc/paper/2673-convergence-and-no-regret-in-multiagent-learning.pdf>
- Buşoniu L, Babuška R, De Schutter B (2010) Multiagent reinforcement learning: an overview. Springer, Berlin/Heidelberg, pp 183–221. [Online]. Available: https://doi.org/10.1007/978-3-642-14435-6_7
- Clouse J (1997) On integrating apprentice learning and reinforcement learning
- da Silva FL, Glatt R, Costa AHR (2017) Simultaneously learning and advising in multiagent reinforcement learning. In: International conference on autonomous agents and multiagent systems (AAMAS), pp 1100–1108
- Dayan P (1993) Improving generalization for temporal difference learning: the successor representation. *Neural Comput* 5(4):613–624
- Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning (ICML), ser. Proceedings of machine learning research, vol 70. PMLR, 06–11 Aug 2017, pp 1126–1135
- Foerster J, Assael IA, de Freitas N, Whiteson S (2016) Learning to communicate with deep multi-agent reinforcement learning. In: *Advances in neural information processing systems*. Curran Associates Inc., pp 2137–2145

- Foerster JN, Farquhar G, Afouras T, Nardelli N, Whiteson S (2017) Counterfactual multi-agent policy gradients, CoRR, vol abs/1705.08926. [Online]. Available: <http://arxiv.org/abs/1705.08926>
- Goldberg PW, Papadimitriou CH, Savani R (2010) The complexity of the homotopy method, equilibrium selection, and lemke-howson solutions, CoRR, vol abs/1006.5352. [Online]. Available: <http://arxiv.org/abs/1006.5352>
- Grover A, Al-Shedivat M, Gupta JK, Burda Y, Edwards H (2018) Learning policy representations in multiagent systems, CoRR, vol abs/1806.06464. [Online]. Available: <http://arxiv.org/abs/1806.06464>
- han Chang Y, Ho T, Kaelbling LP (2004) All learning is local: multi-agent learning in global reward games. In: Thrun S, Saul LK, Schölkopf B (eds) *Advances in neural information processing systems* 16. MIT Press, pp 807–814. [Online]. Available: <http://papers.nips.cc/paper/2476-all-learning-is-local-multi-agent-learning-in-global-reward-games.pdf>
- He H, Boyd-Graber JL, Kwok K, III Daumé H (2016) Opponent modeling in deep reinforcement learning, CoRR, vol abs/1609.05559. [Online]. Available: <http://arxiv.org/abs/1609.05559>
- Hernandez-Leal P, Kartal B, Taylor ME (2018) Is multiagent deep reinforcement learning the answer or the question? A brief survey, CoRR, vol abs/1810.05587. [Online]. Available: <http://arxiv.org/abs/1810.05587>
- Hernandez-Leal P, Kaisers M, Baarslag T, de Cote EM (2017) A survey of learning in multiagent environments: dealing with non-stationarity, CoRR, vol abs/1707.09183. [Online]. Available: <http://arxiv.org/abs/1707.09183>
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780. [Online]. Available: <http://dx.doi.org/10.1162/neco.1997.9.8.1735>
- id Software (1999) <https://www.idsoftware.com/>
- Jaderberg M, Dalibard V, Osindero S, Czarnecki WM, Donahue J, Razavi A, Vinyals O, Green T, Dunning I, Simonyan K, Fernando C, Kavukcuoglu K (2017) Population based training of neural networks, CoRR, vol abs/1711.09846. [Online]. Available: <http://arxiv.org/abs/1711.09846>
- Jaderberg M, Czarnecki WM, Dunning I, Marris L, Lever G, Castañeda AG, Beattie C, Rabinowitz NC, Morcos AS, Ruderman A, Sonnerat N, Green T, Deason L, Leibo JZ, Silver D, Hassabis D, Kavukcuoglu K, Graepel T (2018) Human-level performance in first-person multiplayer games with population-based deep reinforcement learning, CoRR, vol abs/1807.01281, 2018. [Online]. Available: <http://arxiv.org/abs/1807.01281>
- Kim D, Liu M, Omidshafiei S, Lopez-Cot S, Riemer M, Habibi G, Tesauro G, Mourad S, Campbell M, How JP (2019) Learning hierarchical teaching in cooperative multiagent reinforcement learning, CoRR, vol abs/1903.03216. [Online]. Available: <http://arxiv.org/abs/1903.03216>
- Lanctot M, Zambaldi VF, Gruslys A, Lazaridou A, Tuyls K, Pérolat J, Silver D, Graepel T (2017) A unified game-theoretic approach to multiagent reinforcement learning. CoRR, vol abs/1711.00832. [Online]. Available: <http://arxiv.org/abs/1711.00832>
- Leyton-Brown K, Shoham Y (2008) *Essentials of game theory: a concise multidisciplinary introduction*. Morgan & Claypool. [Online]. Available: <https://ieeexplore.ieee.org/document/6812710>
- Littman ML (1994) Markov games as a framework for multi-agent reinforcement learning. In: *Proceedings of the eleventh international conference on international conference on machine learning*, ser. ICML'94. Morgan Kaufmann Publishers, San Francisco, pp 157–163. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3091574.3091594>
- Liu S, Lever G, Heess N, Merel J, Tunyasuvunakool S, Graepel T (2019) Emergent coordination through competition. In: *International conference on learning representations*. [Online]. Available: <https://openreview.net/forum?id=BkG8sjR5Km>
- Lowe R, Wu Y, Tamar A, Harb J, Abbeel OP, Mordatch I (2017) Multi-agent actor-critic for mixed cooperative-competitive environments. In: *Advances in neural information processing systems*. NY Curran Associates, Red Hook, pp 6382–6393
- Nowe A, Vrancx P, De Hauwere Y-M (2012) Game theory and multi-agent reinforcement learning. *Adapt Learn Optim* 12:441–470
- Oliehoek FA, Amato C (2016) *A concise introduction to decentralized POMDPs*, ser. SpringerBriefs in intelligent systems. Springer, May 2016. [Online]. Available: <http://www.fransoliehoek.net/docs/OliehoekAmatoI6book.pdf>
- Omidshafiei S, Pazis J, Amato C, How JP, Vian J (2017) Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In: *Proceedings of the 34th international conference on machine learning-volume 70*. JMLR org, pp 2681–2690
- Omidshafiei S, Kim D, Liu M, Tesauro G, Riemer M, Amato C, Campbell M, How JP (2018) Learning to teach in cooperative multiagent reinforcement learning, CoRR, vol abs/1805.07830. [Online]. Available: <http://arxiv.org/abs/1805.07830>
- Omidshafiei S, Papadimitriou CH, Piliouras G, Tuyls K, Rowland M, Lespiau J, Czarnecki WM, Lanctot M, Pérolat J, Munos R (2019) α -rank: multi-agent evaluation by evolution, CoRR, vol abs/1903.01373. [Online]. Available: <http://arxiv.org/abs/1903.01373>
- OpenAI, Openai five (2018) <https://blog.openai.com/openai-five/>
- Panait L, Luke S (2005) Cooperative multi-agent learning: the state of the art. *Auton Agent Multi-Agent Syst* 11(3):387–434
- Ponsen M, Tuyls K, Kaisers M, Ramon J (2009) An evolutionary game-theoretic analysis of poker strategies, *Entertainment Computing*, vol 1, no 1, pp 39–45. [Online]. Available: <http://www.scencedirect.com/science/article/pii/S1875952109000056>
- Rabinowitz NC, Perbet F, Song HF, Zhang C, Eslami SMA, Botvinick M (2018) Machine theory of mind,

- CoRR, vol abs/1802.07740. [Online]. Available: <http://arxiv.org/abs/1802.07740>
- Southey F, Bowling MP, Larson B, Piccione C, Burch N, Billings D, Rayner C (2012) Bayes' bluff: opponent modelling in poker. arXiv preprint arXiv:1207.1411
- Sukhbaatar S, Fergus R et al (2016) Learning multiagent communication with backpropagation. In: Advances in neural information processing systems. Curran Associates Inc., pp 2244–2252
- Taylor ME, Stone P (2009) Transfer learning for reinforcement learning domains: a survey. J Mach Learn Res 10:1633–1685. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1577069.1755839>
- Tesauro G (2004) Extending q-learning to general adaptive multi-agent systems. In: Advances in neural information processing systems. MIT Press, Cambridge, pp 871–878
- Todorov E, Erez T, Tassa Y (2012) Mujoco: a physics engine for model-based control. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026–5033
- Torrey L, Taylor M (2013) Teaching on a budget: agents advising agents in reinforcement learning. In: Proceedings of the 2013 international conference on autonomous agents and multi-agent systems. International Foundation for Autonomous Agents and Multi-agent Systems, pp 1053–1060
- Tuyts K, Weiss G (2012) Multiagent learning: basics, challenges, and prospects. Ai Mag 33:41–52
- Tuyts K, Weiss G (2012) Multiagent learning: basics, challenges, and prospects. Ai Mag 33(3):41–41
- Tuyts K, Pérolat J, Lanctot M, Leibo JZ, Graepel T (2018) A generalised method for empirical game theoretic analysis, CoRR, vol abs/1803.06376. [Online]. Available: <http://arxiv.org/abs/1803.06376>
- Vinyals O, Babuschkin I, Chung J, Mathieu M, Jaderberg M, Czarnecki WM, Dudzik A, Huang A, Georgiev P, Powell R, Ewalds T, Horgan D, Kroiss M, Danihelka I, Agapiou J, Oh J, Dalibard V, Choi D, Sifre L, Sulsky Y, Vezhnevets S, Molloy J, Cai T, Budden D, Paine T, Gulcehre C, Wang Z, Pfaff T, Pohlen T, Wu Y, Yogatama D, Cohen J, McKinney K, Smith O, Schaul T, Lillicrap T, Apps C, Kavukcuoglu K, Hassabis D, Silver D (2019) AlphaStar: mastering the Real-Time Strategy Game StarCraft II. <https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/>
- Wadhwan S, Kim D-K, Omidshafiei S, How JP (2019) Policy distillation and value matching in multiagent reinforcement learning. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China. [Online]. Available: <https://arxiv.org/abs/1903.06592>
- Yang Y, Luo R, Li M, Zhou M, Zhang W, Wang J (2018) Mean field multi-agent reinforcement learning, arXiv preprint arXiv:1802.05438
- Zinkevich M, Balch T (2001) Symmetry in markov decision processes and its implications for single agent and multi agent learning. In: In Proceedings of the 18th international conference on machine learning, Citeseer

Multi-domain Modeling and Simulation

Martin Otter

Institute of System Dynamics and Control,
German Aerospace Center (DLR), Wessling,
Germany

Abstract

One starting point for the analysis and design of a control system is the block diagram representation of a plant. Since it is nontrivial to convert a physical model of a plant into a block diagram, this can be performed manually only for small plant models. Based on research from the last 40 years, more and more mature tools are available to achieve this transformation fully automatically. As a result, multi-domain plants, for example, systems with electrical, mechanical, thermal, and fluid parts, can be modeled in a unified way and can be used directly as input–output blocks for control system design. An overview of the basic principles of this approach is given, and it is shown how to utilize nonlinear, multi-domain plant models directly in a controller. Finally, the low-level “Functional Mockup Interface” standard is sketched to exchange multi-domain models between many different modeling and simulation environments.

Keywords

Block diagram · Differential-algebraic equation (DAE) system · Flow variable · Functional Mockup Interface (FMI) · Inverse models · Modelica · Modia · Object-oriented modeling · Potential variable · Stream variable · Symbolic transformation · VHDL-AMS

Introduction

Methods and tools for control system analysis and design usually require an input–output block

diagram description of the plant to be controlled. Apart from small systems, it is nontrivial to derive such models from first principles of physics. Since a long time, methods and tools are available to construct such models automatically for one domain, for example, a mechanical model, an electronic, or a hydraulic circuit. These domain-specific methods and tools are, however, only of limited use for the modeling of multi-domain systems.

In the dissertation (Elmqvist 1978), a suitable approach for multi-domain, object-oriented modeling has been developed by introducing a modeling language to define models on a high level based on first principles. The resulting differential-algebraic equation (DAE) systems are automatically transformed with proper algorithms in a block diagram description with input and output signals based on ordinary differential equations (ODEs), where the algebraic variables have been eliminated and all derivatives are explicitly solved for.

In 1978, computers were not powerful enough to apply this method on systems with more as a few hundred equations. In the 1990s the technology has been substantially improved, many different modeling languages appeared (and also disappeared), and object-oriented modeling was introduced in commercial simulation environments.

In Table 1, an overview of the most important standards, languages, and tools in the year 2019 for multi-domain modeling is given:

- The *Modelica* language is a standard from the Modelica Association (2017). The first version was released in 1997. The language is accompanied with the *Modelica Standard Library* (<https://github.com/modelica/ModelicaStandardLibrary>) – an open-source Modelica library with about 1600 model components and 1350 functions from many domains. There are several free and commercial software tools supporting the Modelica language and the Modelica Standard Library.
- The *VHDL-AMS* language is a standard from IEEE (IEEE 1076.1-2017 2017), first released in 1999. It is an extension of the widely used

VHDL hardware description language. This language is especially used in the electronics community.

- There are several (proprietary) vendor-specific modeling languages, notably *EcosimPro* and *gRPOMS* for modeling of thermodynamic processes, *Simscape* from MathWorks as an extension to Simulink®, *MAST* the underlying modeling language of *Saber* (Mantooth and Vlach 1992), and the language of *20-sim*, which supports *bond graphs* (Karnopp et al. 2012). Bond graphs are a special graphical notation to define multi-domain systems based on energy flows. It was invented in 1959 by Henry M. Paynter.
- *Modia* and its supporting packages form an open-source prototyping platform based on the *Julia* programming language (Bezanson et al. 2017). *Modia* is the name of a *Julia* package that supports a modeling language called “*Modia language*” as a domain-specific extension of *Julia* and provides algorithms to symbolically transform *Modia* models into DAE systems that can be numerically solved by standard DAE integrators. The *Modia* language and the new algorithms used in the *Modia* package are intended as inspiration for the development of the next *Modelica* generation.

In section “[Modeling Language Principles](#)”, the principles of multi-domain modeling based on a modeling language are summarized. In section “[Models for Control Systems](#)”, it is shown how such models can be used not only for simulation but also as components in nonlinear control systems. Finally, in section “[The Functional Mockup Interface](#)”, an overview about a low-level standard for the exchange of multi-domain systems is described.

Modeling Language Principles

Schematics: The Graphical View

Modelers nowadays require a simple to use graphical environment to build up models. With very few exceptions, multi-domain environments

define models by schematic diagrams. A typical example is given in Fig. 1, showing a simple direct current electrical motor in Modelica.

In the lower left part, the electrical circuit diagram of the DC motor is visible, consisting mainly of the armature resistance and inductance of the motor, a voltage source, and component “emf” to model in an idealized way the electromotoric forces in the air gap. On the lower right part, the motor inertia, a gear box, and a load inertia are present. In the upper part, the heat transfer of the resistor losses to the environment is modeled with lumped elements.

A component, like a resistor, rotational inertia, or convective heat transfer, is shown as an icon in the diagram. On the border of a component, small

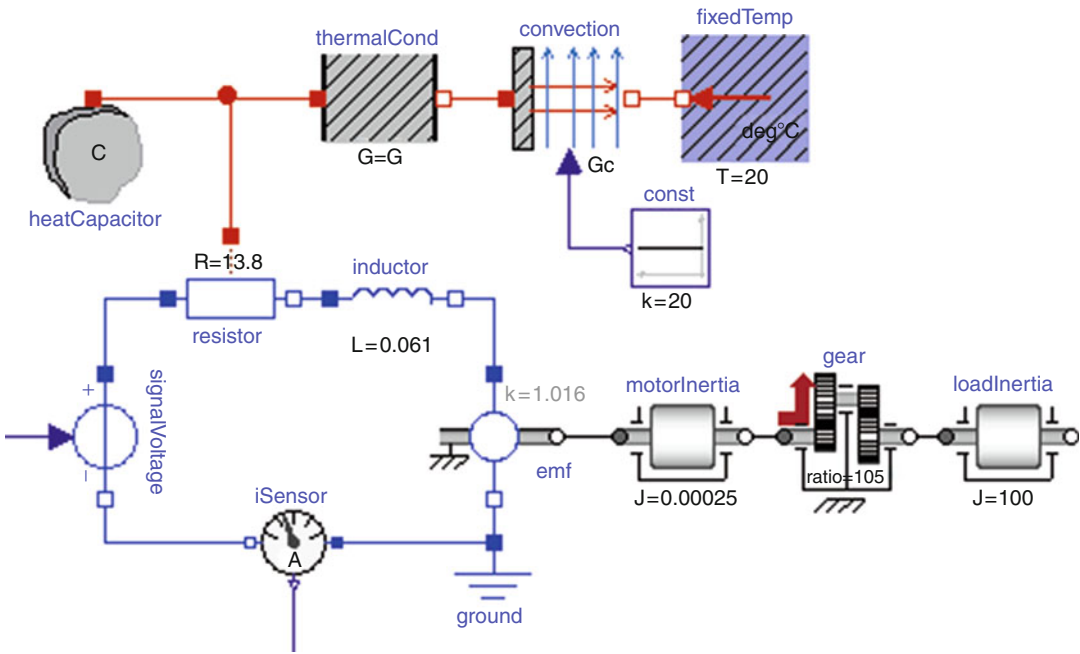
rectangular or circular signs are present representing the “physical ports.” Ports are connected by lines and model the (idealized) physical or signal interaction between ports of different components, for example, the flow of electrical current or heat or the rigid mechanical coupling. Components are built up hierarchically from other components. On the lowest level, components are described textually with the respective modeling language (see section “[Component Equations](#)”).

Coupling Components by Ports

The ports define how a component can interact with other components. A port contains a definition of the variables that describe the interface and defines in which way a tool can automatically

Multi-domain Modeling and Simulation, Table 1
Multi-domain modeling and simulation environments

<i>Environments supporting the Modelica[®] Language Standard (https://www.modelica.org/)</i>	
Altair Activate [®]	https://solidthinking.com/product/activate
ANSYS [®] Twin Builder	https://www.ansys.com/products/systems
Dymola [®]	https://www.dymola.com/
JModelica.org	https://jmodelica.org/
MapleSim	https://www.maplesoft.com/products/maplesim/
MWorks	http://en.tongyuan.cc/
OpenModelica	https://www.openmodelica.org/
OPTIMICA [®] Compiler Toolkit	https://www.modelon.com/
Simcenter [®] Amesim	https://www.plm.automation.siemens.com
SimulationX [®]	https://www.simulationx.com/
Wolfram SystemModeler	https://www.wolfram.com/system-modeler/
<i>Environments supporting the VHDL-AMS Standard (https://standards.ieee.org)</i>	
AMS Designer	https://www.cadence.com/
ANSYS [®] Twin Builder, Simplorer [®]	https://www.ansys.com/products/systems
Saber	https://www.synopsys.com
SMASH [®]	https://www.dolphin-integration.com
SystemVision [®]	https://www.mentor.com/products/sm/
<i>Environments supporting vendor-specific multi-domain modeling languages</i>	
EcosimPro [®]	https://www.ecosimpro.com/
gPROMS	https://www.psenterprise.com/products/gproms
Simscape	https://www.mathworks.com/products/simscape
20-sim	https://www.20sim.com/
<i>Open-source environments supporting other multi-domain modeling languages</i>	
Modia	https://github.com/ModiaSim

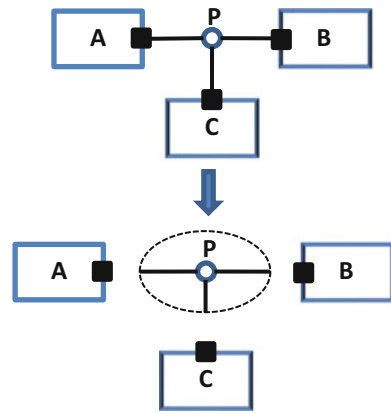


Multi-domain Modeling and Simulation, Fig. 1 Modelica schematic of DC motor with mechanical load and heat losses

construct the equations for the connections. A typical scenario is shown in Fig. 2 where the ports of the three components A, B, C are connected together at one point P.

When cutting away the connection lines, the resulting system consists of three decoupled components A, B, C and a new component around P describing the infinitesimally small connection point. The balance equations and the boundary conditions of the respective domain must hold at all these components. When drawing the connection lines, enough information must be available in the port definitions so that the tool can construct the equations of the infinitesimally small connection points automatically.

To summarize, the component developer is responsible that the balance equations and boundary conditions are fulfilled for every component (A, B, C in Fig. 2), and the tool is responsible that the balance equations and boundary conditions are also fulfilled at the points where the components are connected together (P in Fig. 2). As a consequence, the balance equations and boundary conditions are fulfilled in the overall model containing all components and all connections.



Multi-domain Modeling and Simulation, Fig. 2 Cutting the connections around the connection point P results in three decoupled components A, B, C and a new component around P describing the infinitesimally small connection point

In order that a tool can automatically construct the equations at a connection point, every port variable needs to be associated to a port variable type. In Table 2, some port variable types of Modelica are shown. Hereby it is assumed that $u_1, u_2, \dots, u_n, y, v_1, v_2, \dots, v_n, f_1, f_2, \dots, f_n,$

$s_1, s_2, \dots, s_n$ are corresponding port variables from different components that are connected together at the same point P.

Port variable types “input” and “output” define the usual signal connections in block diagrams. “Potential variables” and “flow variables” are used to define standard physical connections. For example, an electrical port contains the electrical potential and the electrical current at the port, and when connecting electrical ports together, the electrical potentials are identical, and the sum of the electrical currents is zero; see Table 2. These equations correspond exactly to Kirchhoff’s voltage and current laws. “Stream variables” are used to describe the connection semantics of intensive quantities in bidirectional fluid flow, such as specific enthalpy or mass fraction. Here, the idealized balance equation at a connection point states, for example, that the sum of the port enthalpy flow rates is zero and the port enthalpy flow rate is computed as the product of the mass flow rate (a flow variable f_i) and the directional specific enthalpy s_i , which is either the (yet unknown) mixing-specific enthalpy s_{mix} when the flow is from the connection point to the port or the specific enthalpy s_i in the port when the flow is from the port to the connection point. More details and explanations are available from Franke et al. (2009). In Table 3 some of the port definitions are shown that are provided by the Modelica Standard Library and the PlanarMechanics library.

Component Equations

Implementing a component in a modeling language means to define the ports of the component and to provide the equations describing the rela-

tionships between the port variables. For example, an electrical capacitor with constant capacitance C can be defined by the equations in the right side of Fig. 3.

Such a component has two ports, the pins “p” and “n,” and the port variables are the electrical currents i_p, i_n flowing into the respective ports and the electrical potentials v_p, v_n at the ports. The first component equation states that if the current i_p at port “p” is positive, then the current i_n at port “n” is negative (therefore, the current flowing into “p” is flowing out of “n”). Furthermore, the two remaining equations state that the derivative of the difference of the port potentials is proportional to the current flowing into port “p.”

One important question is how many equations are needed to describe such a component? For an input–output block, the answer is simple: all input variables are known, and for all other variables, one equation per unknown is needed. Counting equations for physical components, such as a capacitor, are more involved: the requirement that *any type of component connections shall always result in identical numbers of unknowns and equations of the overall system* leads to the following counting rule (for a proof, see Olsson et al. 2008):

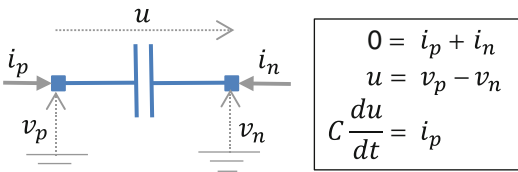
- 1. The number of potential and the number of flow variables in a port must be identical.
- 2. Input variables and variables that appear differentiated are treated as known variables.
- 3. The number of equations of a component must be equal to the number of unknowns minus the number of flow variables.

Multi-domain Modeling and Simulation, Table 2 Some port variable types in Modelica

Port variable type	Connection semantics
Input variables u_i , output variable y	$u_1 = u_2 = \dots = u_n = y$ (exactly one output variable can be connected to n input variables)
Potential variables v_i	$v_1 = v_2 = \dots = v_n$
Flow variables f_i	$0 = \sum f_i$
Stream variables s_i (with associated flow variables f_i)	$0 = \sum f_i \hat{s}_i; \hat{s}_i = \begin{cases} s_{\text{mix}} & \text{if } f_i > 0 \\ s_i & \text{if } f_i \leq 0 \end{cases}$ ($0 = \sum f_i$)

Multi-domain Modeling and Simulation, Table 3
Some port definitions from Modelica

Domain	Port variables
Electrical analog	Electrical potential in [V] (<i>pot.</i>) electrical current in [A] (<i>flow</i>)
Electrical multiphase	Vector of electrical ports
Electrical quasi-stationary	Complex electrical potential (<i>pot.</i>) complex electrical current (<i>flow</i>)
Magnetic flux tubes	Magnetic potential in [A] (<i>pot.</i>) magnetic flux in [Wb] (<i>flow</i>)
Translational (one-dimensional mechanics)	Distance in [m] (<i>pot.</i>) cut-force in [N] (<i>flow</i>)
Rotational (one-dimensional mechanics)	Absolute angle in [rad] (<i>pot.</i>) cut-torque in [Nm] (<i>flow</i>)
Two-dimensional mechanics	Position in x-direction in [m] (<i>pot.</i>) position in y-direction in [m] (<i>pot.</i>) absolute angle in [rad] (<i>pot.</i>) cut-force in x-direction in [N] (<i>flow</i>) cut-force in y-direction in [N] (<i>flow</i>) cut-torque in z-direction in [Nm] (<i>flow</i>)
Three-dimensional mechanics	Position vector in [m] (<i>pot.</i>) transformation matrix in [1] (<i>pot.</i>) cut-force vector in [N] (<i>flow</i>) cut-torque vector in [Nm] (<i>flow</i>)
One-dimensional heat transfer	Temperature in [K] (<i>pot.</i>) heat flow rate in [W] (<i>flow</i>)
One-dimensional thermo-fluid pipe flow	Pressure in [Pa] (<i>pot.</i>) mass flow rate in [kg/s] (<i>flow</i>) spec. enthalpy in [J/kg] (<i>stream</i>) mass fractions in [1] (<i>stream</i>)



Multi-domain Modeling and Simulation, Fig. 3 Equations of a capacitor component

In the example of the capacitor, there are five unknowns (i_p , i_n , v_p , v_n , du/dt) and two flow variables (i_p , i_n). Therefore, $5 - 2 = 3$ equations are needed to define the component in Fig. 3.

Modeling languages are used to provide a textual description of the ports and of the equations in a specific syntax. For example, in Modelica the capacitor from Fig. 3 can be defined as shown in Fig. 4 (keywords of the language are written in boldface). In VHDL-AMS the capacitor model can be defined as shown in Fig. 5.

One difference between Modelica and VHDL-AMS is that in Modelica, all equations need to be explicitly given, and port variables (such as $p.i$) can be directly accessed in the model (Fig. 4). Instead, in VHDL-AMS (and some other modeling languages), port variables cannot be accessed in a model, and instead via the “**quantity .. across .. through .. to ..**” construction, the relationships between the port variables are implicitly defined and correspond to the Modelica equations “ $0 = p.i + n.i$ ” and “ $u = p.v - n.v$ ”.

Simulation of Multi-domain Systems

Collecting all the component equations of a multi-domain system model together with all connection equations results in a DAE system:

$$\mathbf{0} = \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{w}, \mathbf{y}, \mathbf{u}, t) \quad (1)$$

where $t \in \mathbb{R}$ is time, $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is a vector which contains variables that appear differenti-


```

type Voltage = Real (unit="V");
type Current = Real (unit="A");

connector Pin
    Voltage v;
    flow Current i;
end Pin;

model Capacitor
    parameter Real C(unit="F");
    Pin p,n;
    Voltage u;

    equation
        0 = p.i + n.i;
        u = p.v - n.v;
        C*der(u) = p.i;
end Capacitor;

```

Multi-domain Modeling and Simulation,
Fig. 4 Modelica model of capacitor component

```

subtype voltage is real;
subtype current is real;
nature electrical is
    voltage across
    current through
    electrical_ref reference;

entity CapacitorInterface IS
    generic(C: real);
    port (terminal p, n: electrical);
end entity CapacitorInterface;

architecture SimpleCapacitor of
    CapacitorInterface is
    quantity u across i through p to n;
begin
    i == C*u'dot;
end architecture SimpleCapacitor;

```

Multi-domain Modeling and Simulation,
Fig. 5 VHDL-AMS model of capacitor component

ated, $\mathbf{w}(t) \in \mathbb{R}^{n_w}$ is a vector of algebraic variables, $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ is a vector of output variables, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is a vector of input variables, and $\mathbf{f} \in \mathbb{R}^{n_x+n_w+n_y}$ is a vector of functions that represent the right-hand sides of the equations of the DAE system. The equations in (1) can be solved numerically with an integrator for DAE systems; see, for example, Brenan et al. (1996). For DAEs that are linear in their unknowns, a complete theory for solvability is available based

on *matrix pencils*, see, for example, Brenan et al. (1996), and also reliable software for their analysis (Varga 2000).

Unfortunately, only certain classes of *nonlinear* DAEs can be *directly* solved numerically in a reliable way. For example, there are DAEs where no unique solution exists anymore when the step-size of the integrator approaches zero. For such systems step-size control is difficult, and numerical integration may easily fail. Domain-specific software, as, e.g., for mechanical systems, transforms the underlying DAE system into a form that can be more reliably solved using domain-specific knowledge. Hereby, certain equations of the DAE system are analytically differentiated, and special integration methods are used to solve the resulting overdetermined set of DAEs. In multi-domain simulation software, the following approaches are used:

- (a) The DAE system (1) is directly solved numerically using an implicit integration method, such as a linear multistep method. Typically, all VHDL-AMS simulators use this approach.
- (b) The DAE system (1) is symbolically transformed in a form that is equivalent to a set of ODEs, and then either explicit or implicit ODE or DAE integration methods are used to numerically solve the transformed system. The transformation is usually based on the algorithms of Pantelides (1988) and of Mattsson and Söderlind (1993) or some variants of them and might require to analytically differentiate equations. Typically, Modelica-based simulators, but also EcosimPro, use this approach.

For many models both approaches can be applied successfully. There are, however, systems where approach (a) is successful and fails for (b) or vice versa.

DAEs (1) derived from modeling languages usually have a large number of equations but with only a few unknowns in every equation. In order to solve DAEs of this kind efficiently, both with (a) or (b), typically graph theory and/or sparse matrix methods are utilized. For method (b) the

fundamental algorithms have been developed by Elmqvist (1978) and later improved in further publications. For a recent survey and comparison of some of the algorithms, see Frenkel et al. (2012).

Solving the DAE system (1) means to solve an initial value problem. In order that integration can be started, a consistent set of initial variables $\dot{\mathbf{x}}_0 = \dot{\mathbf{x}}(t_0)$, $\mathbf{x}_0 = \mathbf{x}(t_0)$, $\mathbf{w}_0 = \mathbf{w}(t_0)$, $\mathbf{y}_0 = \mathbf{y}(t_0)$, and $\mathbf{u}_0 = \mathbf{u}(t_0)$ has to be determined first at the initial time t_0 which is a nontrivial task. For example, often (1) shall start in steady state, that is, it is required that $\dot{\mathbf{x}}_0 := \mathbf{0}$ and therefore at the initial time (1) is required to satisfy

$$\mathbf{0} = \mathbf{f}(\mathbf{0}, \mathbf{x}_0, \mathbf{w}_0, \mathbf{y}_0, \mathbf{u}_0, t_0) \quad (2)$$

(2) is an algebraic system of nonlinear equations in the unknowns $\mathbf{x}_0$, $\mathbf{w}_0$, $\mathbf{y}_0$, and $\mathbf{u}_0$. These are $n_x + n_w + n_y$ equations for $n_x + n_w + n_y + n_u$ unknowns. Therefore, n_u further conditions must be provided (usually some elements of $\mathbf{u}_0$ and/or $\mathbf{y}_0$ are fixed to desired physical values). Solving (2) for the unknowns is also called “DC operating point calculation” or “trimming.” Nonlinear equation solvers are based on iterative methods that require usually a sufficiently accurate initial guess for all unknowns. In a large multi-domain system model, it is not practical that reasonable initial values must be provided for every DAE variable and therefore methods are needed to solve (2) even if generic guess values in a library are provided that might be far from the solution of the system at hand.

For analog electronic circuit simulations, a large body of theory, algorithms, and software is available to solve (2) based on homotopy methods. The basic idea is to solve a sequence of nonlinear algebraic equation systems by starting with an easy to solve simplified system, characterized by the homotopy parameter $\lambda = 0$. This system is continuously “deformed” until the desired one is reached at $\lambda = 1$. The solution at iteration i is used as guess value for iteration $i + 1$, and at every iteration, the solution is usually computed with a Newton–Raphson method. The simplest such approach is “source stepping”: the initial guess values of all electrical components

are set to “zero voltage” and/or “zero current.” All (voltage and current) sources start at zero, and their values are gradually increased until the desired source values are reached. This method may not converge, typically due to the severe nonlinearities at switching thresholds in logical circuits.

There are several, more involved approaches, called “probability-one homotopy” methods. For these method classes, proofs exist that they converge for all initial conditions globally with probability one (so practically always). These algorithms can only be applied for certain classes of DAEs; see, for example, the “variable stimulus probability-one homotopy” of Melville et al. (1993).

Although strong results exist for analog electrical circuit simulators, it is difficult to generalize them to the large class of multi-domain systems covered by a modeling language. In Modelica a “homotopy” operator was introduced into the language (Sielemann et al. 2011) in order that a library developer can formulate simple homotopy methods like the “source stepping” in a component library. A generalization of probability-one methods for multi-domain systems was developed in the dissertation of Sielemann (2012) and was successfully applied to air distribution systems described as one-dimensional thermo-fluid pipe flow.

Models for Control Systems

Models for Analysis

The multi-domain models from section “[Modeling Language Principles](#)” can be utilized to evaluate the properties of a control system by simulation. Also control systems can be designed by nonlinear optimization where at every optimization step one or several simulations of a plant model are executed. Furthermore, modeling environments usually provide pre- and/or post-processing methods to linearize the nonlinear DAE system (1) around a steady-state operating point or an operating point along the simulation trajectory

$$\begin{aligned} \mathbf{x}(t) &\approx \mathbf{x}_{op} + \Delta \mathbf{x}(t), & \mathbf{w}(t) &\approx \mathbf{w}_{op} + \Delta \mathbf{w}(t), \\ \mathbf{y}(t) &\approx \mathbf{y}_{op} + \Delta \mathbf{y}(t), & \mathbf{u}(t) &\approx \mathbf{u}_{op} + \Delta \mathbf{u}(t) \end{aligned} \quad (3)$$

resulting in

$$\begin{aligned} \Delta \dot{\mathbf{x}}_{\text{red}} &= \mathbf{A} \Delta \mathbf{x}_{\text{red}} + \mathbf{B} \Delta \mathbf{u} \\ \Delta \mathbf{y} &= \mathbf{C} \Delta \mathbf{x}_{\text{red}} + \mathbf{D} \Delta \mathbf{u} \end{aligned} \quad (4)$$

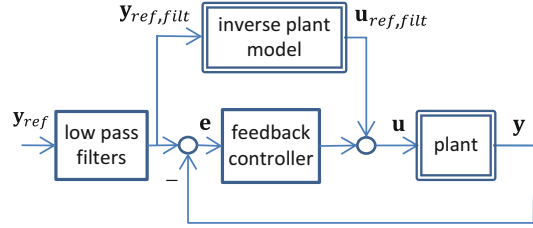
where $\mathbf{x}_{op}$, $\mathbf{w}_{op}$, $\mathbf{y}_{op}$, and $\mathbf{u}_{op}$ define the operating point, $\Delta \mathbf{x}_{\text{red}}$ is a vector consisting of elements of $\Delta \mathbf{x}$, vector $\Delta \mathbf{w}$ is eliminated by exploiting the algebraic constraints, and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are constant matrices. Simulation tools provide linear analysis and synthesis methods on this linearized system and/or export it for usage in an environment such as Julia, Maple, Mathematica®, MATLAB®, or Python®.

Multi-domain models might also be used directly in nonlinear Kalman filters, moving horizon estimators, or nonlinear model predictive control. For example, the company ABB is using moving horizon estimation and nonlinear model predictive control based on Modelica models in its product OPTIMAX® to significantly improve the start-up process of power plants (Franke and Doppelhamer 2006), for the control of virtual power plants and for other purposes.

Inverse Models

A large body of literature exists about the theory of nonlinear control systems that are based on *inverse* plant models; see, for example, (Isidori 1995). Methods such as feedback linearization, nonlinear dynamic inversion, or flat systems use an inverse plant model in the control loop. However, a major obstacle is how to *automatically* utilize an inverse plant model in a controller without being forced to manually set up the equations in the needed form which is not practical for larger systems. Modeling languages can solve this problem as discussed below.

Nonlinear inverse models can be utilized in various ways in a control system. The simplest approach, as feed forward controller, is shown in Fig. 6. Under the assumption that identical equations (1) are used for the plant and the inverse plant model (but the plant model has $\mathbf{u}$ as input and $\mathbf{y}$ as output, whereas the inverse plant



Multi-domain Modeling and Simulation, Fig. 6 Controller with inverse plant model in the feed forward path. The inverse plant model needs usually also derivatives of $\mathbf{y}_{\text{ref}}$ as inputs. These derivatives are provided by appropriate low-pass filters

model has $\mathbf{y}$ as input and $\mathbf{u}$ as output) and start at the same initial state, then from the construction, the control error $\mathbf{e}$ is zero and $\mathbf{y} = \mathbf{y}_{\text{ref},\text{filt}}$, so $\mathbf{y}$ is identical to the low-pass filtered reference signals. In other words, $\mathbf{y} \approx \mathbf{y}_{\text{ref}}$ for reference signals that have a frequency spectrum below the cutoff frequency of the low-pass filters. Since the assumption is actually not fulfilled, there will be a nonzero control error $\mathbf{e}$, and the feedback controller has to cope with it. This controller structure with a nonlinear inverse plant model has the advantage that the feed forward part is useful over the complete operating range of the plant.

Various other structures with nonlinear plant models are discussed in Looye et al. (2005), such as compensation controllers, feedback linearization controllers, and nonlinear disturbance observers. In Olsson et al. (2017), a Modelica extension is defined to directly construct (sampled data) feedback linearization controllers from (continuous-time) Modelica plant models. Furthermore, integration algorithms are proposed to solve nonlinear inverse models with hard real-time requirements.

It turns out that nonlinear inverse plant models can be generated automatically with the techniques that have been developed for modeling languages; see section “[Modeling Language Principles](#).” In particular, constructing an inverse model from (1) means that the inputs $\mathbf{u}$ are defined to be outputs, so they are no longer knowns but unknowns, and outputs $\mathbf{y}$ are defined to be inputs, so they are no longer unknowns but knowns. The resulting system is still a DAE

system and can therefore be handled as any other DAE system. Therefore, defining an inverse model with a modeling language just requires exchanging the definition of input and output signals. In Modelica, this can be graphically performed with the nonstandard input–output block from Fig. 7. This block has two inputs and two outputs and is described by the equations:

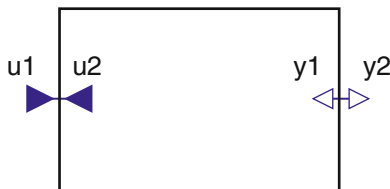
$$u1 = u2; \quad y1 = y2;$$

From a block diagram point of view, this looks strange. However, from a DAE point of view, this just states constraints between two input and two output signals. In Fig. 8, it is shown how this block can be used to invert a simple second-order system: the output of the low-pass filter is connected to the output of the second-order system (note, a direction connection of this kind violates the input–output block semantics, but an indirect connection via the InverseBlockConstraint block is possible) and therefore this model computes the input of the second-order system, from the input of the filter by inverting the second-order system.

A Modelica environment will generate from this type of definition the inverse model, thereby differentiating equations analytically and solving

algebraic variables of the model in a different way as for a simulation model. The whole transformation is nontrivial, but it is just the standard method used by Modelica tools as for any other type of DAE system.

The question arises whether a solution of the inverse model exists or is unique and whether the model is stable (otherwise, it cannot be applied in a control system). In general, a nonlinear inverse model consists of linear and/or nonlinear algebraic equation systems and of linear and/or nonlinear differential equations. Therefore, from a formal point of view, the same theorems as for a general DAE system applies, see Brenan et al. (1996). Furthermore, all these equations need to be solved with a numerical method. For some classes of systems, it can be shown that a unique mathematical solution exists and that the system is stable. However, in general, one cannot expect that it is possible to provide such a proof for complex inverse plant models. In some cases it is possible to approximate the plant model so that its inverse is guaranteed to be stable. In more involved cases, it might only be possible to perform simulations at many operating points to show that the probability of a stable inverse model is high. Note, nonlinear inverse plant models have been successfully utilized by automatic generation from a Modelica plant model, in particular for robots, satellites, aircrafts, vehicles, and thermo-fluid systems.

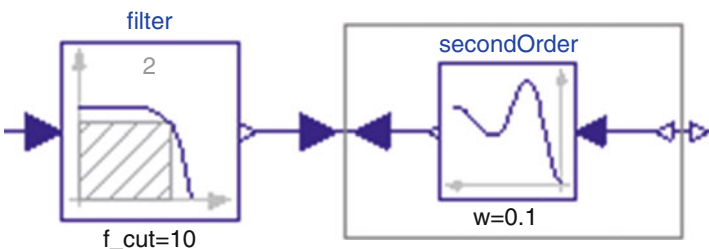


Multi-domain Modeling and Simulation,
Fig. 7 Modelica InverseBlockConstraint block

The Functional Mockup Interface

Many different types of simulation environments are in use. One cannot expect that a generic approach as sketched in section “Modeling

Multi-domain Modeling and Simulation,
Fig. 8 Inversion of a second-order system in Modelica



Language Principles” will replace all these environments with their rich set of domain-specific knowledge, analysis, and synthesis features. Practically, all simulation environments provide a vendor-specific interface in order that a user can import components that are not describable by the simulation environment itself. Typically, this requires to provide a component as a set of C or Fortran functions with a particular calling interface. In the control community, the most widely used approach of this kind is the S-Function interface from the MathWorks, where Simulink is used as integration platform, and model components from other environments are imported as S-Functions.

In 2010 the vendor-independent standard “Functional Mockup Interface 1.0” was developed, followed by a significantly improved version 2.0 in 2014 (Modelica Association 2014). An again significantly improved version 3.0 is under development as of June 2019 (see <https://fmi-standard.org/faq/>). FMI is a low-level standard for the exchange of models between different simulation environments. This standard allows to exchange only the model equations (called “FMI for Model Exchange”) and/or the model equations with an embedded solver (called “FMI for Co-Simulation”). This standard was quickly adopted by many simulation environments, and in 2019 there are more than 130 tools that support it (for an actual list of tools, see <https://fmi-standard.org/tools/>). In particular Modelica environments export Modelica models in this format, and therefore, Modelica multi-domain models can be imported in other environments with low effort.

A software component which implements the FMI is called Functional Mockup Unit (FMU). An FMU consists of one zip file with extension “.fmu” containing all necessary components to utilize the FMU either for Model Exchange, for Co-Simulation, or for both. The following summary is an adapted version from Blochwitz et al. (2012):

1. An *XML-file* contains the definition of all exposed variables of the FMU, as well as other model information. It is then possible to

run the FMU on a target system without this information, that is, without unnecessary overhead. Furthermore, this allows determining all properties of an FMU from a text file, without actually loading and running the FMU.

2. A set of *C-functions* is provided to execute model equations for the Model Exchange case and to simulate the equations for the Co-Simulation case. These C-functions can be provided either in binary form for different platforms or in source code. The different forms can be included in the same model zip file.
3. Further data can be included in the FMU zip file, especially a model icon (bitmap file), documentation files, maps and tables needed by the model, and/or all object libraries or DLLs that are utilized.

The main application area of FMI is offline simulation, but it is also used for (soft) real-time applications, see, for example, the direct support of FMI in the real-time systems of dSPACE (<https://www.dspace.com>). In Brembeck (2019) a nonlinear (continuous-time) Modelica model of a highly maneuverable electric vehicle is used as starting point of a dedicated tool chain to generate automatically (sampled data) FMUs used as discrete-time prediction models for an extended moving horizon state estimator and an extended Kalman filter.

Summary and Future Directions

Multi-domain modeling based on a DAE description and defined with a modeling language is an established approach, and many tools support it. This allows to conveniently define plant models from many domains for the design and evaluation of control systems. Furthermore, nonlinear inverse plant models can be easily constructed with the same methodology and can be utilized in various ways in nonlinear control systems.

Current research focuses on the support of the complete life cycle: defining requirements of a system formally on a “high level,” considerably improving testing by checking these

requirements automatically when evaluating a system design by simulations, see, for example, Bouskela and Jardin (2018), and providing complete tool chains to embedded systems. For example, in the European ITEA project EMPHYSIS (embedded systems with physical models in the production code software; <https://itea3.org/project/emphysis.html>), the variant eFMI of FMI is being developed in the years 2017–2021 so that whole tool chains from multi-domain modeling environments to production code on automotive electronic control units and other embedded devices with *hard real-time requirements* become feasible. This will allow convenient and fast target code generation of nonlinear controllers, optimization-based controllers, extended and unscented Kalman filters, or moving horizon estimators from multi-domain models.

Furthermore, the methodology itself is further improved. Especially, with the prototyping platform Modia (Elmqvist et al. 2017), new directions are evaluated, such as hybrid 2D schematic/3D graphical user interfaces, close integration of complex data structures with equation-based modeling (e.g., to model multiphase, multicomponent fluids or collision handling of 3D mechanical systems), improved and new symbolic transformation algorithms (Otter and Elmqvist 2017), code generation for large models, support of multi-domain Dirac impulses, and changing the number of equations during simulation.

Cross-References

- [Computer-Aided Control Systems Design: Introduction and Historical Overview](#)
- [Descriptor System Techniques and Software Tools](#)
- [Extended Kalman Filters](#)
- [Feedback Linearization of Nonlinear Systems](#)
- [Interactive Environments and Software Tools for CACSD](#)
- [Model Building for Control System Synthesis](#)
- [Modeling of Dynamic Systems from First Principles](#)

- [Model Predictive Control in Practice](#)
- [Moving Horizon Estimation](#)
- [Nonlinear Filters](#)

Bibliography

- Bezanson J, Edelman A, Karpinski S, Shah V (2017) Julia: a fresh approach to numerical computing. *SIAM Rev* 59:65–98. <https://doi.org/10.1137/141000671>
- Blochowitz T, Otter M, Akeesson J, Arnold M, Clauß C, Elmqvist H, Friedrich M, Junghanns A, Mauss J, Neumerkel D, Olsson H, Viel A (2012) Functional Mockup Interface 2.0: the standard for tool independent exchange of simulation models. In: *Proceedings of the 9th international modelica conference*, Munich, 3–5 Sept 2012, pp 173–184. <https://doi.org/10.3384/ecp12076173>
- Bouskela D, Jardin A (2018) A new temporal language for the verification of cyber-physical systems. In: *2018 annual IEEE international systems conference (SysCon)*. <https://doi.org/10.1109/SYSCON.2018.8369502>
- Brembeck J (2019) Nonlinear constrained moving horizon estimation applied to vehicle position estimation. *Sensors* 2019, 19, 2276. <https://doi.org/10.3390/s19102276>
- Brenan KE, Campbell SL, Petzold LR (1996) *Numerical solution of initial-value problems in differential-algebraic equations*. SIAM, Philadelphia
- Elmqvist H (1978) *A structured model language for large continuous systems*. Dissertation. Report CODEN:LUTFD2/(TFRT-1015), Department of Automatic Control, Lund Institute of Technology, Lund. <https://lup.lub.lu.se/search/ws/files/4602422/8570492.pdf>
- Elmqvist H, Henningsson T, Otter M (2017) Innovations for future modelica. In: *Proceedings of the 12th international modelica conference*, Prague. <https://doi.org/10.3384/ecp17132693>
- Franke R, Doppelhamer J (2006) Online application of modelica models in the industrial IT extended automation system 800xA. In: *Proceedings of the 6th international modelica conference*, Vienna, 4–5 Sept 2006, pp 293–302. <https://modelica.org/events/modelica2006/Proceedings/sessions/Session3c2.pdf>
- Franke R, Casella F, Otter M, Sielemann M, Mattsson SE, Olsson H, Elmqvist H (2009) Stream connectors – an extension of modelica for device-oriented modeling of convective transport phenomena. In: *Proceedings of the 7th international modelica conference*, Como, pp 108–121. <https://doi.org/10.3384/ecp09430078>
- Frenkel J, Kunze G, Fritzson P (2012) Survey of appropriate matching algorithms for large scale systems of differential algebraic equations. In: *Proceedings of the 9th international modelica Conference*, Munich, 3–5 Sept 2012, pp 433–442. <https://doi.org/10.3384/ecp12076433>

- IEEE 1076.1-2017 (2017) IEEE standard VHDL analog and mixed-signal extensions. Standard of IEEE. <https://standards.ieee.org/standard/1076.1-2017.html>
- Isidori A (1995) Nonlinear control systems, 3rd edn. Springer, Berlin/New York
- Karnopp DC, Margolis DL, Rosenberg RC (2012) System dynamics: modeling, simulation, and control of mechatronic systems, 5th edn. Wiley, Hoboken
- Looye G, Thümmel M, Kurze M, Otter M, Bals J (2005) Nonlinear inverse models for control. In: Proceedings of the 4th international modelica conference, Hamburg, 7–8 Mar 2005, pp 267–279. https://modelica.org/events/Conference2005/online_proceedings/Session3/Session3c3.pdf
- Mantooth HA, Vlach M (1992) Beyond spice with saber and MAST. In: IEEE international symposium on circuits and systems, San Diego, vol 1, 10–13 May 1992, pp 77–80
- Mattsson SE, Söderlind G (1993) Index reduction in differential-algebraic equations using dummy derivatives. SIAM J Sci Comput 14:677–692
- Melville RC, Trajkovic L, Fang SC, Watson LT (1993) Artificial parameter homotopy methods for the DC operating point problem. IEEE Trans Comput Aided Des Integr Circuits Syst 12:861–877. <https://doi.org/10.1109/43.229761>
- Modelica Association (2014) Functional mock-up interface for model exchange and co-simulation, Version 2.0. <https://fmi-standard.org/downloads/>
- Modelica Association (2017) Modelica – a unified object-oriented language for systems modeling. Language specification, Version 3.4. <https://www.modelica.org/documents/ModelicaSpec34.pdf>
- Olsson H, Otter M, Mattsson SE, Elmqvist H (2008) Balanced models in modelica 3.0 for increased model quality. In: Proceedings of the 6th international modelica conference, Bielefeld, pp 21–33. <https://www.modelica.org/events/modelica2008/Proceedings/sessions/session1a3.pdf>
- Olsson H, Mattsson SE, Otter M, Pfeiffer A, Brger C, Henriksson D (2017) Model-based embedded control using Rosenbrock integration methods. In: Proceedings of the 12th international modelica conference, Prague. <https://doi.org/10.3384/ecp17132517>
- Otter M, Elmqvist H (2017) Transformation of differential algebraic array equations to index one form. In: Proceedings of the 12th international modelica conference, Prague. <https://doi.org/10.3384/ecp17132565>
- Pantelides CC (1988) The consistent initialization of differential-algebraic systems. SIAM J Sci Stat Comput 9:213–233
- Sielemann M (2012) Device-oriented modeling and simulation in aircraft energy systems design. Dissertation, Dr. Hut. ISBN:978-3-8439-0504-6
- Sielemann M, Casella F, Otter M, Clauß C, Eborn J, Mattsson SE, Olsson H (2011) Robust initialization of differential-algebraic equations using homotopy. In: Proceedings of the 8th international modelica conference, Dresden. <https://doi.org/10.3384/ecp1106375>

- Varga A (2000) A descriptor systems toolbox for Matlab. In: Proceedings of CACSD'2000 IEEE international symposium on computer-aided control system design, Anchorage, 25–27 Sept 2000, pp 150–155. <https://ieeexplore.ieee.org/iel5/7209/19415/00900203.pdf>

Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ Control Designs

Xiang Chen¹ and Kemin Zhou²

¹Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada

²School of Electrical and Automation Engineering, Shangdong University of Science and Technology, Shangdong, China

Abstract

In this entry, the implementation of Youla-Kučera parameterization of all stabilizing controllers for feedback control systems is first presented. Robust and optimal control designs are then discussed within the multi-objective design framework. In particular, $\mathcal{H}_\infty$ Gaussian control design and a linear-quadratic-Gaussian (LQG) control with an extra $\tilde{Q}$ -regulation design are showcased and compared. The pros and cons of the two different multi-objective optimal control approaches are compared and the potential further research activities are discussed for the multi-objective control design.

Keywords

Youla-Kučera parameterization · Multi-objective control · $\mathcal{H}_\infty$ Gaussian control · LQG control · $\tilde{Q}$ -regulation

Introduction

Multi-objective optimal designs play significant roles in control system theory due to the fact that multi-performances are always of concerns in

practical systems. The traditional multi-objective optimal control is normally conducted in one integrated controller framework, that is, a single controller structure is developed to address all performance expectations simultaneously. However, the stringent reality facing various expectations is that they might be inherently conflicting to each other, such as the well-known conflicting pair of robustness and optimality in a robust control system. Therefore, in the case of one controller design, the best strategy would be (and has to be) to achieve a trade-off between conflicting performance requirements, which would then incur the fundamental drawback for such kind of control designs, the conservativeness. An ideal remedy of this drawback would call for “complementary control” design framework instead of conventional “trade-off control,” where no significant progress has been observed so far. The main contents of this entry, thus, is to demonstrate and compare the ideas of “one- and two-feedback controller” designs through showcasing the $\mathcal{H}_\infty$ Gaussian control (Chen and Zhou 2001) and an LQG control with extra $\tilde{Q}$ -regulation design proposed most recently in Chen et al. (2019), which is essentially motivated by the equivalent “multi-block” implementation of Youla-Kučera parameterization of all stabilizing controllers (Zhou and Ren 2001; Kučera 1979; Youla et al. 1976a). While both designs are essentially for $\mathcal{H}_2/\mathcal{H}_\infty$ performance, the later apparently facilitates a new multi-objective optimal design framework. The pros and cons of these two different designs are then compared, and the further research activities are suggested for the new multi-objective control design framework.

This entry is divided into five sections. After the Introduction, the Youla-Kučera parameterization of all stabilizing controllers is presented for a feedback system. Section “ $\mathcal{H}_\infty$ Gaussian Control” will be devoted to the $\mathcal{H}_\infty$ Gaussian control design, while, in section “ LQG Control with $\tilde{Q}$ -Regulation”, the multi-objective optimal design is demonstrated by a newly developed LQG control with extra $\tilde{Q}$ -regulation design. The entry is then concluded in section “Summary and Future Directions” with some discussions

for the two design approaches and for the future research activities.

The symbols used are standard. I is an identity matrix with the consistent dimension in contexts. $\delta(t)$ is an impulse function. The transpose of a matrix A is represented by A^T , and the determinant of a square matrix A is denoted as $\det(A)$. $G(s) = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = D + C(sI - A)^{-1}B$ with (A, B, C, D) being a realization of $G(s)$ in the state space. $\mathcal{RH}_\infty$ is the space of all proper and real rational stable transfer function matrices with the norm $\|T(s)\|_\infty = \sup_{\omega} \bar{\sigma}\{T(j\omega)\}$ for any $T(s) \in \mathcal{RH}_\infty$, where $\bar{\sigma}\{\cdot\}$ is the largest singular value of the underlying matrix $\rho(X) = \max_i |\lambda_i(X)|$ is the spectral radius of a matrix X (Zhou et al. 1996), where $\lambda_i(X)$, $i = 1, 2, \dots, n$ are the eigenvalues of X .

Youla-Kučera Parameterization and Implementation

We start with the definition of the coprime factorization over $\mathcal{RH}_\infty$ of a transfer function matrix $P(s)$.

Definition 1 (Zhou et al. 1996) Given a proper real rational transfer function matrix $P(s)$, a right coprime factorization of $P(s)$ over $\mathcal{RH}_\infty$ is defined as $P(s) = N(s)M^{-1}(s)$, where $N(s), M(s) \in \mathcal{RH}_\infty$ and satisfy the Bezout identity:

$$\begin{bmatrix} X_r(s) & Y_r(s) \end{bmatrix} \begin{bmatrix} M(s) \\ N(s) \end{bmatrix} = I, \quad \text{or} \\ X_r(s)M(s) + Y_r(s)N(s) = I,$$

for some $X_r(s), Y_r(s) \in \mathcal{RH}_\infty$.

Correspondingly, a left coprime factorization of $P(s)$ over $\mathcal{RH}_\infty$ is defined as $P(s) = \tilde{M}^{-1}(s)\tilde{N}(s)$, where $\tilde{N}(s), \tilde{M}(s) \in \mathcal{RH}_\infty$ and satisfy the Bezout identity:

$$\begin{bmatrix} \tilde{M}(s) & \tilde{N}(s) \end{bmatrix} \begin{bmatrix} X_l(s) \\ Y_l(s) \end{bmatrix} = I, \quad \text{or}$$

$$\tilde{M}(s)X_I(s) + \tilde{N}(s)Y_I(s) = I,$$

for some $X_I(s), Y_I(s) \in \mathcal{RH}_\infty$.

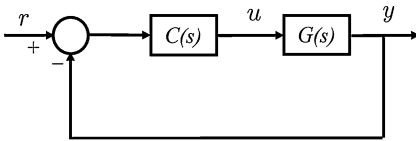
A linear time-invariant (LTI) feedback control system is shown in Fig. 1, where $G(s) = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = D + C(sI - A)^{-1}B$ is a nominal plant model and $C(s)$ is a controller designed such that the closed-loop system is stable.

Let $C(s) = \tilde{V}^{-1}(s)\tilde{U}(s)$ and $G(s) = \tilde{M}^{-1}(s)\tilde{N}(s)$ be the (left) coprime factorizations of $C(s)$ and $G(s)$. The Youla-Kučera parameterization of all stabilizing controllers for $G(s)$ can then be expressed as

$$K(s) = (\tilde{V}(s) - Q(s)\tilde{N}(s))^{-1}(\tilde{U}(s) + Q(s)\tilde{M}(s)),$$

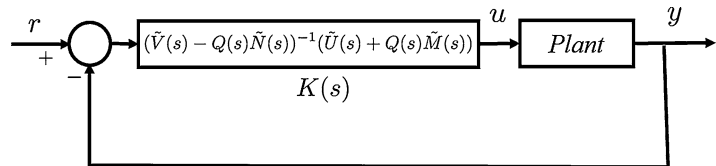
where $Q(s) \in \mathcal{RH}_\infty$ and $\det(\tilde{V}(\infty) - Q(\infty)\tilde{N}(\infty)) \neq 0$ and, apparently, $C(s)$ is a special case of $K(s)$ when $Q(s) = 0$. Note that, equivalently, $K(s)$ can be implemented in a conventional “single-block” controller structure shown in Fig. 2 or, as proposed in Zhou and Ren (2001), in a “multi-block” controller structure featuring a residual signal-driven Q -Regulation, as shown in Fig. 3.

While $Q(s)$ can be arbitrarily chosen in $\mathcal{RH}_\infty$ and independent on $C(s)$, one can see that $Q(s)$ is actually integrated with $C(s)$ to form a single-controller $K(s)$ in Fig. 2, but $Q(s)$ in Fig. 3 essentially brings in another freedom of design



Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ Control Designs, Fig. 1 A feedback control system

Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ Control Designs, Fig. 2 Conventional “single-block” controller structure

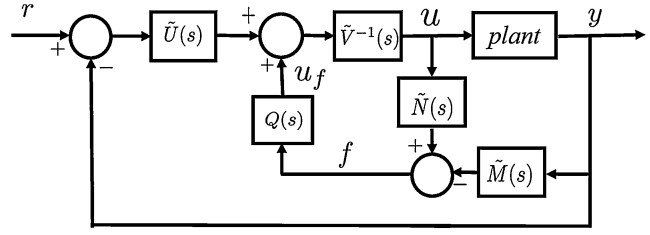
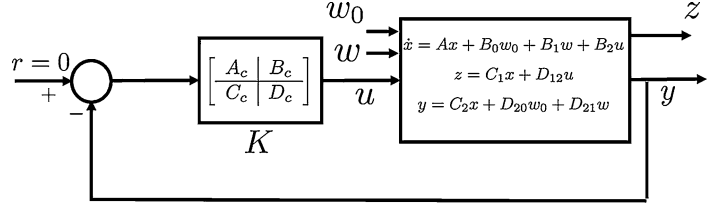


other than $C(s)$ as it is driven by an explicit residue signal $f = \tilde{N}(s)u - \tilde{M}(s)y$ and generates an extra regulating signal u_f . This residual signal f can be calculated as exactly the difference between the sensor output of the real plant and the estimated sensor output of a standard Luenberger observer of the plant, hence, providing a reasonable measure of the modeling error in practice (Chen et al. 2019; Zhou and Ren 2001).

It is easy to see that, with the Youla-Kučera parameterization of all stabilizing controllers, any multi-objective optimal controller for specified performances, if existing, should correspond to an appropriately selected $Q(s)$ since the optimal controller must first be a stabilizing controller. Therefore, the “single-block” and “multi-block” implementations actually point to the different frameworks of multi-objective optimal control designs.

$\mathcal{H}_\infty$ Gaussian Control

It is clear that a multi-objective optimal control design can be conducted within the “centralized” structure depicted in Fig. 2. However, even if Youla-Kučera structure provides excellent insights of the design of optimal controllers, it, unfortunately, does not offer easy solutions in transfer functions to various optimization problems. Practically, instead of directly searching for an optimal $Q(s)$, the design of optimal controllers can be alternatively done in the state space with the same “centralized” controller structure. For example, a popular multi-objective optimal control problem known as the “mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control” (Bernstein and Haddad 1989; Chen and Zhou 2001; Khargonekar and Rotea 1991; Zhou et al. 1994) can be formulated in Fig. 4, or, equivalently, in Fig. 5 as represented in the linear fractional transformation (LFT) format, where (A_c, B_c, C_c, D_c) is the controller realization in

Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ **Control Designs,****Fig. 3** “Multi-block”
controller structure
featuring Q -regulation**Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$** **Control Designs, Fig. 4**Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control
design in state space

the state space, w represents the modeling uncertainty disturbance, and w_0 is a (vector) white noise. It is well-known that $\mathcal{H}_\infty$ and $\mathcal{H}_2$ performances of a control system are normally applied to address the impacts of w and w_0 , respectively, characterized as the robust and optimal performances of the control system. In the remaining part of this section, the “ $\mathcal{H}_\infty$ Gaussian Control” design (Chen and Zhou 2001) is presented as a sample study, which is essentially a mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control design utilizing the “centralized” implementation of Youla-Kučera parameterization.

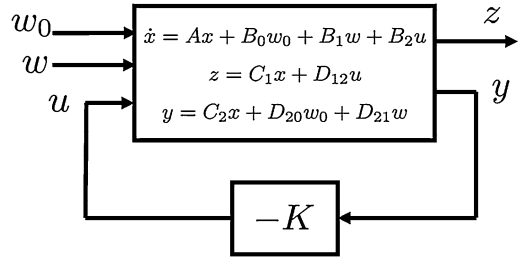
Let us introduce first the set of all bounded power signals and the power norm defined in this set: Given a real vector stochastic signal $u(t)$:

$$u(t) = [u_1(t) \ u_2(t) \ \cdots \ u_m(t)]^T \in \mathbb{R}^m, \quad \forall t \in \mathbb{R},$$

where $u_i(t)$, $i = 1, \dots, m$ are real stationary random processes, define the **mean** and averaged **auto-correlation matrices** of $u(t)$, respectively, as follows:

$$E\{u\} := [E\{u_1(t)\} \ E\{u_2(t)\} \ \cdots \ E\{u_m(t)\}]^T,$$

$$R_{uu}(\tau) := \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T E\{u(t+\tau)u^T(t)\}dt.$$

**Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ Control Designs, Fig. 5**Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control design in LFT expression

The Fourier transform of $R_{uu}(\tau)$, if exists, is $S_{uu} := \frac{1}{2\pi} \int_{-\infty}^{\infty} R_{uu}(\tau)e^{-j\omega\tau}d\tau$. The so-called bounded power signal is defined as follows:

Definition 2 A vector stationary stochastic signal u is said to have bounded power if

1. both R_{uu} and S_{uu} exist;
2. $\lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T E\|u(t)\|^2 dt < \infty$.

Let $\mathcal{P}$ be the space of all signals with bounded power. A semi-norm can be defined on $\mathcal{P}$

$$\|u\|_{\mathcal{P}} := \sqrt{\lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T E\{\|u(t)\|^2\}dt} \\ = \sqrt{\text{trace}[R_{uu}(0)]}, \quad \forall u \in \mathcal{P}.$$

In this entry, the well-known Gaussian white noise $w_0(t)$ is taken as a zero-mean stationary process with an identity power spectrum, that is, $E\{w_0(t)\} \equiv 0$ and $E\{w_0(t)w_0^T(\tau)\} = I\delta(t - \tau)$. Independence between stochastic signals is defined below.

Definition 3 Two vector stationary stochastic signals $w_1(t)$ and $w_2(t)$ are said to be independent if, for any $t_1 \geq 0$ and $t_2 \geq 0$, we have

$$E\{w_1(t_1)w_2^T(t_2)\} = E\{w_1(t_1)\}E\{w_2^T(t_2)\}.$$

Now consider an LTI system described by

$$\dot{x} = Ax + B_0w_0 + B_1w + B_2u, \quad x(0) = 0, \quad (1)$$

$$z = C_1x + D_{12}u, \quad R_1 := D_{12}^T D_{12} > 0, \quad (2)$$

$$y = C_2x + D_{20}w_0, \quad R_0 := D_{20}D_{20}^T > 0, \quad (3)$$

where w is a bounded power signal, w_0 is a white noise signal, w and w_0 are independent. The $\mathcal{H}_\infty$ Gaussian control can be formulated as in Fig. 6 with the control law taking the observer-based form, consisting of a state feedback gain F and an observer gain L :

$$\dot{\hat{x}} = \hat{A}\hat{x} + B_2u - Ly, \quad \hat{x}(0) = 0, \quad (4)$$

$$u = F\hat{x}. \quad (5)$$

Let $e_x := x - \hat{x}$ and define the following performance indices with a given $\gamma > 0$:

$$J_1(u, w, w_0) := \gamma^2 \|w\|_{\mathcal{D}}^2 - \|z\|_{\mathcal{D}}^2,$$

$$J_2(u, w, w_0) := \|e_x\|_{\mathcal{D}}^2.$$

Then the $\mathcal{H}_\infty$ Gaussian control design is to: find a feedback control law u_* in the given form (4)–(5) such that

$$J_1(u_*, w_*, w_0) \leq J_1(u_*, w, w_0), \quad \forall w \in \mathcal{P}_s;$$

$$J_2(u_*, w_*, w_0) \leq J_2(u, w_*, w_0),$$

where w_* is called the worst disturbance signal.

A theorem is stated for this $\mathcal{H}_\infty$ Gaussian control problem and concludes this section. To simplify the notations, the following abbreviations are introduced:

$$A_x := A - B_2R_1^{-1}D_{12}^T C_1,$$

$$A_y := A - B_0D_{20}^T R_0^{-1} C_2,$$

$$P := B_0(I - D_{20}^T R_0^{-1} D_{20})B_0^T,$$

$$Q := C_1^T(I - D_{12}R_1^{-1}D_{12}^T)C_1.$$

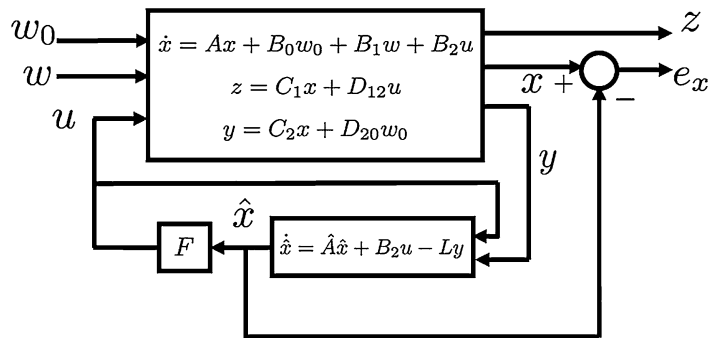
Theorem 1 Let the dynamical system G described by Eqs. (1), (2), and (3) satisfy the following assumptions:

(A1) (A, B_2) is stabilizable and (C_2, A) is detectable.

(A2) $\begin{bmatrix} A - j\omega I & B_2 \\ C_1 & D_{12} \end{bmatrix}$ has full column rank for all ω .

Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ Control Designs, Fig. 6

$\mathcal{H}_\infty$ Gaussian control design



(A3) $\begin{bmatrix} A - j\omega I & B_0 \\ C_2 & D_{20} \end{bmatrix}$ has full row rank for all ω .

If there are stabilizing solutions $P_1 \geq 0$, $P_2 \geq 0$ and $P_3 \geq 0$ solving the following Riccati equations:

$$\begin{aligned} A_x^T P_1 + P_1 A_x + P_1 (B_1 B_1^T / \gamma^2 - B_2 R_1^{-1} B_2^T) P_1 + Q &= 0, \\ P_2 (A_y + \gamma^{-2} B_1 B_1^T P_1 - P_3 C_2^T R_0^{-1} C_2) + (A_y + \gamma^{-2} B_1 B_1^T P_1 - P_3 C_2^T R_0^{-1} C_2)^T P_2 \\ + \gamma^{-2} P_2 B_1 B_1^T P_2 + (D_{12}^T C_1 + B_2^T P_1)^T R_1^{-1} (D_{12}^T C_1 + B_2^T P_1) &= 0, \\ [A_y + \gamma^{-2} B_1 B_1^T (P_1 + P_2)] P_3 + P_3 [A_y + \gamma^{-2} B_1 B_1^T (P_1 + P_2)]^T - P_3 C_2^T R_0^{-1} C_2 P_3 + P &= 0, \end{aligned}$$

i.e., $A_x + (B_1 B_1^T / \gamma^2 - B_2 R_1^{-1} B_2^T) P_1$, and $A_y + \gamma^{-2} B_1 B_1^T (P_1 + P_2) - P_3 C_2^T R_0^{-1} C_2$ are both stable, then there exist the optimal control law u_* and the worst disturbance signal w_* such that:

$$J_1(u_*, w_*, w_0) \leq J_1(u_*, w, w_0),$$

$$J_2(u_*, w_*, w_0) \leq J_2(u, w_*, w_0).$$

If the solutions exist, then $w_* = \gamma^{-2} B_1^T (P_1 x + P_2 e)$ and the optimal controller is given by:

$$\dot{\hat{x}} = \hat{A} \hat{x} + B_2 u_* - L_* y,$$

$$u_* = F_* \hat{x}, \quad \hat{x}(0) = 0,$$

where $\hat{A} = A + \gamma^{-2} B_1 B_1^T P_1 + L_* C_2$, $F_* = -R_1^{-1} (D_{12}^T C_1 + B_2^T P_1)$, and $L_* = -(B_0 D_{20}^T + P_3 C_2^T) R_0^{-1}$.

Conversely, let $P_1 \geq 0$ be a stabilizing solution to

$$\begin{aligned} A_x^T P_1 + P_1 A_x + P_1 (B_1 B_1^T / \gamma^2 \\ - B_2 R_1^{-1} B_2^T) P_1 + Q &= 0. \end{aligned}$$

Suppose there exist a w'_* and a controller u_* (hence, a L_*)

$$\begin{aligned} \dot{\hat{x}} &= (A + \gamma^{-2} B_1 B_1^T P_1 + L_* C_2 + B_2 F_*) \hat{x} \\ &\quad - L_* y, \end{aligned}$$

$$u_* = F_* \hat{x}, \quad F_* = -R_1^{-1} (D_{12}^T C_1 + B_2^T P_1),$$

achieving $0 < J_1(u_*, w'_*, 0) \leq J_1(u_*, w, 0)$. Then there exists a w_* achieving $J_1(u_*, w_*, w_0) \leq J_1(u_*, w, w_0)$. If, furthermore, this w_* also achieves $J_2(u_*, w_*, w_0) \leq J_2(u, w_*, w_0)$, then there exist $P_2 \geq 0$ and $P_3 \geq 0$ solving:

$$\begin{aligned} P_2 (A_y + \gamma^{-2} B_1 B_1^T P_1 - P_3 C_2^T R_0^{-1} C_2) + (A_y + \gamma^{-2} B_1 B_1^T P_1 - P_3 C_2^T R_0^{-1} C_2)^T P_2 \\ + \gamma^{-2} P_2 B_1 B_1^T P_2 + (D_{12}^T C_1 + B_2^T P_1)^T R_1^{-1} (D_{12}^T C_1 + B_2^T P_1) &= 0, \\ [A_y + \gamma^{-2} B_1 B_1^T (P_1 + P_2)] P_3 + P_3 [A_y + \gamma^{-2} B_1 B_1^T (P_1 + P_2)]^T - P_3 C_2^T R_0^{-1} C_2 P_3 + P &= 0. \end{aligned}$$

Moreover, if $A + \gamma^{-2} B_1 B_1^T (P_1 + P_2) - (B_0 D_{20}^T + P_3 C_2^T) R_0^{-1} C_2$ is stable, then $L_* = -(B_0 D_{20}^T + P_3 C_2^T) R_0^{-1}$.

The trade-off performance can be clearly observed as the controller becomes an LQG control (hence, an $\mathcal{H}_2$ control too) exactly when $\gamma \rightarrow \infty$, and, if there is no white noise posed in the model, the controller returns back to the

$\mathcal{H}_\infty$ design. Otherwise, the controller delivers performance that is in between $\mathcal{H}_\infty$ and $\mathcal{H}_2$, hence, a mixed one.

LQG Control with $\tilde{Q}$ -Regulation

It is not difficult to figure out the following two facts related to the “multi-block” with

Q -regulation implementation of the Youla-Kučera parameterization in Fig. 3:

- Figure 3 can be converted into the configuration in Fig. 7:

where $Q(s) = \tilde{V}(s)\tilde{Q}(s) \in \mathcal{RH}_\infty$ is an admissible Youla-Kučera parameter in Fig. 3 if $\tilde{Q}(s)$ in Fig. 7 is designed to be a proper stable transfer function matrix.

- The structure posed by $\tilde{N}(s)$ and $\tilde{M}(s)$ in Figs. 3 and 7 actually mimics exactly a standard Luenberger observer (Chen et al. 2019; Zhou and Ren 2001) and $f = \tilde{N}u - \tilde{M}y = \hat{y} - y$, $\hat{y} = C_2\hat{x}$, as shown in Fig. 8

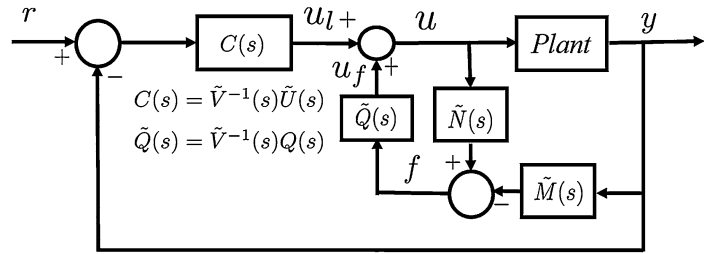
These two insights immediately motivate us to consider the new controller structure in Fig. 9 for the multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ control problem originally posed in Fig. 4

As pointed out in Zhou and Ren (2001), the controller (A_c, B_c, C_c, D_c) and the $\tilde{Q}$ -regulation

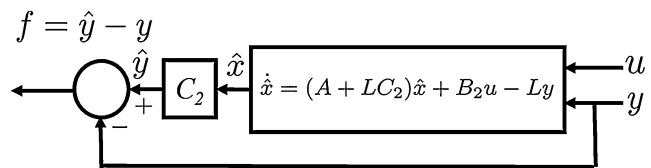
can be designed independently. Considering that $\tilde{Q}$ is driven by the residual signal $f = \hat{y} - y$, the structure in Fig. 9 brings in the hope that the conservativeness rooted in the “single-block” controller design could be meaningfully reduced, if not completely removed. $C(s)$ with extra $\tilde{Q}$ -regulation design, hence, can be deemed as a new mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control design although it still originates from the Youla-Kučera parameterization. Furthermore, it is noted that, in Fig. 9 the LQG controller is in Luenberger observer-based structure, characterized by an observer gain L and a state feedback gain F , and driven by the same sensor output y . Therefore it is then reasonable to take the advantage of the controller structure with the shared observer as shown in Fig. 10

In this section, a sample study originated from (Chen et al. 2019) is presented, by taking (A_c, B_c, C_c, D_c) as an LQG controller and designing $\tilde{Q}$ -regulation in the $\mathcal{H}_\infty$ sense.

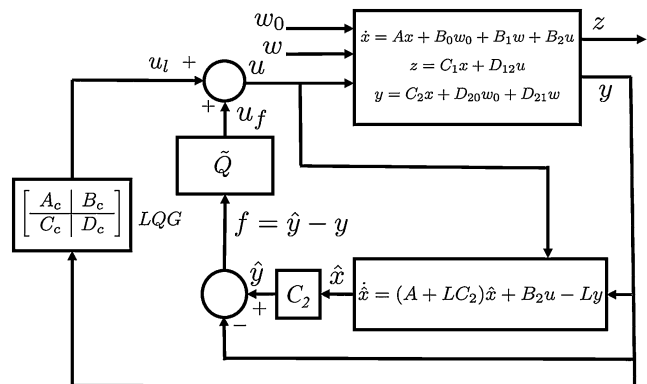
Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ Control Designs, Fig. 7
Converted “multi-block” controller structure with $\tilde{Q}$ -regulation



Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ Control Designs, Fig. 8
State-space interpretation of $\tilde{N}(s)$ and $\tilde{M}(s)$

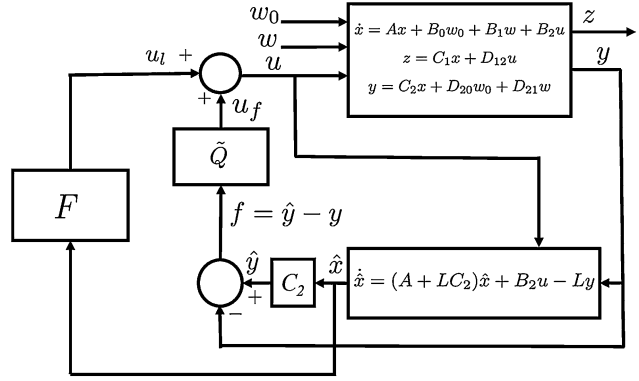


Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ Control Designs, Fig. 9
Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ control in “multi-block” design



Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ Control Designs, Fig. 10

Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ control with shared observer



The following model is a slightly extended with case of that in (1), (2), and (3) with the same interpretation of signals:

$$\dot{x} = Ax + B_1w + B_0w_0 + B_2u, \quad (6)$$

$$z = C_1x + D_{12}u, \quad (7)$$

$$y = C_2x + D_{21}w + D_{20}w_0, \quad (8)$$

Extra (but standard) assumptions are made for the system in (6), (7), and (8) (Doyle et al. 1989), other than the Assumptions (A1)–(A3):

(A4) (A, B_1) is stabilizable and (C_1, A) is detectable.

(A5) $\begin{bmatrix} A - j\omega I & B_1 \\ C_2 & D_{21} \end{bmatrix}$ has full row rank for all ω .

(A6) $R_1 = D_{12}^T D_{12} > 0$, $R_2 = D_{21} D_{21}^T > 0$, and $R_0 = D_{20} D_{20}^T > 0$.

Let F and L be the state feedback, and the observer gains designed for an LQG control of the nominal system ($w = 0$) in (6), (7), and (8), that is, $F = -R_1^{-1}(D_{12}^T C_1 + B_2^T P_{10})$, $L = -(B_0 D_{20}^T + P_{20} C_2^T) R_0^{-1}$, where $P_{10} \geq 0$ and $P_{20} \geq 0$ solve

$$P_{10}A_{10} + A_{10}^T P_{10} - P_{10}B_2R_1^{-1}B_2^T P_{10} + Q_{10} = 0, \quad (9)$$

$$P_{20}A_{20}^T + A_{20}P_{20} - P_{20}C_2^T R_0^{-1}C_2 P_{20} + Q_0 = 0, \quad (10)$$

$$A_{10} = A - B_2R_1^{-1}D_{12}^T C_1,$$

$$A_{20} = A - B_0D_{20}^T R_0^{-1}C_2,$$

$$Q_{10} = C_1^T (I - D_{12}R_1^{-1}D_{12}^T)C_1,$$

$$Q_0 = B_0(I - D_{20}^T R_0^{-1}D_{20})B_0^T.$$

Apparently, the system with the implemented LQG controller carries the model as:

$$\dot{x} = Ax + B_1w + B_2u_l + B_2u_f, \quad (11)$$

$$\dot{\hat{x}} = (A + LC_2)\hat{x} + B_2u_l + B_2u_f - Ly, \quad (12)$$

$$z = C_1x + D_{12}u_l + D_{12}u_f, \quad (13)$$

$$y = C_2x + D_{21}w, \quad (14)$$

$$u_l = F\hat{x}, \quad u_f = \tilde{Q}(f), \quad f = C_2\hat{x} - y. \quad (15)$$

Since the LQG control is not designed to address the uncertainty disturbance w , the additional regulating signal $u_f = \tilde{Q}(f)$, hence, should be designed to bring back the robust performance which naturally falls into the category of $\mathcal{H}_\infty$ control. Therefore, the problem addressed in this section can be formulated as follows:

For the augmented system (11), (12), (13), (14), and (15) under the LQG control u_l characterized with F and L , as shown in Fig. 10, we need to find a regulating controller $\tilde{Q}$ such that $u_f = \tilde{Q}(f)$ achieves $\|T_{zw}(s)\|_\infty < \gamma$, where γ is a prescribed value, and $T_{zw}(s)$ is the closed-loop transfer function from w to z .

Note that the system model (11), (12), (13), (14), and (15) can be equivalently converted into the following augmented format, after implementing F and L and introducing $e = x - \hat{x}$ as the estimation error variable for the LQG control:

$$\begin{aligned}\dot{\hat{x}} &= (A + B_2 F)x - B_2 F e + B_1 w + B_2 u_f, \\ \dot{e} &= (A + LC_2)e + (B_1 + LD_{21})w, \\ z &= (C_1 + D_{12}F)x - D_{12}Fe + D_{12}u_f, \\ f &= -C_2 e - D_{21}w, \\ u_f &= \tilde{Q}(f).\end{aligned}$$

Define:

$$\bar{x} = \begin{bmatrix} x \\ e \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} A + B_2 F & -B_2 F \\ 0 & A + LC_2 \end{bmatrix},$$

$$\bar{D}_{12} = D_{12},$$

$$\bar{B}_1 = \begin{bmatrix} B_1 \\ B_1 + LD_{21} \end{bmatrix}, \quad \bar{B}_2 = \begin{bmatrix} B_2 \\ 0 \end{bmatrix},$$

$$\bar{D}_{21} = -D_{21},$$

$$\bar{C}_1 = [C_1 + D_{12}F - D_{12}F], \quad \bar{C}_2 = [0 - C_2].$$

Then the model can be finally rewritten as:

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}_1 w + \bar{B}_2 u_f, \quad (16)$$

$$z = \bar{C}_1 \bar{x} + D_{12} u_f, \quad (17)$$

$$f = \bar{C}_2 \bar{x} + \bar{D}_{21} w, \quad (18)$$

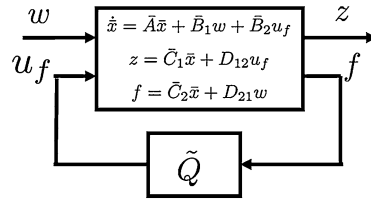
$$u_f = \tilde{Q}(f). \quad (19)$$

This control problem can be, again, expressed in the LFT format, as shown in Fig. 11.

Now we can present the following Theorem (Chen et al. 2019):

Theorem 2 For the system in (16), (17), (18), and (19) as shown in Fig. 11, given a $\gamma > 0$, there exists an admissible regulating controller $\tilde{Q}$ such that $\|T_{zw}(s)\|_\infty < \gamma$ if and only if the following conditions are satisfied:

1. There exist $P_1 \geq 0$ and $P_2 \geq 0$ solve the following two Riccati equations:



Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ Control Designs, Fig. 11
Control of augmented system

$$\begin{aligned}P_1 A_{10} + A_{10}^T P_1 \\ + P_1 \left(\frac{B_1 B_1^T}{\gamma^2} - B_2 R_1^{-1} B_2^T \right) P_1 + Q_{10} = 0, \quad (20)\end{aligned}$$

$$\begin{aligned}P_2 A_2^T + A_2 P_2 \\ + P_2 \left(\frac{C_1^T C_1}{\gamma^2} - C_2^T R_2^{-1} C_2 \right) P_2 + Q_2 = 0, \quad (21)\end{aligned}$$

where

$$A_2 = A - B_1 D_{21}^T R_2^{-1} C_2,$$

$$Q_2 = B_1 (I - D_{21}^T R_2^{-1} D_{21}) B_1^T.$$

2. $\rho(P_1 P_2) < \gamma^2$.

Moreover, one such admissible controller, if exists, can be obtained as:

$$\dot{x}_q = A_q x_q + B_q f,$$

$$u_f = F_* x_q,$$

where

$$\begin{aligned}A_q &= \bar{A} + \gamma^{-2} \bar{B}_1 \bar{B}_1^T \bar{P}_1 + \bar{B}_2 F_* \\ &\quad - B_q (\bar{C}_2 - \gamma^{-2} D_{21} \bar{B}_1^T \bar{P}_1),\end{aligned}$$

$$B_q = -(I - \gamma^{-2} \bar{P}_2 \bar{P}_1)^{-1} L_*,$$

$$\bar{P}_1 = \begin{bmatrix} P_1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \bar{P}_2 = \begin{bmatrix} P_2 & P_2 \\ P_2 & P_2 \end{bmatrix},$$

$$F_* = -[F + R_1^{-1} (B_2^T P_1 + D_{12}^T C_1) - F],$$

$$L_* = \begin{bmatrix} (P_2 C_2^T + B_1 D_{21}^T) R_2^{-1} \\ (P_2 C_2^T + B_1 D_{21}^T) R_2^{-1} + L \end{bmatrix}.$$

Remark 1 Based on Theorem 2, the new multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ control design procedures can be summarized as follows:

1. Solving P_{10} and P_{20} from (9) and (10) and finding $F = -R_1^{-1}(D_{12}^T C_1 + B_2^T P_{10})$, $L = -(B_0 D_{20}^T + P_{20} C_2^T) R_0^{-1}$, this is the LQG control design.
2. Solving P_1 and P_2 from (20) and (21) and finding $F_* = -[F + R_1^{-1}(B_2^T P_1 + D_{12}^T C_1) - F]$, $L_* = \begin{bmatrix} (P_2 C_2^T + B_1 D_{21}^T) R_2^{-1} \\ (P_2 C_2^T + B_1 D_{21}^T) R_2^{-1} + L \end{bmatrix}$, this is the $\mathcal{H}_\infty$ control design.
3. The multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ controller can be constructed as:

$$\begin{aligned}\dot{x}_q &= A_q x_q + B_q f, \\ u_f &= F_* x_q,\end{aligned}$$

where

$$\begin{aligned}A_q &= \bar{A} + \gamma^{-2} \bar{B}_1 \bar{B}_1^T \bar{P}_1 + \bar{B}_2 F_* \\ &\quad - B_q (\bar{C}_2 - \gamma^{-2} D_{21} \bar{B}_1^T \bar{P}_1), \\ B_q &= -(I - \gamma^{-2} \bar{P}_2 \bar{P}_1)^{-1} L_*, \\ \bar{P}_1 &= \begin{bmatrix} P_1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \bar{P}_2 = \begin{bmatrix} P_2 & P_2 \\ P_2 & P_2 \end{bmatrix}.\end{aligned}$$

Summary and Future Directions

While both the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ design with single controller and the LQG control with extra $\tilde{Q}$ -regulation address multi-objective performance in $\mathcal{H}_2$ and $\mathcal{H}_\infty$ senses, the fundamental difference between these two approaches lies in the fact that the design goal of the former is to achieve the (best) trade-off of conflicting performances, while that of the latter is to provide appropriate complements within conflicting performance by taking the advantage of the extra (and independent) regulating control signal. Although the traditional multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ design may

enjoy the benefit of low controller order, it is, however, at the expense of conservative control performance. For example, in the $\mathcal{H}_\infty$ Gaussian control design, γ actually can be used as the tuning parameter as the weight between the robustness and the optimality, noting the fact that, by taking $\gamma \rightarrow \infty$, the solution will be reduced to the LQG control, hence, achieving the optimal performance but the least robustness. On the other hand, the lower the γ value, the closer the solution to the $\mathcal{H}_\infty$ control, meaning a better robustness but more compromised optimal performance. In the case of the LQG with $\tilde{Q}$ -regulation design, it is obvious that the design of LQG control is independent of the design of $\tilde{Q}$ and that the regulating action of $\tilde{Q}$ is “proportional” to the residual signal f which, in some sense, indicates the modeling mismatch. Hence, it is not a surprise that the new controller structure could provide a robust regulation with much less compromised optimal performance achieved by LQG control, albeit at the expense of higher controller order overall.

It is noted that $\tilde{Q}$ could be designed through other approaches in practice, for example, as a predictive controller, to further improve the robustness of the LQG performance for the closed-loop system. Essentially, the role played by the observer in $\tilde{Q}$ structure can be deemed as serving certain kind of “model-guided” function, hence, leaving the window open for the use of more complicated model or even a nonlinear model in the case of a nonlinear plant being addressed.

The structure of “multi-block” implementation of Youla-Kučera parameterization in Fig. 3 could potentially be extended to other control designs than LQG and $\mathcal{H}_\infty$, thanks to the complementary nature of the controller $C(s)$ and the $\tilde{Q}$ -regulation, as well as the extra design freedom brought in by $\tilde{Q}$. It would then be interesting, both theoretically and practically, to explore different combined designs of $\tilde{Q}$ and the controller $C(s)$ to illustrate the effectiveness of this complementary paradigm.

Cross-References

- [H-Infinity Control](#)
- [H₂ Optimal Control](#)
- [Linear Quadratic Optimal Control](#)
- [Optimization-Based Robust Control](#)
- [Robust Synthesis and Robustness Analysis Techniques and Tools](#)
- [Robust \$\mathcal{H}_2\$ Performance in Feedback Control](#)

Bibliography

- Bernstein DS, Haddad WM (1989) LQG control with an $\mathcal{H}_\infty$ performance bound: a Riccati equation approach. *IEEE Trans Autom Control* AC-34: 293–305
- Chen X, Zhou K (2001) Multiobjective $\mathcal{H}_2/\mathcal{H}_\infty$ control design. *SIAM J Control Optim* 40(2): 628–660
- Chen X, Zhou K, Tan Y (2019) Revisit of LQG control—a new paradigm with recovered robustness. In: *Proceedings of 58th IEEE conference on decision and control*, Nice, Dec 2019, pp 5819–5825
- Doyle JC, Glover K, Khargonekar PP, Francis BA (1989) State-space solutions to standard $\mathcal{H}_\infty$ and $\mathcal{H}_2$ control problems. *IEEE Trans Autom Control* AC-34(8):831–847
- Khargonekar PP, Rotea MA (1991) Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control: a convex optimization approach. *IEEE Trans Autom Control* AC-36(7):824–837
- Kučera V (1979) *Discrete linear control: the polynomial equation approach*. Wiley, New York
- Youla D, Bongiorno J, Jabr H (1976a) Modern Wiener-Hopf design of optimal controllers, part I: the single input case. *IEEE Trans Autom Control* 21:3–148
- Youla D, Jabr H, Bongiorno J (1976b) Modern Wiener-Hopf design of optimal controllers, part II: the multivariable case. *IEEE Trans Autom Control* 21: 319–338
- Zhou K, Ren Z (2001) A new controller architecture for high performance, robust, and fault-tolerant control. *IEEE Trans Autom Control* 46(10):1613–1618
- Zhou K, Glover K, Bodenheimer B, Doyle J (1994) Mixed $\mathcal{H}_\infty$ and $\mathcal{H}_2$ performance objectives I: robust performance analysis. *IEEE Trans Autom Control* 39(8):1564–1574
- Zhou K, Doyle JC, Glover K (1996) *Robust and optimal control*. Prentice Hall, Upper Saddle River

Multiple Mobile Robots

Filippo Arrichiello

Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Cassino, Italy

Abstract

This section focuses on multiple mobile robots, i.e., team of grounded, aerial, or marine autonomous mobile robots performing cooperative missions. To the purpose, the section provides a brief literature review and an overview about some of the main aspects related to the control of multiple mobile robots, as their possible aims, applications, and control architectures.

Keywords

Multi-robot systems · Control architecture · Robotic swarms · Multi-agent systems

Introduction

In the recent years, several research efforts have been directed toward the use of multiple mobile robots (MMRs). The interest in this field is well justified by the advantages that such systems present with respect to single autonomous robots, and it is well supported by the improvements in the technologies that allow the interaction and the integration among multiple robotic systems. Generally speaking, the term MMRs includes different typologies of robotic systems; in this context, the term will be used referring to a team of cooperating grounded, aerial, or marine autonomous mobile robots.

One of the main motivations for employing MMRs can be found in the possibility they offer in terms of improvement of the system effectiveness, that is, with respect to a single autonomous robot or to a team of not cooperating robots, a

MMRs system can better perform a mission in terms of time and quality, it can achieve tasks not executable with a single robot (e.g., moving a large object), or it can take advantages of distributed sensing and actuation. Moreover, instead of building and using a single powerful robot, a multi-robot solution can be easier and cheaper, it can provide flexibility to the execution of the task, and it can make the system tolerant to possible robots' faults.

The term *cooperation* underlines the interaction or the integration among multiple mobile robots; that means that the robots have to communicate, exchange information, or interact in some way with each other in order to achieve an overall common objective. The meaning of cooperation among robots has been widely discussed in the scientific community, and different definitions have been proposed. Three of them, reported in Cao et al. (1997), are: cooperation is

- “a joint collaborative behavior that is directed toward some goal in which there is a common interest or reward”
- “a form of interaction, usually based on communication”
- “joining together for doing something that creates a progressive result such as increasing performance or saving time”.

The first definition leads to the study of *task decomposition*, *task allocation*, and other *distributed artificial intelligence* issues; the second underlines the requirements of *communication* or other common resources; finally, the third is related to the *performance measurements* of *cooperation*, such as speedup in time to complete a task. Moreover, these definitions point out different aspects of cooperation: the task, the mechanism of cooperation, and the system performance.

The *task* can be considered as the aim of the multi-robot system; thus, it changes depending on the different applications and on the typologies of the MMRs system. The task is usually decomposed in elementary sub-tasks (task decomposition) to make it easier to understand and execute.

These sub-tasks can be distributed among multiple resources (task allocation), while the overall behavior of the system depends on how these sub-tasks are recombined to obtain the final action or motion command for the system.

The *mechanism of cooperation* represents the logic that originates the cooperation, and it may depend on the control architectures and strategies, on aspects of the tasks specification, or on the interaction dynamics. The MMRs system has to exhibit a collective behavior, or a set of actions that accomplishes the same behavior that was required for the single more complex robot; thus, the members of the MMRs system have to communicate directly through an explicit communication channel or indirectly through one robot sensing the others.

The *system performance* can be represented through characteristics, e.g., mission execution time, computational complexity, robustness, and fault tolerance, and it may depend on the global structure of the system, e.g., typologies of the system, control architectures and strategies, task definition and actuation, and communication characteristics.

Due to the multiple aspects that characterize a MMRs system, the progress of the research in cooperative robotics has concerned most of these aspects simultaneously. Thus, characteristics like control architectures and algorithms, differentiation among the robots of the team, communication structures, resource conflicts, and mechanisms of cooperation or learning have been often developed by the scientific community in an integrated way. This strong correlation makes hard a classification focused on single aspects; thus, different works have been presented only to introduce a taxonomy valid for the field. The work in Dudek et al. (1996) presents a taxonomy for multi-agent robotic systems that proposes a classification depending on the size of the team (how many robots compose the team), the communication parameters (communication range, topology, and bandwidth), the reconfigurability of the team, the processing ability of each member, and the composition of the team (homogeneous vs. heterogeneous robots). The work in Farinelli

et al. (2004) presents a classification based on different levels of coordination (unaware, aware but non-coordinated, weakly coordinated, strongly coordinated systems), and it introduces a classification based on the so-called coordination dimensions (cooperation, knowledge, coordination, organization) and system dimensions (communication, team composition, system architectures, and team size). Finally, the work in Gerkey and Mataric (2004) presents a taxonomy based on coordination mechanisms and on multi-robot task allocation.

Aims and Applications

The large variety of MMRs developed in the recent years makes possible a distinction based on the typologies of vehicles and on their missions. The first applications concerned systems composed of grounded mobile robots (e.g., see Fig. 1) that are autonomous robots with wheels or crawlers. Such robots can be built both for indoor environments (i.e., buildings, museums, and laboratories) (Noreils 1993; Howard et al. 2006) and for outdoor environments (i.e., open space with terrain, grass, or rocky floor) (Huntsberger et al. 2003; Carpin and Parker 2002), and they result the most commonly used ones.

However, in the latest years, several applications with different kinds of vehicles have been proposed in the literature, as multi-robot systems composed of unmanned aerial vehicles (AUVs) (Beard et al. 2001; Stipanovic et al.

2004; Kushleyev et al. 2013; Spurný et al. 2019), underwater autonomous vehicles (AUVs) (Fiorelli et al. 2004; Stilwell and Bishop 2000; Abreu et al. 2016), or fleets of marine crafts (Encarnacao and Pascoal 2001; Ihle et al. 2006; Børhaug et al. 2010) (see Figs. 2 and 3).

The applications of multi-robot systems may involve different fields, e.g., industrial, military, service robotics, and research and study of biological systems, and they may concern largely different kinds of missions as exploration, box pushing, military operation, navigation in unstructured environment, traffic control, entertainment, simulations of biological systems, and environmental monitoring.

Among the possible applications, a typical one concerns the possibility to move large objects using multiple robots (sometimes with manipulators on board). A single robot, in fact, may be not powerful enough to push alone the



Multiple Mobile Robots, Fig. 2 Multiple aerial mobile robots. Courtesy of prof. Vijay Kumar from UPenn



Multiple Mobile Robots, Fig. 1 Multiple grounded mobile robots



Multiple Mobile Robots, Fig. 3 Multiple underwater mobile robots. (Courtesy of Graal Tech)

object, and it can be unable to apply forces in all the generalized directions. Thus, a multi-robot solution can be useful to share the needed power among multiple robots, and, using multiple contact points, it may permit to apply forces in the all generalized directions. On the other hand, the multi-robot solution needs for a strong coordination among the robots; that means the robots have to share information about their relative positions and about the shape and position of the object. To achieve this kind of mission (generally called box-pushing mission), different MMRs systems have been presented in the literature, e.g., in Mataric et al. (1995) where some experimental results of box pushing using two-legged robots are presented or in Yamada and Saito (2001) where a team of differential-drive mobile robots performs the box-pushing mission without explicit communication. The work in Wang et al. (2003) focuses on function distribution and behavior design for cooperative object handling with multiple mobile robots; moreover, the work in Yamashita et al. (2003) presents a motion-planning method for multiple-mobile robots to cooperatively transport large objects, while the work in Tanner et al. (2003) presents an approach to carry a deformable object by means of two mobile robots with manipulators on board.

Another common kind of applications concerns the possibility to use a team of cooperating robots to explore an unknown environment and build its map. To achieve this mission in a cooperative way, the robots must be coordinated to explore different parts of the environment and share partial maps elaborated by each single robot to build the global map of the environment; as a consequence, they can explore the environment in reduced amount of time with respect to a single robot. Simultaneously, the robots have to localize themselves with respect to the environment to correctly achieve the mission and merge the partial maps. This kind of applications, known in the literature as simultaneous localization and mapping (SLAM), has been object of extensive research efforts, and different solutions have been proposed. For example, in Rekleitis et al. (2001) a team of two robots collaborate to explore and map a large environment, and, sens-

ing one another, they try to reduce the odometric errors and also coping with obstacles with hard-to-sense reflectance characteristics. The work in Zlot et al. (2002) proposes an approach for exploration and mapping that exploits a market architecture in order to maximize information gain while minimizing incurred cost, while the work in Burgard et al. (2005) presents experiments with a team of up to three heterogeneous robots used to reduce exploration time also in situation where the communication ranges of the robots are limited.

Part of the works in cooperative mobile robotics has a biological inspiration; this kind of approaches began after the introduction of the robotics paradigm of behavior-based control (Arkin 1989, 1998; Brooks 1986) that can be described by the relationship between the three primitives of robotics: sense, plan, and act. The behavior-based paradigm for mobile robotics has been useful for robotic researchers to examine the social characteristics of insects and animals and to apply these findings to the design of multi-robot systems. Another common application is the use of elementary control rules of various biological animals (e.g., ants, bees, birds, and fishes) to reproduce a similar behavior (e.g., foraging, flocking, homing, dispersing Mataric 1995) in cooperative robotic systems. The first works have been motivated by application in computer graphics, e.g., in 1986, Reynolds (1987) made a computer model for coordinating animal motion as bird flocks or fish schools. This pioneer work inspired significant efforts in the study of group behaviors (Parker et al. 1993; Mataric 1992; Kube and Zhang 1993) and then in the study of multi-robot formations (Wang 1991; Parker 1996; Balch and Arkin 1998).

Control Architectures and Strategies

Coordination and interaction of multiple intelligent agents have been actively studied in the field of distributed artificial intelligence since the early 1970s mainly concerning problems involving software agents. Then, in the late 1980s and early 1990s, the robotic research community became very active in the cooperative robotic

field with projects like CEBOT, SWARM, and ALLIANCE.

The works in Fukuda and Nakagawa (1988) and Fukuda et al. (1989) present the distributed hierarchical system CEBOT, inspired by cellular organization of biological entities. Such architecture results to be dynamically reconfigurable, that is, the cells are able to communicate with each other, approach, connect, and separate automatically. Moreover, if a cell is damaged, then it can be automatically repaired or replaced to maintain the system functions.

The work in Beni (1988) presents a distributed system with a large number of autonomous robots (which in the successive works has been named SWARM) that is also based on cellular robotics, and it allows the robots to accomplish cooperation tasks like assembly, communication, and computing.

The work in Parker (1994, 1996) presents a behavior-based architecture for cooperative control, called ALLIANCE. Such architecture has been developed in order to study the cooperation in an heterogeneous, small-/medium-size team of robots that should result in a fault-tolerant, reliable, and adaptive mechanism for cooperative robot control.

Successively, largely different kinds of control architectures for MMRs have been presented in literature; however, the main distinction can be done between centralized and decentralized systems. In centralized systems, a core unit collects and manages information about the environment to coordinate and control the motion of the robots and to guarantee the correct achievement of the mission. In such approaches, the core unit plays a fundamental role since it manages the whole system, i.e., it has to coordinate the information received by the distributed sensors, to manage the global information about the environment, to take all the eventual decisions, and to communicate with all the robots of the team; thus, it should be powerful enough to satisfy all the technological requirement. In decentralized approaches, instead, the resources are distributed among all the robots. Each vehicle uses its own sensors to extrapolate local information of the environment and the relative positions of the close robots to take its own decisions; moreover, each vehicle

can communicate and share information only with the close vehicles, and it is aimed at achieving only a part of the global mission.

Advantages and disadvantages of centralized and decentralized systems have been object of several discussions in the scientific community. Centralized systems, for example, can manage global information and optimize the coordination among the robots and the accomplishment of the mission; moreover, they can easily manage faults of some of the robots. On the other hand, the core unit may represent a weakness of the system; in fact, it can be the bottleneck of the system both for computational and communication time requirements; moreover, its eventual fault compromises the whole system. Decentralized systems, instead, permit to take all the advantages of distributed sensing and actuation, i.e., they make it possible to use less powerful robots or to use more cheaper sensors; they permit to optimize the allocation of the resources and to equip the robots of the team with different actuation and sensor systems; moreover, decentralized systems can easily result in tolerance to possible vehicle faults. On the other hand, within decentralized systems it is difficult to coordinate the robots and optimize the execution of the mission, and problems like global localization and mapping and communication bandwidth represent limits of this system.

In practice, many systems are not strictly centralized/decentralized. In fact, many largely decentralized architectures utilized leader agents (e.g., in Tanner et al. 2004); moreover, different hybrid centralized/decentralized architectures were presented to take partial advantages of both the typologies (Antonelli 2013); e.g., the hybrid architectures in Feddema et al. (2002), Das et al. (2002), and Stilwell et al. (2005) have central planners that perform a high-level control over mostly autonomous robots.

Beyond the architectures, different control strategies for multi-robot systems have been proposed in literature. These strategies, aimed at elaborating the motion directives to each vehicle, may differ both for mathematical characteristics and implementation aspects. As a limited set of examples, the use of control graphs to address the problem of changing the platoon formation

is discussed in Desai et al. (1999). The work in Belta and Kumar (2004) presents a formal abstraction-based approach to control a large number of robots required to move as a group; this mathematical approach is based on Lie algebra, and it is aimed at controlling the shape of the team of robots that should be kept inside an ellipsoid or rectangular area. The work in Tabuada et al. (2005) focuses on formation motion feasibility of multi-agent systems, that is, it focuses on the algebraic conditions that guarantee formation feasibility given the individual agent kinematics. The paper (Vig and Adams 2006) presents experimental results of coalition formation of a multi-robot system while simultaneously performing heterogeneous missions. Several recent control solutions rely on graph theoretic methods for multi-multi-agent networks (Mesbahi et al. 2010) or on consensus-based distributed algorithm (Ren and Randal 2008).

Future Directions

The research in multi-robot systems has matured to the point where systems with hundreds of robots (swarms) have been proposed (Parker 2003; Konolige et al. 2005; Howard et al. 2006). However, to achieve a cooperative task, the robots have to share information; thus, the increase of the team dimension directly requires an increase in the needed computational and communication resources, and it may introduce static and dynamic scalability issues. Specifically, a system is statically scalable when the control architecture can be kept exactly the same whether thousands of robots are deployed or only few are used; a system is dynamically scalable when robots can be added to or removed from the system on the fly, and they may also have the ability to reallocate and redistribute themselves in a self-organized way. In this sense, scalability still represents a limit for most of the actual platforms and control solutions for MMRs, and it can still represent an open issue for the research in this field. Furthermore, other future research directions can also be found in the field

of distributed planning and decision-making, cooperative/collective learning for multi-robot system, cloud robotics, optimal control and optimization methods, and performance evaluation and benchmarking for MMRs.

Cross-References

- [Averaging Algorithms and Consensus](#)
- [Control of Networks of Underwater Vehicles](#)
- [Estimation and Control Over Networks](#)
- [Networked Systems](#)

Bibliography

- Abreu P, Antonelli G, Arrichiello F, Caffaz A, Caiti A, Casalino G, Catenacci Volpi N, De Jong I, De Palma D, Duarte H, Gomes J, Grimsdale J, Indiveri G, Jesus S, Kebkal K, Kelholt E, Pascoal A, Polani D, Pollini L, Simetti E, Turetta A (2016) Widely scalable mobile underwater sonar technology: an overview of the H2020 WiMUST project. *Mar Technol Soc J* 50(4):42–53
- Antonelli G (2013) Interconnected dynamic systems: an overview on distributed control. *IEEE Control Syst Mag* 33(1):76–88
- Arkin RC (1989) Motor schema based mobile robot navigation. *Int J Robot Res* 8(4):92–112
- Arkin RC (1998) Behavior-based robotics. The MIT Press, Cambridge
- Asama H, Matsumoto A, Ishida Y (1989) Design of an autonomous and distributed robot system: actress. In: *Proceedings IEEE/RSJ international workshop on intelligent robots and systems' 89 (IROS'89)*, pp 283–290
- Balch T, Arkin RC (1998) Behavior-based formation control for multirobot teams. *IEEE Trans Robot Autom* 14(6):926–939
- Beard RW, Lawton J, Hadaegh FY (2001) A coordination architecture for spacecraft formation control. *IEEE Trans Control Syst Technol* 9(6):777–790
- Belta C, Kumar VK (2004) Abstraction and control of groups of robots. *IEEE Trans Robot* 20(5):865–875
- Beni G (1988) The concept of cellular robotic system. In: *Proceedings IEEE international symposium on intelligent control*, pp 57–62
- Børhaug E, Pavlov A, Panteley E, Pettersen KY (2010) Straight line path following for formations of underactuated marine surface vessels. *IEEE Trans Control Syst Technol* 19(3):493–506

- Brooks RA (1986) A robust layered control system for a mobile robot. *IEEE J Robot Autom* 2:14–23
- Burgard W, Moors M, Stachniss C, Schneider FE (2005) Coordinated multi-robot exploration. *IEEE J Robot* 21(3):376–386
- Cao YU, Fukunaga AS, Kahng AB (1997) Cooperative mobile robotics: antecedents and directions. Robot colonies. Special issue of autonomous robots, volume 4, chapter. Arkin RC, Bekey GA (eds). Kluwer Academic Publisher, Boston
- Carpin S, Parker LE (2002) Cooperative motion coordination amidst dynamic obstacles. *Distributed Auton Robot Syst* 5:145–154
- Das AK, Fierro R, Kumar V, Ostrowski JP, Spletzer J, Taylor CJ (2002) A vision-based formation control framework. *IEEE Trans Robot Autom* 18(5): 813–825
- Desai JP, Kumar V, Ostrowski JP (1999) Control of changes in formation for a team of mobile robots. In: Proceedings 1999 IEEE international conference on robotics and automation, pp 1556–1561, Detroit
- Dudek G, Jenkin MRM, Milios E, Wilkes D (1996) A taxonomy for multiagent robotics. *Auton Robot* 3(4): 375–397
- Encarnacao P, Pascoal A (2001) Combined trajectory tracking and path following for marine craft. In: Proceedings 9th Mediterranean Conference on Control and Automation
- Farinelli A, Iocchi L, Nardi D (2004) Multirobot systems: a classification focused on coordination. *IEEE Trans Syst Man Cybern Part B(Cybernetics)* 34(5):2015–2028
- Feddema JT, Lewis C, Schoenwald DA (2002) Decentralized control of cooperative robotic vehicles: theory and application. *IEEE Trans Robot Autom* 18(5): 852–864
- Fiorelli E, Leonard NE, Bhatta P, Paley D, Bachmayer R, Fratantoni DM (2004) Multi-auv control and adaptive sampling in monterey bay. In: Proceedings IEEE autonomous underwater vehicles 2004: workshop on multiple AUV operations, pp 134–147, Sebasco
- Fukuda T, Nakagawa S (1988) Dynamically reconfigurable robotic system. In: Proceedings 1988 IEEE international conference on robotics and automation, pp 1581–1586
- Fukuda T, Nakagawa S, Kawauchi Y, Buss M (1989) Structure decision method for self organising robots based on cellstructures-CEBOT. In: Proceedings 1989 IEEE international conference on robotics and automation, pp 695–700
- Gerkey BP, Mataric MJ (2004) A formal framework for the study of task allocation in multi-robot systems. *Int J Robot Res* 23(9):939–954
- Howard A, Parker LE, Sukhatme GS (2006) Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection. *Int J Robot Res* 25(5–6):431
- Huntsberger T, Pirjanian P, Trebi-Ollennu A, Das Nayar H, Aghazarian H, Ganino AJ, Garrett M, Joshi SS, Schenker PS (2003) CAMPOUT: a control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration. *IEEE Trans Syst Man Cybern Part A* 33(5):550–559
- Ihle I, Jouffroy J, Fossen T (2006) Formation control of marine surface craft: a Lagrangian approach. *IEEE J Ocean Eng* 31(4):922–934
- Konolige K, Fox D, Ortiz C, Agno A, Eriksen M, Limketkai B, Ko J, Morisset B, Schulz D, Stewart B, et al (2005) Centibots: Very large scale distributed robotic teams. In: Experimental robotics: the 9th international symposium, Springer Tracts in Advanced Robotics (STAR). Springer
- Kube CR, Zhang H (1993) Collective robotics: from social insects to robots. *Adapt Behav* 2(2):189–218
- Kushleyev A, Mellinger D, Powers C, Kumar V (2013) Towards a swarm of agile micro quadrotors. *Auton Robot* 35(4):287–300
- Mataric MJ (1992) Designing emergent behaviors: from local interaction to collective intelligence. In: Proceedings of the international conference on simulation of adaptive behavior: from animal to animal, pp 432–441
- Mataric MJ (1995) Issues and approaches in the design of collective autonomous agents. *Robot Auton Syst* 16(2):321–331
- Mataric MJ, Nilsson M, Simsarian KT (1995) Cooperative multi-robot box-pushing. In: Proceedings of the 1995 IEEE/RSJ international conference on intelligent robots and systems, pp 556–561
- Mesbahi M, Egerstedt M (2010) Graph theoretic methods in multiagent networks. Princeton University Press, Princeton
- Noreils FR (1993) Toward a robot architecture integrating cooperation between mobile robots: application to indoor environment. *Int J Robot Res* 12(1):79–98
- Parker LE (1994) ALLIANCE: an architecture for fault tolerant, cooperative control of heterogeneous mobile robots. In: Proceedings of the 1994 IEEE/RSJ/GI international conference on intelligent robots and systems, pp 776–783
- Parker LE (1996) On the design of behavior-based multi-robot teams. *Adv Robot* 10(6):547–578
- Parker LE (2003) The effect of heterogeneity in teams of 100+ mobile robots. *Multi-robot systems*. Springer, Dordrecht
- Parker LE (1993) Designing control laws for cooperative agent teams. In: Proceedings 1993 IEEE international conference on robotics and automation, volume 3, pp 582–587, Atlanta
- Rekleitis I, Dudek G, Milios E (2001) Multi-robot collaboration for robust exploration. *Ann Math Artif Intell* 31(1):7–40
- Ren W, Randal B (2008) Distributed consensus in multi-vehicle cooperative control. Springer, London
- Reynolds C (1987) Flocks, herd and schools: a distributed behavioral model. *Comput Graph* 21(4):25–34
- Spurný V, Báča T, Saska M, Pinička R, Krajník T, Thomas J, Dinesh Thakur T, Loianno G, Kumar V (2019) Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles. *J Field Robot* 36(1):125–148

- Stilwell DJ, Bishop BE (2000) Platoons of underwater vehicles. *IEEE Control Syst Mag* 20(6):45–52
- Stilwell DJ, Bishop BE, Sylvester CA (2005) Redundant manipulator techniques for partially decentralized path planning and control of a platoon of autonomous vehicles. *IEEE Trans Syst Man Cybern* 35(4):842–848
- Stipanovic DM, Inalhan G, Teo R, Tomlin CJ (2004) Decentralized overlapping control of a formation of unmanned aerial vehicles. *Automatica* 40:1285–1296
- Tabuada P, Pappas GJ, Lima P (2005) Motion feasibility of multi-agent formations. *IEEE Trans Robot* 21(3):387–392
- Tanner HG, Loizou SG, Kyriakopoulos KJ (2003) Non-holonomic navigation and control of cooperating mobile manipulators. *IEEE Trans Robot Autom* 19(1):53–64
- Tanner HG, Pappas GJ, Kumar V (2004) Leader-to-formation stability. *IEEE Trans Robot Autom* 20(3):443–455
- Vig L, Adams JA (2006) Multi-robot coalition formation. *IEEE Trans Robot* 22(4):637–649
- Wang PKC (1991) Navigation strategies for multiple autonomous robots moving in formation. *J Robot Syst* 8(2):177–195
- Wang Z, Nakano E, Takahashi T (2003) Solving function distribution and behavior design problem for cooperative object handling by multiple mobile robots. *IEEE Trans Syst Man Cybern Part A* 33(5):537–549
- Yamada S, Saito J (2001) Adaptive action selection without explicit communication for multirobot box-pushing. *IEEE Trans Syst Man Cybern Part C* 31(3):398–404
- Yamashita A, Arai T, Ota J, Asama H (2003) Motion planning of multiple mobile robots for cooperative manipulation and transportation. *IEEE Trans Robot Automat* 19(2):223
- Zlot R, Stentz A, Dias MB, Thayer S (2002) Multi-robot exploration controlled by a market economy. In: *Proceedings 2002 IEEE international conference on robotics and automation*, 3

Multiscale Multivariate Statistical Process Control

Julian Morris

School of Chemical Engineering and Advanced Materials, Centre for Process Analytics and Control Technology, Newcastle University, Newcastle Upon Tyne, UK

Abstract

Dynamic processes, both continuous and batch, are characterised by autocorrelated measurements which are allied to the effects

of process dynamics and disturbances. The common multivariate statistical process control (MSPC) approaches have been to use principal component analysis (PCA) or projection to latent structures (PLS) to build a model that captures the simultaneous correlations amongst the variables, but that ignores the serial correlation in the data during normal operations. Under such conditions it is difficult to perform efficient fault detection and diagnosis. An alternative approach to account for the process dynamics in MSPC is to use multiresolution analysis (MRA) by way of wavelet decomposition. Here, the individual measurements are decomposed into different scales (or frequencies) and the signals in each decomposed scale are then used for MSP which provides an indirect way of handling process dynamics.

Keywords

Multiresolution analysis · Partial least squares (PLS) · Principal component analysis (PCA) · Projection to latent structures (PLS) · Wavelet transform

Synonyms

MSPCA

Definition

Multiscale principal component analysis (MSPCA) and its extension multiscale projection to latent structures (MSPLS) combine the abilities of these multivariate tools to de-correlate the variables by extracting linear relationships with that of wavelet analysis, to extract deterministic features and approximately de-correlate autocorrelated measurements. Multiscale modeling makes use of the wavelet transform which allows a signal (measurement) to be viewed in multiple resolutions with each resolution representing a different frequency. That is, wavelet transform allows complex information to be decomposed

into basic components at different positions and scales.

Motivation and Background

One of the drawbacks of the conventional PCA (or PLS)-based MSPC is that although the PCA/PLS model captures the correlations among the variables, it ignores the serial (auto)correlation in the process variables and measurements. One way to overcome this issue is to include time-lagged variables in the PCA or PLS model. In this way, PCA and PLS will explicitly model both the correlations among the variables and the serial correlations in the individual variables. The impact is an increase in the number principal components required, but the multivariate monitoring model will be able to detect any changes in the serial correlation of the variables as well as changes in relationships among the variables. This article focuses on multiscale-multiway PCA using batchwise data unfolding. However, the methodology can equally be applied to PLS-based process performance monitoring (MSPC).

In multivariate statistical process control (MSPC), the multivariate statistical techniques of principle component analysis (PCA) and projection to latent structures {Partial Least Squares} (PLS) together with monitoring metrics based on Hotelling's T^2 (directly related to the Mahalanobis distance that monitors the fit of new observations to the model space) and the squared prediction error (SPE) or Q statistic (that monitors the residual space-model mismatch) are used to simultaneously monitor the process variables (Kourti and MacGregor 1996; Qin 2003). A recent survey provides an excellent state-of-the-art review of the methods and applications of data-driven fault detection and diagnosis that have been developed over the last two decades (Qin 2012).

Process measurements typically exhibit multiscale behavior as a consequence of representing the cumulative effect of a number of underlying process phenomena including process dynamics, measurement noise, and disturbances. To

address these issues, methodologies are required to address (i) the multiscale nature of process data and (ii) the inability of some existing algorithms to handle autocorrelation. One approach is through the use of multiresolution analysis and wavelets Mallat (1998). Informative discussions and application studies related to using multiresolution analysis and wavelet decompositions to enhance PCA-based process monitoring and fault detection have been presented, for example, by Bakshi (1998), Misra et al. (2002), Aradhye et al. (2003), Lu et al. (2003), Yoon and MacGregor (2004) and Reis and Saraiva (2006). Yoon and MacGregor in their comprehensive MSPCA study discussed their approach in the context of other multiscale approaches and illustrated the methodology using simulated data from a continuous stirred-tank reactor system. A major contribution of the paper was to extend fault isolation methods based on contribution plots to multiscale PCA approaches. Although some 9 years old, Ganesan et al. (2004) provided review of wavelet-based multiscale statistical process monitoring.

The Approach

Multiresolution analysis (MRA) provides the theoretical basis for the derivation of a computationally efficient algorithm for the wavelet transform Mallat (1998). MRA allows the dynamic aspects of the data in to be taken into account in MSPC. The individual signals are decomposed into different scales (frequencies), and data in each decomposed scale are then used for MSPC which provides an indirect approach to handling process dynamics. Multiscale MSPC (MSPCA) enables the simultaneous extraction of process correlations across data as well as accounting for autocorrelation within sensor data. In this way, it captures correlations among the process variables made by various events occurring at different scales.

MSPCA calculates the principal components of wavelet coefficients at each scale and combines these at the relevant wavelet scales. Due to its multiscale nature, MSPCA is very use-

ful for the modeling of data containing contributions from events whose behavior changes over both time and frequency. Process monitoring by MSPCA, and process prediction by MSPLS, involves combining those scales where significant events are detected. Approximate decorrelation of wavelet coefficients also makes MSPCA effective for the monitoring of autocorrelated measurements.

The Algorithm

Wavelets are a family of basis functions that provide a mapping from the time domain to the time-frequency domain. They can be used to decompose the signal into different resolutions by projecting onto the corresponding wavelet basis functions using multiresolution analysis (MRA). A wavelet set is constructed from a fundamental basis function or the mother wavelet by a process of translation and dilation. The wavelet set is defined as wavelet analysis which provides methodologies for the extraction of the time and frequency content of a signal. Conventional frequency analysis based on the Fourier transform consists of decomposing a signal into sine waves of different frequencies. Wavelet analysis decomposes the original signal in a similar manner. The major difference is that while Fourier analysis uses sine waves of infinite length, multiresolution analysis uses waveforms of finite length. The finite length of the wavelets allows them to describe local events in both the time and frequency domain.

The wavelet transform, an extension to the Fourier transform, projects the original signal down onto wavelet basis functions, providing a mapping from the time domain to the timescale plane. The wavelet functions, which are localized in the time and frequency domain, are obtained from a single prototype wavelet, the *mother wavelet*, by dilation and translation. The wavelet set is defined as

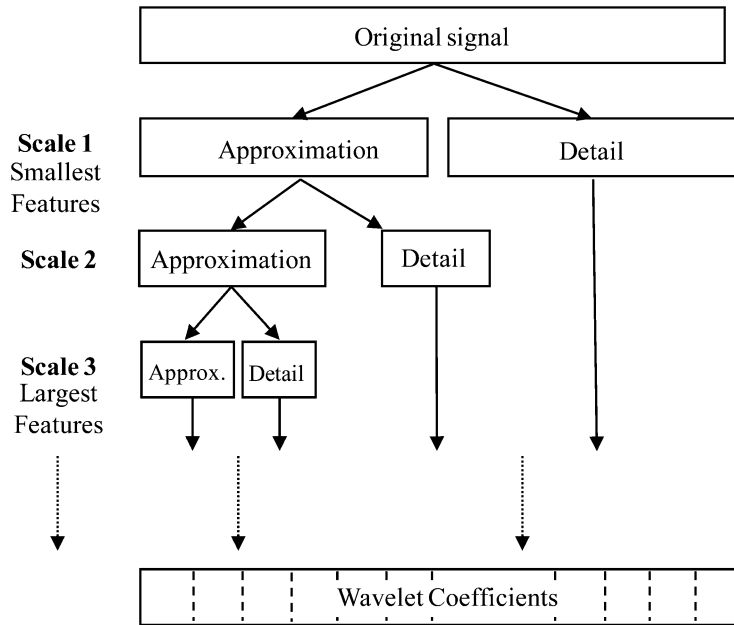
$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right)$$

where ψ is the mother wavelet function, a the dilation parameter, and b the translation parameters, and the factor $\frac{1}{\sqrt{|a|}}$ is used to ensure that each wavelet function has the same energy as the mother wavelet. The discrete wavelet transform with dyadic dilation and translation is used in this overview. A definition of continuous and discrete wavelet transforms can be found in Daubechies (1992). In the discrete case, the dilation and translation parameters are discretized as $a = a_0^m$ and $b = kb_0a_0^m$. If $a_0 = 2$ and $b_0 = 1$, a *dyadic* dilation and translation is carried out; however, a_0 and b_0 are not restricted to these values. The discrete wavelet form, which is widely used in process monitoring and chemical signal analysis, is

$$\psi_{jk}(t) = a_0^{-j/2} \psi(a_0^{-j}t - kb_0)$$

A recursive algorithm for wavelet decomposition and the reconstruction of a discrete signal of dyadic length is often used Mallat (1998) and is known as the pyramid algorithm. The fast discrete wavelet decomposition consists of three components, low-pass filters $L(n)$, high-pass filters $H(n)$, and dyadic decimation. By passing the input signal through this pair of filters, the projection of the original signal onto the scaling and wavelet functions for the multiresolution analysis is performed. Dyadic decimation, or down-sampling, removes every odd member of a sequence, thus halving the original number of samples. The low-pass filter resembles a moving average, while the high-pass filter extracts the detailed information contained in the signal. The discrete wavelet transform operates by taking a sequence of values, applying $L(n)$ and $H(n)$ and then repeating this same procedure to the approximation coefficients. In this way, the original signal vector is *smoothed and halved* through L , and the vector of approximation coefficients is again *smoothed and halved* through L . Successive application of the low-pass filter results in the approximation coefficients, becoming an increasingly smooth version of the original signal. At the same time as smoothing the signal, each iteration extracts

Multiscale Multivariate Statistical Process Control, Fig. 1 Schematic of multiresolution wavelet decomposition



the high frequency information in the data. The repeated application of L , followed by H is, in effect, a band-pass filter. The result of applying high-/low-pass filters to a signal is a set of coefficients describing the details of the signals $\mathbf{D}_L$ and a second set describing the approximations of the signals $\mathbf{A}_L$. The original signal s can then be represented by

$$x(t) = \sum_{j=1}^L D_j(t) + A_L(t)$$

where D_j and A_j are referred to as the j th level wavelet details and approximation, respectively.

Figure 1 shows schematically the multi-resolution-based wavelet decomposition.

One of the most popular choices of wavelets are those of the Daubechies' family. These wavelets are compactly supported in the time domain and have good frequency domain decay. Moreover, Daubechies' wavelets (DaubN) possess a different type of smoothness which is determined by the vanishing moments N . This makes it possible to match the wavelet smoothness to the smoothness of the signals to be analyzed. The signal can then be decomposed into its contributions from multiple scales

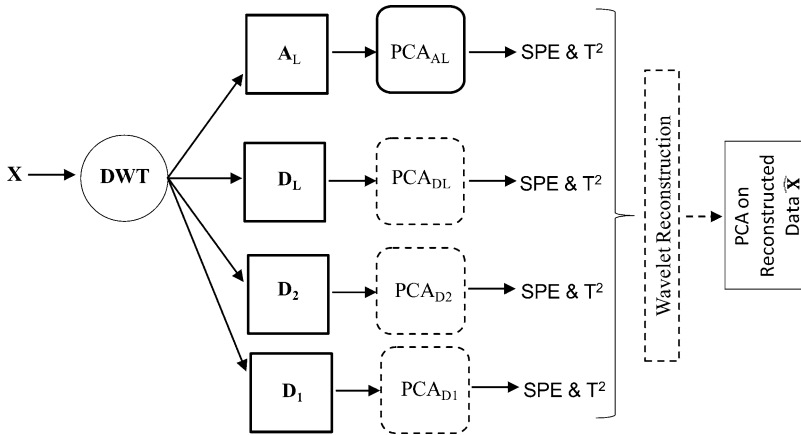
as a weighted sum of dyadically discretized orthonormal basis functions:

$$x(t) = \sum_{m=1}^L \sum_{k=1}^N d_{mk} \psi_{mk}(t) + \sum_{k=1}^N a_{lk} \phi_{lk}(t)$$

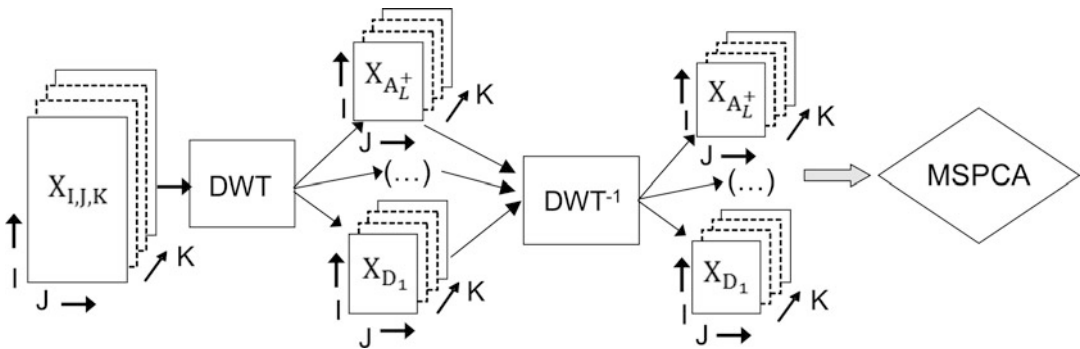
where $x(t)$ represents the process measurements, d_{mk} represents the wavelet or detail signal coefficient at scale m and location k , and a_{lk} represent the scaled signal or scaling function coefficient of $\phi(t)$ at the coarsest scale L and location k . The scaling function, or father wavelet, ϕ_{mk} captures the low-frequency content of the original signal that is not captured by wavelets at the corresponding or finer scales.

The wavelet transformation is applied to decompose a multivariate signal $\mathbf{X}$ into its approximate, $\mathbf{A}_1$ to $\mathbf{A}_L$, and detail, $\mathbf{D}_1$ to $\mathbf{D}_L$, coefficients for the first to L th level, respectively. For more information, see Bakshi (1998), Misra et al. (2002) and Aradhye et al. (2003). Figure 2 shows a schematic representation of a typical MSPCA multivariate statistical process control scheme.

An example of the application of multiway-multiscale MPCA to a benchmark-fed batch



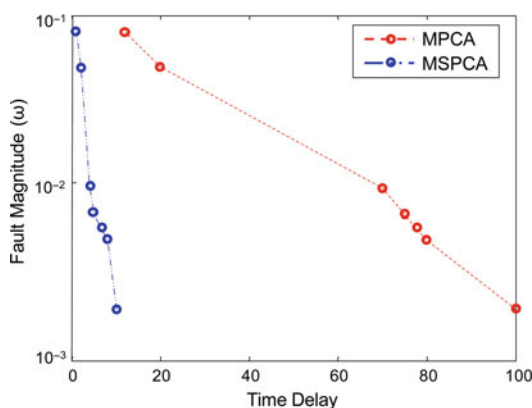
Multiscale Multivariate Statistical Process Control, Fig. 2 Schematic representation of multiscale PCA-based MSPC



Multiscale Multivariate Statistical Process Control, Fig. 3 Multiscale-multiway batch process monitoring scheme

fermentation process (Birol et al. <http://www.chee.iit.edu~control/software.html>) was presented by Alawi and Morris (2007). The application used a combination of multiblock statistical modeling approaches together with multiscale-multiway batch monitoring. Figure 3 shows the multiscale-multiway monitoring scheme for process monitoring and fault detection. At every time point, the batch process variables are decomposed into scales to the wavelet domain and then reconstructed back to the time domain. The scales/details and the approximations are collected into separate matrices (blocks). Multiblock PCA is then applied to the wavelets details and approximation. Fault detection based on the T_s^2 and Q_s statistics was used along with contribution plots incorporating confidence bounds to enhance fault diagnosis.

Figure 4 compares the monitoring statistics for the multiscale-multiway PCA and conventional multiway PCA for a slowly drifting sensor fault showing the potential for multiscale MPCA (MSPCA) in being able to detect faster subtle process and sensor faults than conventional multiway MSPC. It is noted that sensor drift is confined to one scale band at low frequency. It has been observed that multiscale approaches appear to provide little improvement if a fault effect is spread over more than one frequency band or the fault effect occurs mainly in a scale with dominant variance. Thus, a monitoring method that gives the best detection and identification of faults will depend on the fault characteristics with multiscale approaches, providing an advantage when the faults localized in frequency or that appear in scales that normally have small variance.



Multiscale Multivariate Statistical Process Control, Fig. 4 Comparison of MPCA and multiscale MSPCA for a range of subtle sensor drift faults magnitude ω against fault detection delay

Other Applications of Multiscale MPCA

There have a number of nonlinear extensions. For example, multiscale PLS approaches have been developed, e.g., Teppola and Minkkinen (2000) and Lee et al. (2009). Nonlinear approaches have also been explored. For example, Lee et al. (2004) proposed a batch monitoring approach using multiway kernel principal component analysis, Shao et al. (1999) proposed a wavelet-based nonlinear PCA algorithm, Choi et al. (2008) described a study of a kernel-based MSPCA algorithm for nonlinear multiscale monitoring, and most recently Zhang and Ma (2011) compared fault diagnosis of nonlinear processes using multiscale KPCA and multiscale KPLS. Wavelet multiscale approaches have also been widely discussed in spectroscopic data processing (Shao et al. 2004).

Cross-References

- [Controller Performance Monitoring](#)
- [Fault Detection and Diagnosis](#)
- [Statistical Process Control in Manufacturing](#)

Bibliography

- Alawi A, Zhang J, Morris AJ (2007) Multiscale Multiblock Batch Monitoring: Sensor and Process Drift and Degradation. DOI: 10.1021/op400337x April 25 2014
- Aradhye HB, Bakshi BR, Strauss A, Davis JF (2003) Multiscale SPC using wavelets: theoretical analysis and properties. *AIChE J* 49(4):939–958
- Bakshi BR (1998) Multiscale PCA with application to multivariate statistical process monitoring. *AIChE J* 44:1596–1610
- Birrol G, Undey C, Cinar AA (2002) Modular simulation package for fed-batch fermentation: penicillin production. *Comput Chem Eng* 26:1553–1565
- Choi SW, Morris J, Lee I-B (2008) Nonlinear multiscale modelling for fault detection and identification. *Chem Eng Sci* 63:2252–2266
- Daubechies I (1992) *Ten lectures on wavelets*. SIAM, Philadelphia
- Ganesan R, Das TT, Venkataraman V (2004) Wavelet-based multiscale statistical process monitoring: a literature review. *IEE Trans* 36:787–806
- Kourti T, MacGregor JF (1996) Multivariate SPC methods for process and product monitoring. *J Qual Technol* 28:409–428
- Lee J-M, Yoo C-K, Lee I-B (2004) Fault detection of batch processes using multiway Kernel principal component analysis. *Comput Chem Eng* 28(9):1837–1847
- Lee HW, Lee MW, Park JM (2009) Multi-scale extension of PLS algorithm for advanced on-line process monitoring. *Chemom Intell Lab Syst* 98: 201–212
- Liu Z, Cai W, Shao X (2009) A weighted multiscale regression for multivariate calibration of near infrared spectra. *Analyst* 134:261–266
- Lu N, Wang F, Gao F (2003) Combination method of principal component and wavelet analysis for multivariate process monitoring and fault diagnosis. *Ind Eng Chem Res* 42:4198–4207
- Mallat SG (1998) Multiresolution approximations and wavelet orthonormal bases. *Trans Am Math Soc* 315:69–87
- Misra MH, Yue H, Qin SJ, Ling C (2002) Multivariate process monitoring and fault diagnosis by multi-scale PCA. *Comp Chem Eng* 26:1281–1293
- Qin SJ (2003) *Statistical process monitoring: basics and beyond*. *J Chemom* 17:480–502
- Qin SJ (2012) Survey on data-driven industrial process monitoring and diagnosis. *Annu Rev Control* 36:220–234
- Reis MS, Saraiva PM (2006) Multiscale statistical process control with multiresolution data. *AIChE J* 52:2107–2119
- Shao R, Jia F, Martin EB, Morris AJ (1999) Wavelets and nonlinear principal components analysis for process monitoring. *Control Eng Pract* 7:865–879
- Shao X-G, Leung AK-M, Chau F-T (2004) Wavelet: a new trend in chemistry. *Acc Chem Res* 36:276–283

- Teppola P, Minkkinen P (2000) Wavelet-PLS regression models for both exploratory data analysis and process monitoring. *J Chemom* 14:383–399
- Yoon S, MacGregor JF (2004) Principal-component analysis of multiscale data for process monitoring and fault diagnosis. *AIChE J* 50(11):2891–2903
- Zhang Y, Ma C (2011) Fault diagnosis of nonlinear processes using multiscale KPCA and multiscale KPLS. *Chem Eng Sci* 66:64–72

Multi-vehicle Routing

Emilio Frazzoli¹ and Marco Pavone²

¹Massachusetts Institute of Technology,
Cambridge, MA, USA

²Stanford University, Stanford, CA, USA

Abstract

Multi-vehicle routing problems in systems and control theory are concerned with the design of control policies to coordinate several vehicles moving in a metric space, in order to complete spatially localized, exogenously generated tasks in an efficient way. Control policies depend on several factors, including the definition of the tasks, of the task generation process, of the vehicle dynamics and constraints, of the information available to the vehicles, and of the performance objective. Ensuring the stability of the system, i.e., the uniform boundedness of the number of outstanding tasks, is a primary concern. Typical performance objectives are represented by measures of quality of service, such as the average or worst-case time a task spends in the system before being completed or the percentage of tasks that are completed before certain deadlines. The scalability of the control policies to large groups of vehicles often drives the choice of the information structure, requiring distributed computation.

Keywords

Cooperative control · Decentralized control · Dynamic routing · Networked robots · Task allocation

Introduction

Multi-vehicle routing problems in systems and control theory are concerned with the design of control policies to coordinate several vehicles moving in a metric space, in order to complete spatially localized, exogenously generated tasks in an efficient way. Key features of the problem are that tasks arrive *sequentially* over time and planning algorithms should provide *control policies* (in contrast to preplanned routes) that prescribe how the routes should be updated as a function of those inputs that change in real time. This problem is usually referred to as dynamic vehicle routing (DVR). In DVR problems, ensuring the stability of the system, i.e., the uniform boundedness of the number of outstanding tasks, is a primary concern.

Motivation and Background

As a motivating example, consider the following scenario: a team of unmanned aerial vehicles (UAVs) is responsible for investigating possible threats over a region of interest. As possible threats are detected, by intelligence, high-altitude or orbiting platforms, or by ground sensor networks, one of the UAVs must visit its location and investigate the cause of the alarm, in order to enable an appropriate response if necessary. Performing this task may require the UAV not only to fly to the possible threat's location but also to spend additional time on site. The objective is to minimize the average time between the appearance of a possible threat and the time one of the UAVs completes the close-range inspection task. Variations may include priority levels, time windows during which the inspection task must be completed, and sensors with limited range.

In order to perform the required mission, the UAVs (or, more in general, mission control) need to repeatedly solve three *coupled* decision-making problems:

1. **Task allocation:** which UAV shall pursue each task? What policy is used to assign tasks to UAVs? How often should the assignment be revised?

2. **Service scheduling:** given the list of tasks to be pursued, what is the most efficient ordering of these tasks?
3. **Loitering paths:** what should UAVs without pending assignments do?

The optimization process must take into account, for example, algebraic or differential constraints (such as obstacle avoidance or bounded curvature, respectively), sensing constraints, communication constraints, and energy constraints. Furthermore, one might require a decentralized control architecture.

DVR problems, including the above UAV routing problem, are generally *intractable* due to their multifaceted combinatorial, differential, and stochastic nature, and consequently solution approaches have been devised that look either at heuristic algorithms or at approximation algorithms with some guarantee on their performance.

Related Problems

DVR problems represent the dynamic counterpart of the well-known static vehicle routing problem (VRP), whereby (i) a team of n vehicles is required to service a set of n_T “static” tasks in a metric space, (ii) each task requires a certain amount of on-site service, (iii) and the goal is to compute a set of routes that minimizes the cost of servicing the tasks; see Toth and Vigo (2001) for a thorough introduction to this problem. The VRP is *static* in the sense that vehicle routes are computed assuming that no new tasks arrive. The VRP is an important research topic in the operations research community.

Approaches for Multi-vehicle Routing

Broadly speaking, there are three main approaches available in the literature to tackle dynamic vehicle routing problems. The first approach relies on heuristic algorithms. In the second approach, called “competitive analysis approach,” routing policies are designed to minimize the *worst-case* ratio between their performance and the performance of an optimal off-line algorithm

which has a priori knowledge of the entire input sequence. In the third approach, the routing problem is embedded within the framework of queueing theory. Routing policies are then designed to *stabilize* the system in terms of uniform boundedness of the number of outstanding tasks and to minimize typical queueing-theoretical cost functions such as the *expected time* the tasks remain in the queue. Since the generation of tasks and motion of the vehicles is within an Euclidean space, one can refer to this third approach as “spatial queueing theory.”

Heuristic Approach

The main aspect of the heuristic approach is that routing algorithms are evaluated primarily via numerical, statistical and experimental studies, and formal performance guarantees are not available. A naïve, yet reasonable approach to design a heuristic algorithm for DVR would be to adapt classic queueing policies. However, perhaps surprisingly, this adaptation is not at all straightforward. For example, routing algorithms based on a first-come-first-served policy, whereby tasks are fulfilled in the order in which they arrive, are unable to stabilize the system for all stabilizable task arrival rates, in the sense that with such routing algorithms the average number of tasks grows over time without bound, even though there exist alternative routing algorithms that would maintain the number of tasks uniformly bounded (Bertsimas and van Ryzin 1991).

The most widely applied approach is to combine static routing methods (e.g., VRP-like methods, nearest neighbor strategies, or genetic algorithms) and sequential re-optimization, where the re-optimization horizon is chosen heuristically. In particular, greedy nearest neighbor strategies, whose formal characterization still represents an open problem, are known to perform particularly well in some notable cases (Bertsimas and van Ryzin 1991). However, the joint selection of a static routing method and of the re-optimization horizon in the presence of vehicle and task constraints (e.g., differential motion constraints, or task priorities) makes the application of this

approach far from trivial. For example, one can show that an erroneous selection of the re-optimization horizon can lead to pathological scenarios where no task *ever* receives service (Pavone 2010). Additionally, performance criteria in dynamic settings commonly differ from those of the corresponding static problems. For example, in a dynamic setting, the time needed to complete a task may be a more important factor than the total vehicle travel cost.

Competitive Analysis Approach

The distinctive feature of the competitive analysis approach is the method used to evaluate an algorithm's performance, which is called *competitive analysis*. In competitive analysis, the performance of a (causal) algorithm is compared to the performance of a corresponding off-line algorithm (i.e., a non-causal algorithm that has a priori knowledge of the entire input) in the worst-case scenario. Specifically, an algorithm is c -competitive if its cost on *any* problem instance is at most c times the cost of an optimal off-line algorithm:

$$\text{Cost}_{\text{causal}}(I) \leq c \text{Cost}_{\text{optimal off-line}}(I),$$

for all problem instances I .

In the recent past, several dynamic vehicle routing problems have been successfully studied in this framework, under the name of the online traveling repairman problem (Jaillet and Wagner 2006), and many interesting insights have been obtained. However, the competitive analysis approach has some potential disadvantages. First, competitive analysis is a *worst-case* analysis; hence, the results are often overly pessimistic for normal problem instances, and potential statistical information about the problem (e.g., knowledge of the spatial distribution of future tasks) is often neglected. Second, the worst-case analysis usually requires a *finite* horizon problem formulation, which precludes the study of useful properties such as stability. Third, competitive analysis is used to bound the performance relative to an optimal *off-line* algorithm, which, by being

non-causal, does *not* belong to the feasible set of routing algorithms one is optimizing over. Hence, with this approach one minimizes the “cost of causality” in the worst-case scenario, but not necessarily the worst-case cost (which would require comparison with an optimal *causal* routing algorithm). Finally, many important real-world constraints for DVR, such as time windows, priorities, differential constraints on vehicle's motion, and the requirement of teams to fulfill a task, have so far proved to be too complex to be considered in the competitive analysis framework (Golden et al. 2008, page 206). Some of these drawbacks have been recently addressed by Van Hentenryck et al. (2009) where a combined stochastic and competitive analysis approach is proposed for a general class of combinatorial optimization problems and is analyzed under some technical assumptions.

Spatial Queueing Theory

Spatial queueing theory embeds the dynamic vehicle routing problem within the framework of queueing theory. Spatial queueing theory consists of three main steps, namely, development of a spatial queueing model, establishment of fundamental limitations of performance, and design of algorithms with performance guarantees. More specifically, the formulation of a model entails detailing four main aspects:

1. A model for the *dynamic* component of the environment: this is usually achieved by assuming that new events are generated (either adversarially or stochastically) by an exogenous process.
2. A model for targets/tasks: tasks are usually modeled as points in a physical environment distributed according to some (possibly unknown) distribution, might require a certain level of on-site service time, and can be subject to a variety of constraints, e.g., time windows, priorities, etc.
3. A model for the vehicles and their motion: besides their number, one needs to specify whether the vehicles are subject to algebraic (e.g., obstacles) or differential (e.g., min-

imum turning radius) constraints, sensing constraints, and fuel constraints. Also, the control could be centralized (i.e., coordinated by a central station) or decentralized and subject to communication constraints.

4. Performance criterion: examples include the minimization of the waiting time before service, loss probabilities, expectation-variance analysis, etc.

Once the model is formulated, one seeks to characterize fundamental limitations of performance (in the form of lower bounds for the best achievable cost); the purpose of this step is essentially twofold: it allows the quantification of the degree of optimality of a routing algorithm and provides structural insights into the problem. As for the last step, the design of a routing algorithm usually relies on a careful combination of static routing methods with sequential re-optimization. Desirable properties for the static methods are the following: (i) the static problem can be solved (at least approximately) in polynomial time and (ii) the static method is amenable to a statistical characterization (this is essential for the computation of performance bounds). Formal performance guarantees on a routing algorithm are then obtained by quantifying the ratio between an upper bound on the cost delivered by that algorithm and a lower bound for the best achievable cost. Such a ratio, being an estimate of the degree of optimality of the algorithm, should be close to one and possibly independent of system parameters. The proposed algorithms are finally evaluated via numerical, statistical and experimental studies, including Monte-Carlo comparisons with alternative approaches.

An interesting feature of this approach is that the performance analysis usually yields scaling laws for the quality of service in terms of model data, which can be used as useful guidelines to select system parameters when feasible (e.g., number of vehicles).

In order to make the model tractable, the arrival process of tasks is assumed stationary (with possibly unknown parameters) with

statistically independent arrival times. These assumptions, however, can be unrealistic in some scenarios, in which case the competitive analysis approach may represent a better alternative. From a technical standpoint, one should note that spatial queueing models are *inherently* different from traditional, nonspatial queueing models. The main reason is that in spatial queueing models, the “service time” per task has both a *travel* and an *on-site* component. Although the on-site service requirements can often be modeled as “statistically” independent, the travel times are *inherently* statistically coupled. Hence, in contrast to standard queueing models, service times in spatial queueing models are statistically *dependent*, and this deeply affects the solution to the problem.

Pioneering work in this context is that of Bertsimas and van Ryzin (1991), who introduced queueing methods to solve the baseline DVR problem (a vehicle moves along straight lines and visits tasks whose time of arrival, location, and on-site service are stochastic; information about task location is communicated to the vehicle upon task arrival). Next section provides an overview of the application of spatial queueing theory to such simplified DVR problem, referred to in the literature as dynamic traveling repairman problem (DTRP).

Applying Spatial Queueing Theory to DVR Problems

Spatial Queueing Theory Workflow for DTRP

Model

The DTRP, which, incidentally, captures well the salient features of the UAV scenario outlined in the Motivation Section, can be modeled as follows. In a geographical region $\mathcal{Q}$ of area $\mathcal{A}$, a dynamic process generates spatially localized tasks. The process generating tasks is modeled as a spatio-temporal Poisson process, i.e., (i) the time between consecutive generation instants has

an exponential distribution with intensity $\lambda > 0$ and (ii) upon arrival, the locations of tasks are independently and uniformly distributed in $\mathcal{Q}$. The location of the new tasks is assumed to be immediately available to a team of n servicing vehicles. The vehicles provide service in $\mathcal{Q}$, traveling at a speed at most equal to v ; the vehicles are assumed to have unlimited fuel and task-servicing capabilities. Each task requires an independent and identically distributed amount of on-site service with finite mean duration $\bar{s} > 0$. A task is completed when one of the vehicles moves to its location and performs its on-site service. The objective is to design a *routing policy* that maximizes the quality of service delivered by the vehicles in terms of the average steady-state time delay $\bar{T}$ between the generation of a task and the time it is completed (in general, in a dynamic setting, the focus is on the quality of service as perceived by the “end user,” rather than, for example, fuel economies achieved by the vehicles). Other quantities of interest are the average number $\bar{n}_T$ of tasks waiting to be completed and the waiting time $\bar{W}$ of a task before its location is reached by a vehicle. These quantities, however, are related according to $\bar{T} = \bar{W} + \bar{s}$ (by definition) and by Little’s law, stating that $\bar{n}_T = \lambda \bar{W}$, for stable queues.

The system is considered stable if the expected number of waiting tasks is uniformly bounded at all times, or equivalently, that tasks are removed from the system at least at the same rate at which they are generated. In the case at hand, the time to complete a task is the sum of the time to reach its location (which depends on the routing policy) plus the time spent at that location in on-site service (which is independent of the routing policy). Since, by definition, the service time is no shorter than the on-site service time $\bar{s}$, then a weaker necessary condition for stability is $\varrho := \lambda \bar{s} / n < 1$; the quantity ϱ measures the fraction of time the vehicles are performing on-site service. Remarkably, it turns out that this is also a sufficient condition for stability, in the sense that, if this condition is satisfied, one can find a stabilizing policy. Note that this stability condition is independent of the size and shape of $\mathcal{Q}$ and of the speed of the vehicles.

Fundamental Limitations of Performance

To derive lower bounds, the main difficulty consists in bounding (possibly in a statistical sense) the amount of time spent to reach a target location. The derivation of these bounds becomes simpler in asymptotic regimes, i.e., looking at cases when $\varrho \rightarrow 0^+$ and $\varrho \rightarrow 1^-$, which are often called “light-load” and “heavy-load” conditions, respectively.

Consider first the case in which $\varrho \rightarrow 0^+$ (light-load regime). A set of n points is called the n -median of $\mathcal{Q}$ if it globally minimizes the expected distance between a random point sampled uniformly from $\mathcal{Q}$ and the closest point in such set. In other words, the n -median of $\mathcal{Q}$ globally minimizes the function

$$\begin{aligned} H_n(p_1, p_2, \dots, p_n) \\ &:= \mathbb{E} [\min_{k \in \{1, \dots, n\}} \|p_k - q\|] \\ &= \frac{1}{A} \int_{\mathcal{Q}} \min_{k \in \{1, \dots, n\}} \|p_k - q\| dq. \end{aligned}$$

Let H_n^* be the global minimum of this function. Geometric considerations show that H_n^* scales proportionally to $\sqrt{A/n}$.

Incidentally, the n -median of $\mathcal{Q}$ induces a Voronoi partition that is called *Median Voronoi Tessellation*, whose importance will become clear in the next section. Recall that the Voronoi diagram of $\mathcal{Q}$ induced by points $(p_1, \dots, p_n)$ is defined by

$$\begin{aligned} V_i = \left\{ q \in \mathcal{Q} \mid \|q - p_i\| \leq \|q - p_j\|, \forall j \neq i, \right. \\ \left. j \in \{1, \dots, n\} \right\}, \end{aligned}$$

where V_i is the region associated with the i -th “generator” point p_i (see also ► [Optimal Deployment and Spatial Coverage](#)). The distance H_n^* certainly provides a lower bound on the expected distance traveled by a vehicle to reach a task, and hence one obtains the lower bound

$$\bar{T} \geq \frac{H_n^*}{v} + \bar{s}.$$

This lower bound is tight in light-load conditions ($\varrho \rightarrow 0^+$), as it will be seen in the next section.

Consider now the case in which $\varrho \rightarrow 1^-$ (heavy load). Let $\bar{D}$ be the average travel distance per task for some routing policy. By using arguments from geometrical probability (independent of algorithms), one can show that $\bar{D} \geq \beta_2 \sqrt{A}/\sqrt{2n_T}$ as $\varrho \rightarrow 1^-$, where β_2 is a constant that will be specified later. As discussed, for stability, one needs $\bar{s} + \bar{D}/v < n/\lambda$. Combining the stability condition with the bound on the average travel distance per task, one obtains

$$\bar{s} + \frac{\beta_2 \sqrt{A}}{v \sqrt{2n_T}} \leq \frac{n}{\lambda}.$$

Since, by Little's law, $\bar{n}_T = \lambda \bar{W}$ and $\bar{T} = \bar{W} + \bar{s}$, one finally obtains (recall that $\varrho = \lambda \bar{s}/n$):

$$\bar{T} \geq \frac{\beta_2^2 A}{2} \frac{\lambda}{v^2 n^2 (1 - \varrho)^2} + \bar{s}, \quad (\text{as } \varrho \rightarrow 1^-).$$

A salient feature of the above lower bound is that it scales *quadratically* with the number of vehicles (as opposed to the square-root scaling law one has in light-load conditions); note, however, that congestion effects are not included in this model. This bound also shows that the quality of service, which is proportional to $1/(1 - \varrho)^2$, degrades much faster as the target load increases than in nonspatial queueing systems (where the growth rate is proportional to $1/(1 - \varrho)$).

Design of Routing Algorithms

The design of an optimal light-load policy essentially relies on mimicking the proof strategy employed for the light-load lower bound. Specifically, a routing policy whereby (1) one vehicle is assigned to each of the n median locations of $\mathcal{Q}$, (2) new tasks are assigned to the nearest median location and its corresponding vehicle, and (3) each vehicle services tasks according to a first-come-first-served policy is asymptotically optimal, i.e.,

$$\bar{T} \rightarrow \frac{H_n}{v} + \bar{s}, \quad (\text{as } \varrho \rightarrow 0^+).$$

Note that under this strategy “regions of dominance” are implicitly assigned to vehicles according to a Median Voronoi Tessellation.

The heavy-load case is more challenging. Consider, first, the following single-vehicle routing policy, based on a partition of $\mathcal{Q}$ into $p \geq 1$ subregions $\{Q_1, Q_2, \dots, Q_p\}$ of equal area A/p . Such a partition can be obtained, e.g., as sectors centered at the median of $\mathcal{Q}$. Define a cyclic ordering for the subregion, such that, e.g., if the vehicle is in region Q_i , the “next” region is Q_j , where j follows i in the cyclic ordering (in other words, $j = (i + 1) \bmod p$).

1. If there are no outstanding tasks, move to the median of the region Q .
2. Otherwise, visit the “next” subregion; subregions with no tasks are skipped. Compute a minimum-length path from the vehicle's current position through all the outstanding tasks in that subregion. Complete all tasks on this path, ignoring new tasks generated in the meantime. Repeat.

The problem of computing the shortest path through a number of points is related to the well-known traveling salesman problem (TSP). While the TSP is a prototypically hard combinatorial optimization problem, it is well known that the Euclidean version of the problem can be approximated efficiently (Vazirani 2001). Furthermore, the length ETSP(n_T) of a Euclidean TSP through n_T points independently and uniformly sampled in $\mathcal{Q}$ is known to satisfy the following property:

$$\lim_{n_T \rightarrow \infty} \text{ETSP}(n_T)/\sqrt{n_T} = \beta_2 \cdot \sqrt{A}, \quad \text{almost surely,}$$

where $\beta_2 \approx 0.712$ is a constant (the same β_2 constant that appeared in the previous section) (Steele 1990).

It can be shown that, using the above routing policy, the average system time $\bar{T}$ satisfies

$$\bar{T} \leq \gamma(p) \frac{A}{v^2} \frac{\lambda}{(1 - \varrho)^2} + \bar{s}, \quad (\text{as } \varrho \rightarrow 1^-),$$

where $\gamma(1) = \beta_2^2$ and $\gamma(p) \rightarrow \beta_2^2/2$ for large p . These results critically exploit the statistical characterization of the length of an optimal TSP tour. Hence, the proposed policy achieves a quality of service that is arbitrarily close to the optimal one, in the asymptotic regime of heavy load (and, indeed, also of light load).

The above single-vehicle routing policies can be fairly easily lifted to an efficient multi-vehicle routing policy. The key idea (akin to the one in the light-load case) is to (1) partition the workspace into n *regions of dominance* (with disjoint interiors and whose union is $\mathcal{Q}$), (2) assign one vehicle to each region, and (3) have each vehicle follow a single-vehicle routing policy within its own region. This approach leads to the following multi-vehicle routing policy for the DTRP problem:

1. Partition $\mathcal{Q}$ into n regions of dominance of *equal area* and assign one vehicle to each region.
2. Each vehicle executes a single-vehicle DTRP policy in its own subregion.

Using as single-vehicle policy the routing policy described above, the average system time $\bar{T}$ in heavy-load satisfies

$$\bar{T} \leq \gamma(p) \frac{A}{v^2} \frac{\lambda}{n^2 (1 - \rho)^2} + \bar{s}, \quad (\rho \rightarrow 1^-).$$

Hence, by comparing this result with the corresponding lower bound, one concludes that a simple partitioning strategy leads to a multi-vehicle routing policy whose performance is arbitrarily close to the optimal one in heavy load.

Mode of Implementation

The scalability of the control policies to large groups of vehicles often requires a distributed implementation of multi-vehicle routing strategies. For the DTRP, a distributed implementation can be obtained by devising *decentralized*

algorithms for environment partitioning. In the solution proposed in Pavone (2010), power diagrams are the key geometric concept to obtain, in a decentralized fashion, partitions suitable for both the light-load case (requiring, as seen before, a Median Voronoi Tessellation) and the heavy-load case (requiring an equal-area partition). The power diagram of $\mathcal{Q}$ is defined as

$$V_i = \left\{ q \in \mathcal{Q} \mid \|q - p_i\|^2 - w_i \leq \|q - p_j\|^2 - w_j, \right. \\ \left. \forall j \neq i, j \in \{1, \dots, n\} \right\},$$

where $(p_i, w_i) \in \mathcal{Q} \times \mathbb{R}$ are a set of “power points” and V_i is the subregion associated with the i -th power point. Note that power diagrams are a generalization of Voronoi diagrams: when all weights are equal, the power diagram and the Voronoi diagram are identical. The basic idea, then, is to associate to each vehicle i a *virtual* power point, which is an artificial (or logical) variable whose value is locally controlled by the i -th vehicle. The cell V_i becomes the region of dominance for vehicle i , and each vehicle updates its own power point according to a *decentralized* gradient-descent law with respect to a coverage function (► [Optimal Deployment and Spatial Coverage](#)), until the desired partition is achieved. The reader is referred to Pavone (2010) for more details.

Extensions and Discussion

By integrating additional ideas from dynamics, teaming, and distributed algorithms, the spatial queueing theory approach has been recently applied to scenarios with complex models for the tasks such as time constraints, service priorities, translating tasks, and adversarial generation; has been extended to address aspects concerning robotic implementation such as complex vehicle dynamics, limited sensing range, and team forming; and has even been tailored to integrate humans in the design space; see Bullo et al. (2011) and references therein. Despite the significant modeling differences, the “workflow” is essentially the same as in

the DTRP: a queueing model that captures the salient features of the problem at hand, characterization of the fundamental limitations of performance, and design of algorithms with provable performance bounds. The last step, as for the DTRP, often involves lifting a single-vehicle policy to a multi-vehicle policy through the strategy of environment partitioning. Within this context, a number of partitioning schemes and corresponding *decentralized* partitioning algorithms relevant to a large variety of DVR problems are discussed in Pavone et al. (2009).

This workflow efficiently and transparently *decouples* the three decision-making problems mentioned in the Introduction Section, i.e., “task allocation,” “service scheduling,” and “loitering paths.” In fact, task allocation is addressed via the strategy of environment partitioning, service scheduling is addressed by applying a single-vehicle routing policy within the individual regions of dominance, and the loitering paths resolve in placing the vehicles at or around specific points within the dominance regions (e.g., the median). Note, however, that in some important cases, e.g., DVR problems where goods have to be transported from a pickup location to a delivery location or where vehicles are differentially constrained and operate in a “congested” workspace, multi-vehicle policies that rely on static partitions perform poorly or are not even feasible (Pavone et al. 2009), and task allocation and service scheduling need to be addressed as tightly coupled.

Through spatial queueing theory one is usually able to characterize the performance of multi-vehicle routing policies in asymptotic regimes. To ensure “satisfactory” performance under general operation conditions, a common strategy is to consider heuristic modifications to a baseline asymptotically efficient routing policy in such a way that, on the one hand, asymptotic performance is preserved, and, on the other hand, light- and heavy-load performances are “smoothly” and efficiently blended in the intermediate load case. The interested reader can find more information in Bullo et al. (2011).

Summary and Future Directions

The three main approaches available to tackle DVR problems are (i) heuristic algorithms, (ii) competitive analysis, and (iii) spatial queueing theory. Broadly speaking, the competitive analysis approach is well suited when worst-case guarantees are sought, e.g., because there is not enough statistical information about the problem at hand. Spatial queueing theory represents a powerful alternative in cases where it is possible to leverage statistical information and one seeks average-case guarantees. Finally, for some problems the complexity of the model makes an analytical treatment very difficult, in which case the only option is to resort to an heuristic approach (possibly relying on insights derived by applying competitive analysis and/or spatial queueing theory to a simplified version of the problem).

Future directions include the extension of the three aforementioned approaches to increasingly complex problem setups, for example, higher-fidelity vehicle dynamics and environments and sophisticated sensing and communication constraints, novel applications (e.g., search and rescue missions, map maintenance, and pursuit-evasion), and inclusion of game-theoretical tools to address adversarial scenarios. Specifically, for the spatial queueing theory approach, key future directions include the problem of addressing optimality of performance in intermediate regimes (current optimality results are only available either in the light or heavy-load regimes), online estimation of the statistical parameters (e.g., spatial distribution of the tasks), and formulations that take into account second-order moments and large-deviation probabilities.

Cross-References

- [Averaging Algorithms and Consensus](#)
- [Flocking in Networked Systems](#)
- [Networked Systems](#)
- [Optimal Deployment and Spatial Coverage](#)
- [Particle Filters](#)

Bibliography

- Bertsimas DJ, van Ryzin GJ (1991) A stochastic and dynamic vehicle routing problem in the Euclidean plane. *Oper Res* 39:601–615
- Bullo F, Frazzoli E, Pavone M, Savla K, Smith SL (2011) Dynamic vehicle routing for robotic systems. *Proc IEEE* 99(9):1482–1504
- Golden B, Raghavan S, Wasil E (2008) The vehicle routing problem: latest advances and new challenges. Volume 43 of operations research/computer science interfaces. Springer, New York
- Jaillet P, Wagner MR (2006) Online routing problems: value of advanced information and improved competitive ratios. *Transp Sci* 40(2):200–210
- Pavone M (2010) Dynamic vehicle routing for robotic networks. PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology
- Pavone M, Savla K, Frazzoli E (2009) Sharing the load. *IEEE Robot Autom Mag* 16(2):52–61
- Steele JM (1990) Probabilistic and worst case analyses of classical problems of combinatorial optimization in Euclidean space. *Math Oper Res* 15(4):749
- Toth P, Vigo D (eds) (2001) The vehicle routing problem. Monographs on discrete mathematics and applications. SIAM, Philadelphia. ISBN:0898715792
- Van Hentenryck P, Bent R, Upfal E (2009) Online stochastic optimization under time constraints. *Ann Oper Res* 177(1):151–183
- Vazirani V (2001) Approximation algorithms. Springer, New York

Nash Equilibrium

► [Strategic Form Games and Nash Equilibrium](#)

Nash Equilibrium Seeking over Networks

Lacra Pavel
Department of Electrical and Computer
Engineering, University of Toronto, Toronto,
ON, Canada

Abstract

In the classical approach of Nash equilibrium seeking, players make decisions such that their own cost is minimized assuming that they observe all others' actions/decisions. This requirement can be impractical in distributed multi-agent systems and networks. When extending Nash equilibrium seeking to networks, one has to consider the issue of incomplete information on the opponents' decisions and how handle it via networked information exchange. This entry provides an overview of algorithms/dynamics that can accomplish this. We focus on the interplay between game and communication graph properties in achieving convergence and on system theoretic connections.

Keywords

Nash equilibrium · Learning in games ·
Networks · Graphs · Consensus ·
Multi-agent systems

Introduction

Many examples of decision-making problems in multi-agent networks can be modeled as games on/over networks (Marden and Shamma 2013; Pavel 2012; Stankovic et al. 2012; Bauso 2016; Yin et al. 2011). The agents can be the network nodes, the users in a communication network, or the robots in a coverage control problem. They are modeled as players in a game, each with their own goal to pursue (e.g., maximum transmission bandwidth or minimum interference). Agents are taking actions/decisions so as to maximize their own individual payoff/utility or equivalently, to minimize their cost, but their payoff/cost depends on (is coupled to) what others are doing. The relevant game equilibrium solution is the Nash equilibrium (Nash 1950), a state where each agent's decision is individually optimal with respect to the others' decisions/strategies. This Nash equilibrium (NE) state should be achieved in an online and autonomous manner, even though agents' own decision-making is coupled to the others' decisions. The dynamic decision-making processes by which players attempt to reach a NE are called Nash equilibrium seeking algorithms/dynamics, part of the larger topic of

learning in games (Fudenberg and Levine 1998). In the classical approach, each player observes all others' actions/decisions. In recent years, there have been sustained efforts to generalize the classical setting of Nash equilibrium seeking and consider the issue of incomplete information on the opponents' decisions and how to deal with it via networked information exchange (Gharesifard and Cortes 2013; Wang et al. 2013; Li and Marden 2013; Swenson et al. 2015; Koshal et al. 2016; Salehisadaghiani and Pavel 2016).

This effort is motivated by the proliferation of engineering networked applications requiring distributed protocols that operate under local information (e.g., vehicle-to-vehicle, peer-to-peer, ad hoc, smart-grid networks), where a player is inherently limited to being observable to and communicate with a few other players. In this entry, we discuss approaches proposed for Nash equilibrium over networks, focused on continuous-kernel games and pointing out some of the challenges and future directions.

Motivating Examples

As a motivating example, consider distributed routing over a network modeled as a noncooperative congestion game. A set of N drivers select their route by relying on local information obtained from their neighbors, such as in vehicular ad hoc networks via community-based navigation, e.g., Waze. Drivers/agents select a route so that they minimize their own driving time, but each does it based on communicating with drivers in their own neighborhood in order to gather information about the other drivers and the state of the traffic. Another example is opportunistic spectrum access for cognitive radio networks, where unlike traditional wireless networks, users adaptively adjust their operating parameters based on interactions with the environment and other users in the network (Cheng et al. 2014; Swenson et al. 2015). These examples are noncooperative in the way actions are taken (each agent takes an action to minimize its own individual cost function), but collaborative, in the sense that agents agree to share some information with neighbors to compensate for the lack of global information on others' decisions. In both

cases, the goal is for agents to reach a Nash equilibrium.

NE Characterization and Classical NE Seeking

We start by giving a formal characterization of a Nash equilibrium (NE) and classical NE seeking schemes. Consider a set $\mathcal{N} = \{1, \dots, N\}$ of N players (agents) involved in a game. Each player $i \in \mathcal{N}$ controls its action/decision $x_i \in \Omega_i$, where $\Omega_i \subseteq \mathbb{R}^{n_i}$ and aims to minimize its own cost function $J_i(x_i, x_{-i})$, $J_i : \Omega \rightarrow \mathbb{R}$, which depends on possibly all other players' actions, x_{-i} . By $x_{-i} \in \Omega_{-i}$ we denote the $(N-1)$ -tuple of all agents' actions/decisions except agent i 's. Let $x = (x_i, x_{-i}) \in \Omega$, or in stacked notation, $x = [x_1^T \dots x_N^T]^T \in \Omega$, denote the profile of all agents' actions, where $\Omega = \prod_{i \in \mathcal{N}} \Omega_i \subseteq \mathbb{R}^n$, $n = \sum_{i \in \mathcal{N}} n_i$. We denote this game by $\mathcal{G}(\mathcal{N}, \Omega_i, J_i)$.

Definition 1 Given a game $\mathcal{G}(\mathcal{N}, \Omega_i, J_i)$, an action profile $x^* = (x_i^*, x_{-i}^*) \in \Omega$ is a Nash equilibrium (NE) of the game if

$$(\forall i \in \mathcal{N})(\forall x_i \in \Omega_i) \quad J_i(x_i^*, x_{-i}^*) \leq J_i(x_i, x_{-i}^*)$$

At a Nash equilibrium, no agent has any incentive to unilaterally change its decision; hence, a NE is a self-enforceable solution. Existence of a Nash equilibrium is guaranteed based on Kakutani's fixed-point theorem, when Ω_i is non-empty, compact, and convex and $J_i(x_i, x_{-i})$ is jointly continuous and convex in x_i for every given x_{-i} , cf. Theorem 4.4 in Başar and Olsder (1999). For continuously differentiable costs, a NE $x^* \in \Omega$ is characterized as a solution of the following variational inequality:

$$(x - x^*)^T F(x^*) \geq 0 \quad \forall x \in \Omega \quad (1)$$

where $F(x) := (\nabla_{x_i} J_i(x_i, x_{-i}))_{i \in \mathcal{N}}$ denotes the stacked vector with all partial gradients $\nabla_{x_i} J_i(x_i, x_{-i})$, of J_i with respect to x_i , cf. Proposition 1.4.2 (Facchinei and Pang 2007). Agents have different cost functions so F :

$\Omega \rightarrow \mathbb{R}^n$ is not a true gradient, and it is called the *pseudo-gradient (game) map*. Alternatively, $x^* = (x_i^*, x_{-i}^*)$ is a (fixed-point) solution to the set of equations, cf. Proposition 1.5.8 (Facchinei and Pang 2007):

$$x_i^* = P_{\Omega_i}(x_i^* - \alpha_i \nabla_{x_i} J_i(x_i^*, x_{-i}^*)), \forall i \in \mathcal{N}, \quad (2)$$

where P_{Ω_i} denotes the Euclidean projection onto Ω_i .

The inherent coupling between the agents' decisions, (2), introduces challenges: when optimizing for their own reward, agents need to know the others' decisions. In a repeated game, players use previous game iterations to gather information about other agents or the game and adjust their decisions/actions. The hope is for agents to come up with a solution of the game, hence find by themselves or seek the unknown NE x^* . The process that describes how an agent i updates its decision in time gives the agent i 's learning/seeking dynamics, denoted by Σ_i . Typical classes of dynamics are best-response play (fictitious-play), projected-gradient (better-response) play, and reinforcement-learning (payoff-based learning) (Fudenberg and Levine 1998; Frihauf et al. 2012), each with different information requirements and convergence guarantees. In this entry, we consider the first two classes. In the classical approach to Nash equilibrium seeking via best-response or gradient (better-response) play, agents observe all other players' actions as in Li and Başar (1987), Shamma and Arslan (2005), Facchinei and Pang (2007), and Scutari et al. (2014). For example, based on (2), in a discrete-time projected gradient-play, at iteration k each player i updates its action as in

$$\Sigma_i : x_i(k+1) = P_{\Omega_i}(x_i(k) - \alpha_{k,i} \nabla_{x_i} J_i(x_i(k), x_{-i}(k))), \quad (3)$$

where $\alpha_{k,i} > 0$ is a step-size. This can be modeled in continuous-time as

$$\Sigma_i : \dot{x}_i = \Pi_{\Omega_i}(x_i, -\nabla_{x_i} J_i(x_i, x_{-i})), \quad (4)$$

where $\dot{x}_i$ denotes the time derivative of x_i and $\Pi_{\Omega_i}(x_i, v) = \lim_{\delta \rightarrow 0} \frac{P_{\Omega_i}(x_i + \delta v) - x_i}{\delta}$ denotes the orthogonal projection of a vector v onto the tangent cone of Ω_i at x_i , hence a projected dynamical system (Nagurney and Zhang 1996). The overall interconnected dynamical system obtained from all agents' dynamics Σ_i is denoted by $\Sigma = (\Sigma_i, \Sigma_{-i})$ and is given in stacked notation as

$$\Sigma : \dot{x} = \Pi_{\Omega}(x, -F(x)), \quad (5)$$

Thus, the overall projected-gradient dynamics is a feedback interconnection between a bank of integrators and a static map involving the pseudo-gradient map F . Either the discrete-time scheme (3) with diminishing/constant step-sizes (Facchinei and Pang 2007) or the continuous-time one (4) (Arrow et al. 1951) (Flåm and Morgan 2004) converge globally to the NE when F is Lipschitz and strictly/strongly monotone.

Assumption 1 (i) F is strictly monotone: $(x - y)^T (F(x) - F(y)) > 0, \forall x, y \in \Omega, x \neq y$.
(ii) F is μ -strongly monotone: $(x - y)^T (F(x) - F(y)) \geq \mu \|x - y\|^2, \forall x, y \in \Omega, \mu > 0$.

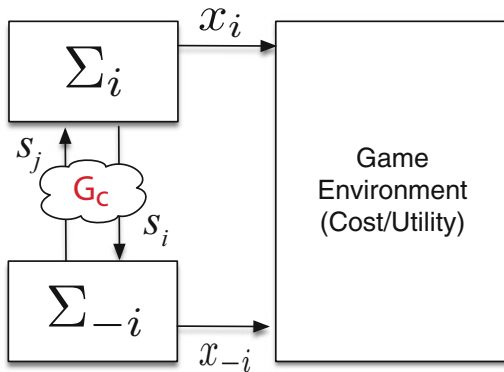
Note that in (3) or (4), the knowledge required by each player i at every time is its own cost J_i and all the others' actions x_{-i} , i.e., *perfect-decision information*. This is also the case when this perfect-decision information is obtained via broadcast from a *central control node or coordinator*, (Grammatico 2017; Paccagnan et al. 2019). In this entry, we discuss NE seeking over networks under *partial-decision information* setting, where agents exchange information only locally with their neighbors over a network of arbitrary topology, in a *center-free (distributed)* manner.

NE Seeking over Networks

We next introduce the *partial-decision information* setting over networks. Consider that each player i does not know the actions of all other players x_{-i} (even though this is needed in order

to make a decision) and there is no central node to disseminate this information. To compensate for the lack of global information, players communicate locally with their neighbors. Consider that the information sharing between players is described by a communication graph $G_c = (\mathcal{N}, E)$ of arbitrary topology, undirected and connected, where E denotes the edge set. Let $\mathcal{N}_i = \{j \in \mathcal{N} | (j, i) \in E\}$ denote the set of neighbors of agent i in G_c . Each player i uses local feedback from its neighbors to learn the decisions/actions of all others, i.e., to estimate x_{-i} . For example, in Fig. 1, the feedback signal s_j from neighbor $j \in \mathcal{N}_i$ can be its decision x_j and/or its signaling/state variables. Thus, assume that player i maintains its own estimate $\mathbf{x}_{-i}^i$ of x_{-i} and updates its action based on this estimate. Hence, its state $\mathbf{x}^i = (x_i, \mathbf{x}_{-i}^i)$ encompasses its own decision/action and its estimate of the others' decisions.

Unlike the classical setting of perfect information, in a partial-decision information setting for game-theoretic decision-making, each agent runs two dynamical processes: one for the estimation of the others' decisions/actions and another one for the update of its own action/decision. These two processes depend on one another and depend also on what the other players are doing. This interdependence and dynamic coupling makes it difficult to analyze their convergence.



Nash Equilibrium Seeking over Networks, Fig. 1 NE seeking with partial information over network G_c

Two Time-Scale Separation or Diminishing Step-Sizes

The simplest way to handle the two dynamical processes (decision estimation and action update) is to decouple them via two time-scale separation, either via diminishing step-sizes or via high-gains. In a discrete-time algorithm proposed by Salehisadaghiani and Pavel (2016), the estimates are updated via an asynchronous gossip-based iteration, and the actions are updated via projected-gradient using these estimates. The algorithm generalizes the aggregative game algorithm of Koshal et al. (2016) to generally coupled games. At each iteration k , one randomly selected player i_k communicates with a neighbor $j_k \in \mathcal{N}_{i_k}$; they average their estimates and update their actions as in

$$\begin{aligned} \Sigma_i : \mathbf{x}_{-i}^i(k) &= \frac{\mathbf{x}_{-i}^i(k) + \mathbf{x}_{-i}^j(k)}{2}, \\ &\forall i, j \in \{i_k, j_k\} \\ x_i(k+1) &= P_{\Omega_i} \left(x_i(k) - \alpha_{k,i} \nabla_i J_{x_i}(x_i(k), \right. \\ &\quad \left. \mathbf{x}_{-i}^i(k) \right) \end{aligned} \quad (6)$$

with $\mathbf{x}_i^i(k) = x_i(k)$, where $\alpha_{k,i}$ are diminishing step-sizes, such that $\sum_{k=1}^{\infty} \alpha_{k,i}^2 < \infty$, $\sum_{k=1}^{\infty} \alpha_{k,i} = \infty$. Under such diminishing step-sizes, the algorithm behaves as if on a two-time scale, with all estimates converging quickly to consensus, which in turn allows to show that players' actions converge almost surely to the NE of the game (cf. Theorem 2 in Salehisadaghiani and Pavel 2016 or Proposition 3 in Koshal et al. 2016). However, with constant step-sizes, gossip-based projected-gradient algorithms converge only to a neighborhood of NE.

In a continuous-time version, each individual player has dynamics Σ_i given as

$$\Sigma_i : \begin{bmatrix} \dot{x}_i \\ \epsilon \dot{\mathbf{x}}_{-i}^i \end{bmatrix} = \begin{bmatrix} \Pi_{\Omega_i} \left(x_i, -\nabla_{x_i} J_i(x_i, \mathbf{x}_{-i}^i) \right) \\ - \sum_{j \in \mathcal{N}_i} (\mathbf{x}_{-i}^i - \mathbf{x}_{-i}^j) \end{bmatrix} \quad (7)$$

where $\epsilon > 0$ is a small parameter. For games with unconstrained action sets, this is in the standard form of a singularly perturbed system (Khalil 2002), where the estimate dynamics is the fast subsystem (with high-gain $\frac{1}{\epsilon}$) and the projected-gradient dynamics is the slow subsystem, respectively. The quasi-steady state of the fast subsystem is on the estimate-consensus subspace, and, for small enough ϵ , convergence to the NE can be shown when the pseudo-gradient is strictly/strongly monotone (see Theorem 3 in Gadjov and Pavel 2019 or Theorem 2 in Ye and Hu 2017). Thus, continuous-time dynamics or discrete-time schemes that simply combine consensus dynamics with projected-gradient operate as if on a two-times scale. In order to achieve exact convergence, it is critical that the estimate disagreement is quickly decaying (requiring diminishing step-sizes or small ϵ), while when using constant step-sizes, these algorithms can only guarantee convergence to a neighborhood of the NE.

Single-Time-Scale Dynamics

Under time-scale separation of the two dynamical processes (decision estimation and action update), convergence is easier to establish, but such schemes/dynamics have generally slow convergence. Next, we discuss the more difficult case when the two processes operate on a single-time scale, i.e., when agents estimate the others' decisions and simultaneously update their own action. This amounts to agents having to track a moving target (which affects their own action and also depends on it) while also acting. Essentially players perform simultaneous player-by-player optimization and consensus of estimates. One such dynamics proposed by Gadjov and Pavel (2019) achieves convergence on a single-time scale by employing a consensus correction term on the action update. In this case, each player dynamics Σ_i is given as

$$\Sigma_i : \begin{bmatrix} \dot{x}_i \\ \dot{\mathbf{x}}_{-i}^i \end{bmatrix} = \begin{bmatrix} \Pi_{\Omega_i} \left(x_i, -\nabla_{x_i} J_i(x_i, \mathbf{x}_{-i}^i) - \sum_{j \in \mathcal{N}_i} (x_i - \mathbf{x}_i^j) \right) \\ - \sum_{j \in \mathcal{N}_i} (\mathbf{x}_{-i}^i - \mathbf{x}_{-i}^j) \end{bmatrix} \quad (8)$$

N

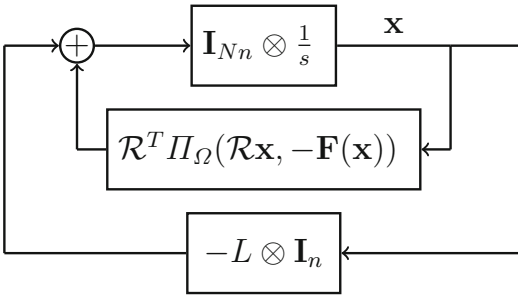
This is a projected-gradient with consensus correction based on local feedback. The correction term is inspired by the multi-agent agreement/consensus problem (Bürger and De Persis 2015) and arises naturally from a passivity-based design. With $\mathbf{x}^i = (x_i, \mathbf{x}_{-i}^i)$ and $\mathbf{x} = (\mathbf{x}^i)_{i \in \mathcal{N}}$, the overall interconnected dynamics $\Sigma = (\Sigma_i, \Sigma_{-i})$ are written in stacked form on the augmented space of decisions and estimates as

$$\Sigma : \quad \dot{\mathbf{x}} = \mathcal{R}^T \Pi_{\Omega}(\mathcal{R}\mathbf{x}, -\mathbf{F}(\mathbf{x}) - \mathcal{R}\mathbf{L}\mathbf{x}) - \mathcal{S}^T \mathbf{S}\mathbf{L}\mathbf{x}$$

where $\mathbf{F}(\mathbf{x}) := (\nabla_{x_i} J_i(\mathbf{x}^i))_{i \in \mathcal{V}} : \Omega^N \rightarrow \mathbb{R}^n$ is the *extended* pseudo-gradient of the game, $\mathbf{L} = L \otimes \mathbf{I}_n$, L is the Laplacian of G_c , and $\mathcal{R}, \mathcal{S}$ are matrices used to select actions and estimates from $\mathbf{x}$, e.g., $x = \mathcal{R}\mathbf{x}$. The overall Σ is represented as the feedback interconnected system in Fig. 2,

where the correction term appears as an extra feedback loop via the Laplacian of G_c .

Convergence to the NE can be shown by exploiting incremental passivity properties (Pavlov and Marconi 2008) and the fact that the Laplacian excess passivity can be used to balance the other game-dependent terms. Under monotonicity (passivity) of the extended pseudo-gradient $\mathbf{F}$, single-time-scale convergence to the NE holds over any connected communication graph G_c , cf. Theorem 4 in Gadjov and Pavel (2019). When monotonicity of $\mathbf{F}$ does not hold, single-time-scale convergence can be guaranteed over *any sufficiently connected* graph G_c , under just Lipschitz continuity of $\mathbf{F}$. The idea is to decompose the augmented space into the consensus subspace, where monotonicity holds by Assumption 1, and its orthogonal complement. Then, one can balance the shortage



Nash Equilibrium Seeking over Networks, Fig. 2
Block diagram of $\Sigma = (\Sigma_i, \Sigma_{-i})$

of passivity of $\mathbf{F}$ off the consensus subspace by excess passivity (strong monotonicity) of L on its complement, cf. Theorem 5 in Gadjov and Pavel (2019). Thus, a passivity-based approach highlights the trade-off between game properties and communication graph properties in Nash equilibrium seeking over networks. A discrete-time variant developed by Salehisadaghiani et al. (2019), based on an ADMM approach, has a similar correction term on the action update. Such a term seems to be critical in order to guarantee exact convergence on a single-time scale under constant step-sizes.

Comparison to Distributed Optimization and Multi-agent Agreement

Distributed NE seeking over networks is related to distributed optimization, a problem extensively studied over the past decade, e.g., Nedic and Ozdaglar (2010). However, there are differences between the two settings. In distributed optimization, optimality is in terms of an aggregate convex objective function; each agent optimizes a part of this aggregate objective (cooperate) and has complete authority over its full argument (decoupled problems). Because of this and the summable cost structure, even when the cost is not separable, one can extend the problem to an augmented space of estimates, where it becomes convex and separable, thus easily leading to distributed algorithms. In contrast, in a game setting, there are multiple self-interested decision-makers, each with their own individual objective function which is partially convex only. Furthermore, each objective depends on the others'

decisions (coupled problems), but each agent has authority only over its own actions. This translates into challenges when developing distributed Nash equilibrium seeking algorithms over networks. The major technical difficulty is due to the fact that one cannot guarantee the preservation of pseudo-gradient monotonicity when extended to the augmented space of actions and estimates. As a result, discrete-time distributed convergent schemes with fixed step-sizes (or single-time-scale continuous ones) are harder to develop than their optimization counterparts.

On the other hand, the design of NE seeking schemes over networks is consistent with standard multi-agent consensus design, e.g., Olfati-Saber et al. (2007), except that here consensus is on the decision estimates and agents' dynamics are not integrators or decoupled identical LTI MIMO systems as in typical consensus literature. Rather, the agent dynamics are nonlinear and heterogeneous (e.g., projected-gradient dynamics) similar to multi-agent agreement for general nonlinear systems (Bürger and De Persis 2015). However, unlike (Bürger and De Persis 2015), in a game agents do not have individually passive dynamics, but rather, only partially incrementally passive dynamics, which are coupled to the other players' dynamics. As such, results from the multi-agent agreement literature cannot be directly used. Rather, the analysis has to rely on the incremental passivity (monotonicity) properties of the pseudo-gradient and of the overall system of all agents' interconnected dynamics (Pavel 2019b).

Summary and Future Directions

Nash equilibrium seeking over networks is a recent research direction with many open problems. There is currently a flurry of activity in this area. Recent efforts are focused on how to accelerate convergence (Tatarenko et al. 2018), how to generalize the setting to games with coupled constraints for generalized Nash equilibrium (GNE) seeking (Liang et al. 2017; Yi and Pavel 2019a; Pavel 2019a), how to relax game or communication graph properties

(Yi and Pavel 2019b; Ye and Hu 2018), or how to extend Nash equilibrium seeking to dynamical processes and networks (De Persis and Monshizadeh 2019; Romano and Pavel 2019; De Persis and Grammatico 2018). As of now, there is no method yet that can achieve distributed single-time-scale convergence in (G)NE seeking over networks under partial-decision information for (not strictly/strongly) monotone games. Operator-theoretic splitting methods seem to be particularly useful as they provide elegant, concise convergence proofs (Bauschke and Combettes 2011), (Belgioioso and Grammatico 2018), but distributing them proves to be challenging. Also open is how to develop efficient algorithms/dynamics that exploit the structure of the agents' strategic interaction (e.g., in aggregative games, or at the other end of the spectrum, in games with sparse interference graph), how to deal with delayed/asynchronous information and improve agents' privacy, or how to design incentives against cheating so that agents truthfully share some information.

Cross-References

- [Distributed Optimization](#)
- [Information-Based Multi-agent Systems](#)
- [Learning in Games](#)
- [Nash Equilibrium](#)

Bibliography

- Arrow K, Hurwitz L, Uzawa H (1951) A gradient method for approximating saddle points and constrained maxima. Rand Corporation, United States Army Air Forces
- Başar T, Olsder G (1999) Dynamic noncooperative game theory: second edition. Classics in applied mathematics. SIAM
- Bauschke HH, Combettes PL (2011) Convex analysis and monotone operator theory in Hilbert spaces, 1st edn. Springer Publishing Company, Incorporated
- Bauso D (2016) Game theory with engineering applications. Advances in design and control series. SIAM
- Belgioioso G, Grammatico S (2018) A Douglas-Rachford splitting for semi-decentralized equilibrium seeking in generalized aggregative games. In: Proceedings of the IEEE CDC-ECC, pp 3541–3546
- Bürger M, De Persis C (2015) Dynamic coupling design for nonlinear output agreement and time-varying flow control. *Automatica* 51:210–222
- Cheng N, Zhang N, Lu N, Shen X, Mark JW, Liu F (2014) Opportunistic spectrum access for CR-VANETs: a game-theoretic approach. *IEEE Trans Veh Technol* 63(1):237–250
- De Persis C, Grammatico S (2018) Distributed averaging integral Nash equilibrium seeking on networks. ArXiv e-prints 1812.00445
- De Persis C, Monshizadeh N (2019) A feedback control algorithm to steer networks to a Cournot-Nash equilibrium. *IEEE Trans Control Netw Syst*. <https://doi.org/10.1109/TCNS.2019.2897907>
- Facchinei F, Pang J (2007) Finite-dimensional variational inequalities and complementarity problems. Springer, New York
- Flâm S, Morgan J (2004) Newtonian mechanics and nash play. *Int Game Theory Rev* 6(2):181–194
- Frihauf P, Krstic M, Başar T (2012) Nash equilibrium seeking in noncooperative games. *IEEE Trans Autom Control* 57(5):1192–1207
- Fudenberg D, Levine DK (1998) The theory of learning in games. The MIT Press, Cambridge
- Gadjov D, Pavel L (2019) A passivity-based approach to Nash equilibrium seeking over networks. *IEEE Trans Autom Control* 64(3):1077–1092
- Gharesifard B, Cortes J (2013) Distributed convergence to Nash equilibria in two-network zero-sum games. *Automatica* 49(6):1683–1692
- Grammatico S (2017) Dynamic control of agents playing aggregative games with coupling constraints. *IEEE Trans Autom Control* 62(9):4537–4548
- Khalil H (2002) Nonlinear systems. Prentice Hall
- Koshal J, Nedic A, Shanbhag UV (2016) Distributed algorithms for aggregative games on graphs. *Oper Res* 64(3):680–704
- Li S, Başar T (1987) Distributed algorithms for the computation of noncooperative equilibria. *Automatica* 23(4):523–533
- Li N, Marden JR (2013) Designing games for distributed optimization. *IEEE J Sel Top Signal Process* 7(2):230–242
- Liang S, Yi P, Hong Y (2017) Distributed Nash equilibrium seeking for aggregative games with coupled constraints. *Automatica* 85:179–185
- Marden JR, Shamma JS (2013) Game theory and distributed control. *Handb Game Theory* 4:2818–2833. Elsevier Science
- Nagurney A, Zhang D (1996) Projected dynamical systems and variational inequalities with applications. Springer
- Nash J (1950) Equilibrium points in N-person games. *Proc Natl Acad Sci* 36(1):48–49
- Nedic A, Ozdaglar A (2010) Cooperative distributed multi-agent optimization. In: Eldar Y, Palomar D (eds) *Convex optimization in signal processing and communications*. Cambridge University Press, pp 340–386

- Olfati-Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. *Proc IEEE* 95(1):215–233
- Paccagnan D, Gentile B, Parise F, Kamgarpour M, Lygeros J (2019) Nash and Wardrop equilibria in aggregative games with coupling constraints. *IEEE Trans Autom Control* 64(4):1373–1388
- Pavel L (2012) Game theory for control of optical networks. Birkhäuser-Springer
- Pavel L (2019a) Distributed GNE seeking under partial-decision information over networks via a doubly-augmented operator splitting approach. *IEEE Trans Autom Control*. Accepted
- Pavel L (2019b) On incremental passivity in network games. In: Walrand J et al (eds) *Network games, control, and optimization, static & dynamic game theory: foundations & applications*. Birkhauser, Cham, pp 183–200
- Pavlov A, Marconi L (2008) Incremental passivity and output regulation. *Syst Control Lett* 57:400–409
- Romano AR, Pavel L (2019) Dynamic NE seeking for multi-integrator networked agents with disturbance rejection. *IEEE Trans Control Netw Syst*. <https://doi.org/10.1109/TCNS.2019.2920590>
- Salehisadaghiani F, Pavel L (2016) Distributed Nash equilibrium seeking: a gossip-based algorithm. *Automatica* 72:209–216
- Salehisadaghiani F, Shi W, Pavel L (2019) Distributed Nash equilibrium seeking under partial-decision information via the alternating direction method of multipliers. *Automatica* 103:27–35
- Scutari G, Facchinei F, Pang JS, Palomar DP (2014) Real and complex monotone communication games. *IEEE Trans Inform Theory* 60(7):4197–4231
- Shamma JS, Arslan G (2005) Dynamic fictitious play, dynamic gradient play, and distributed convergence to Nash equilibria. *IEEE Trans Autom Control* 50(3):312–327
- Stankovic MS, Johansson KH, Stipanovic DM (2012) Distributed seeking of Nash equilibria with applications to mobile sensor networks. *IEEE Trans Autom Control* 57(4):904–919
- Swenson B, Kar S, Xavier J (2015) Empirical centroid fictitious play: an approach for distributed learning in multi-agent games. *IEEE Trans Signal Process* 63(15):3888–3901
- Tatarenko T, Shi W, Nedic A (2018) Accelerated gradient play algorithm for distributed Nash equilibrium seeking. In: *Proceedings of the 57th IEEE conference on decision and control (CDC)*, pp 3561–3566
- Wang X, Xiao N, Wongpiromsarn T, Xie L, Frazzoli E, Rus D (2013) Distributed consensus in noncooperative congestion games: an application to road pricing. In: *IEEE 10th international conference on control and automation (ICCA)*, pp 1668–1673
- Ye M, Hu G (2017) Distributed Nash equilibrium seeking by a consensus based approach. *IEEE Trans Autom Control* 62(9):4811–4818
- Ye M, Hu G (2018) Distributed Nash equilibrium seeking in multiagent games under switching communication topologies. *IEEE Trans Cybern* 48(11):3208–3217
- Yi P, Pavel L (2019a) An operator splitting approach for distributed generalized Nash equilibria computation. *Automatica* 102:111–121
- Yi P, Pavel L (2019b) Distributed generalized Nash equilibria computation of monotone games via double-layer preconditioned proximal-point algorithms. *IEEE Trans Control Netw Syst* 6(1):299–311
- Yin H, Shanbhag U, Mehta PG (2011) Nash equilibrium problems with scaled congestion costs and shared constraints. *IEEE Trans Autom Control* 56(7):1702–1708

Network Games

R. Srikant

Department of Electrical and Computer Engineering and the Coordinated Science Lab, University of Illinois at Urbana-Champaign, Champaign, IL, USA

Abstract

Game theory plays a central role in studying systems with a number of interacting players competing for a common resource. A communication network serves as a prototypical example of such a system, where the common resource is the network, consisting of nodes and links with limited capacities, and the players are the computers, web servers, and other end hosts who want to transfer information over the shared network. In this entry, we present several examples of game-theoretic interaction in communication networks and a simple mathematical model to study one such instance, namely, resource allocation in the Internet.

Keywords

Congestion games · Network economics · Price-taking users · Routing games · Strategic users

Introduction

A communication network can be viewed as a collection of resources shared by a set of competing users. If the network were totally unregulated,

then each user would attempt to grab as many resources in the network as possible, resulting in poor network performance, a situation commonly referred to as the *tragedy of the commons* (Hardin 1968). In reality, there is a carefully designed set of network protocols and pricing mechanisms which provide incentives to users to act in a socially responsible manner. Since game theory is the mathematical discipline which studies the interactions between selfish users, it is a natural tool to use to design these network control mechanisms. We now provide a few examples of network problems which naturally lend themselves to game-theoretic analysis. Later, we will elaborate on the game-theoretic formulation of one of these examples.

- *Resource Allocation:* A network such as the Internet is a collection of links, where each link has a limited data-carrying capacity, usually measured in bits per second. The Internet is shared by billions of users, and the actions of these users have to be regulated so that they share the resources in the network in a fair manner. Equivalently, this problem can be viewed as one in which a network designer has to design a collection of protocols so that the users of the network can equitably allocate the available resources among themselves without the intervention of a central authority. Such protocols are built into every computer connected to the Internet today, to allow for seamless operation of the network. The problem of designing such protocols can be posed as a game-theoretic problem in which the players are the network and the traffic sources using the network (Kelly 1997).
 - *Routing Games:* Finding appropriate routes for each user's data traffic is a particular form of resource allocation mentioned above. However, routing has applications beyond communication networks (with the other major application area being transportation networks), so it is useful to discuss routing separately. In communication networks, each user may attempt to find the minimum-delay route for its traffic, with help from the network, to minimize the delay experienced by its packets.
- In a transportation network, each automobile on the road attempts to take the path of least congestion through the network. An active area of research in game theory is one which tries to understand the impact of individual user decisions on the global performance of the network (Roughgarden 2005). An interesting result in this regard is the *Braess paradox* which is an example of a road transportation network in which the addition of a road leads to increased delays when each user selfishly choose a route to minimize its delay. Of course, if routes are chosen to minimize the overall delay experienced in the network, such a paradox will not arise.
- *Peer-to-Peer Applications:* Many studies have indicated that file sharing between users (also known as peers) directly, without using a centralized web site such as YouTube, is a dominant source of traffic in the Internet. For such a peer-to-peer service to work, each peer should not only download files from others but should also be willing to sacrifice some of its resources to upload files to others. Naturally, peers would prefer to only download and not upload to minimize their resource usage. The design of incentive schemes to induce users to both download and upload files is another example of a game-theoretic problem in a network (Qiu and Srikant 2004).
 - *Network Economics:* In addition to end-user interaction, Internet service providers (ISPs) have to interact with each other to allow their customers access to all the web sites in the world. For example, one ISP may have a customer who wants to access a web site connected to another ISP. In this case, the data traffic must cross ISP boundaries, and thus, one ISP has to transport data destined for a customer of another ISP. Thus, ISPs must be willing to contribute resources to satisfy the needs of customers who do not directly pay them. In such a situation, ISPs must have bilateral agreements (commonly known as *peering agreements*) to ensure that the selfish interest of each ISP to minimize its resource usage is aligned with the needs of its customers. Again, game theory is the right tool to study

such inter-ISP interactions (Courcoubetis and Weber 2003).

- *Spectrum Sharing*: Large portions of the radio spectrum are severely underutilized. Typically, portions of the spectrum are assigned to a primary user, but the primary user does not use it most of the time. There has been a surge of interest recently in the concept of *cognitive radio*, whereby radios are cognitive of the presence or absence of the primary user, and when the primary user is absent, another radio can use the spectrum to transmit its data. When there are many users and the available spectrum is split into many channels, it is impossible for users to perfectly coordinate their transmissions to achieve maximum network utilization. In these situations, game-theoretic protocols which take into account the noncooperative behavior of the users can be designed to allow secondary users to access the available channels as efficiently as possible (Saad et al. 2009).

In the next section, we will elaborate on one of the applications above, namely, resource allocation in the Internet, and show how game-theoretic modeling can be used to design fair resource sharing.

Resource Allocation and Game Theory

Consider a network consisting of L links, with link l having capacity c_l . Suppose that there are R users sharing the network, with each user r being characterized by a set of links which connect the user's source to its destination. Since each user uses a fixed route in our model, we will use r to denote both the user and the route used by the user. We use the notation $l \in r$ to denote that link l is a part of route r . Let x_r denote the rate at which user r transmits data. Thus, we have the following natural constraints, which state that the total data rate on any link must be less than or equal to the capacity of the link:

$$\sum_{r:l \in r} x_r \leq c_l, \quad \forall l. \quad (1)$$

Associated with each user is a concave utility function $U_r(x_r)$ which is the utility that user r derives by transmitting data at rate x_r . The network utility maximization problem is to solve

$$\max_{x \geq 0} \sum_r U_r(x_r), \quad (2)$$

subject to the constraint (1). In (2), x denotes the vector $(x_1, x_2, \dots, x_R)$, and $x \geq 0$ means that each component of x must be greater than or equal to zero. Note that the goal of the network in (2) is to maximize the sum of the utilities of the users in the network.

Let p_l be the Lagrange multiplier corresponding to the capacity constraint in (1) for link l . Then the Lagrangian for the problem is given by

$$L(x, p) = \sum_r U_r(x_r) - \sum_l p_l (y_l - c_l), \quad (3)$$

where we have used the notation $y_l := \sum_{r:l \in r} x_r$ to denote the total data rate on link l . If p is known, then the optimal x can be calculated by solving

$$\max_{x \geq 0} L(x, p).$$

Notice that the optimal solution for each x_r can be obtained by solving

$$\max_{x_r \geq 0} U_r(x_r) - q_r x_r, \quad (4)$$

where $q_r = \sum_{l \in r} p_l$. Thus, if the Lagrange multipliers are known, then the network utility maximization can be interpreted as a game in the following manner. Suppose that the network charges each user q_r dollars for every bit transmitted by user r though the network. Then, $q_r x_r$ is the dollar per second spent by the user if x_r is measured in bits per second. Interpreting $U_r(x_r)$ as the dollars per second that the user is willing to pay for transmitting at rate x_r , the optimization problem in (4) is the problem faced by user r which wants to maximize its net utility, i.e., utility minus cost. Thus, the individual optimal

solution for each user is also the solution to the network utility maximization problem. The above game-theoretic interpretation of the network utility maximization problem is somewhat trivial since, given the p_l 's or q_r 's, there is no interaction between the users. Of course, this interpretation relies on the ability of the network to compute p . We next present a scheme to compute p , which couples the users closely and thus allows for a richer game-theoretic interpretation.

Suppose that the network wants to compute p but does not have access to the utility functions of the users. The network asks each user r to bid an amount w_r which is interpreted as the dollars per second that the user is willing to pay. The network then assumes that user r 's utility function is $w_r \log x_r$ and solves the network utility maximization. While this choice of utility function may seem arbitrary, the resulting solution x has a number of attractive properties, including a form of fairness called *proportional fairness*. The proportionally fair resource allocation solution to (4) is given by

$$\frac{w_r}{x_r} = q_r. \quad (5)$$

The network then allocates rate x_r to user r and charges q_r dollars per bit. From (5), the amount charged to user r per second is w_r , thus satisfying the original interpretation of w_r . Knowing that the network charges users in this manner, how might a user choose its bid w_r ? Recall that user r 's goal is to solve (4). Substituting from (5), the problem in (4) can be rewritten as

$$\max_{w_r \geq 0} U_r \left(\frac{w_r}{q_r} \right) - w_r. \quad (6)$$

Thus, the users' problem of selecting w can be viewed as a game, with each user's objective given by (6). Note that q_r is given by (5) and thus depends on all the w_r 's. Depending upon the application, the game can be solved under one of two assumptions:

- *Price-Taking Users:* Under this assumption, users are assumed to take the price q_r as

given, i.e., they do not attempt to infer the impact of their actions on the price. This is a reasonable assumption in a large network such as the Internet, where the impact of a single user on the link prices is negligible, and it is practically impossible for any user to infer the impact of its decisions on the prevailing price of the network resources. When the users are price taking, the socially optimal solution, i.e., the solution to the network utility maximization problem, coincides with the Nash equilibrium of the game. To see this, note that the solution to (6) is given by

$$\frac{1}{q_r} U'_r \left(\frac{w_r}{q_r} \right) - 1 = 0,$$

under the assumption that the utility function is differentiable and the solution is bounded away from zero. Using (5), this equation reduces to

$$U'_r(x_r) = q_r,$$

which maximizes the Lagrangian (3). It is not difficult to see that the complementary slackness equations in the Karush-Kuhn-Tucker conditions are satisfied since the constraints for (2) and the proportionally fair solution are the same. Thus, under the price-taking assumption, the equilibrium of the game solution is the same as the socially optimal solution provided the network computes q_r using the proportionally fair resource allocation formulation.

- *Strategic Users:* In networks where the number of users is small, it may be possible for each user to know the topology of the network, and thus each user may be able to solve for the proportionally fair resource allocation if it has access to other users' bids. In other words, it may be possible to compute a Nash equilibrium by taking into account the impact of the w_r 's on the q_r 's. When the users are strategic, the socially optimal solution could be quite different from the Nash equilibrium. The ratio of the network utility under the socially optimal solution to the network utility

under a Nash equilibrium is called the *price of anarchy*.

There is a rich literature associated with both interpretations of the network congestion game. In the case of price-taking users, much of the emphasis in the literature has been on designing distributed algorithms to achieve the socially optimal solution (Shakkottai and Srikant 2007). In the case of strategic users, the focus has been on characterizing the price of anarchy (Johari and Tsitsiklis 2004; Yang and Hajek 2007).

Summary and Future Directions

We have presented a number of applications which involve the interactions of selfish users over a network. For the resource allocation application, we have also described how simple mathematical models can be used to provide incentives for users to act in a socially optimal manner. In particular, we have shown that, under the reasonable price-taking assumption and an appropriate computation of link prices, selfish users automatically maximize network utility. In the case where the users are strategic, the goal is to characterize the price of anarchy.

Moving forward, two areas which require considerable further research are the following: (i) inter-ISP routing and (ii) spectrum sharing. The Internet is a fairly reliably network, and any unreliability often arises due to routing issues among ISPs. As mentioned in the introduction, peering arrangements between ISPs are necessary to make sure that ISPs carry each others' traffic and are appropriately compensated for it, either through reciprocal traffic-carrying agreements or actual monetary transfer. Thus, the policy that an ISP uses to route traffic may be governed by these peering agreements. The more complicated these policies are, the more chances there are for routing misconfigurations that lead to service interruptions. This interplay between policies and technology in the form of routing algorithms is an interesting topic for further study.

Cognitive radios and spectrum sharing are expected to be significant technological components of future wireless networks. Designing algorithms for selfish radios to share the available spectrum while respecting the rights of the primary user of the spectrum is a challenge that requires considerable further attention. This area of research requires one to combine sensing technologies to sense the presence of other users with game-theoretic models to ensure fair channel access to the secondary users, subject to the constraint that the primary user should not be affected by the presence of the secondary users.

Cross-References

- ▶ [Game Theory: A General Introduction and a Historical Overview](#)
- ▶ [Networked Systems](#)
- ▶ [Optimal Deployment and Spatial Coverage](#)

Bibliography

- Courcoubetis C, Weber R (2003) Pricing communication networks: economics, technology and modelling. Wiley, Hoboken
- Hardin G (1968) The tragedy of the commons. *Science* 162:1243–1248
- Johari R, Tsitsiklis JN (2004) Efficiency loss in a network resource allocation game. *Math Oper Res* 29:407–435
- Kelly FP (1997) Charging and rate control for elastic traffic. *Eur Trans Telecommun* 8:33–37
- Qiu D, Srikant R (2004) Modeling and performance analysis of BitTorrent-like peer-to-peer networks. *Proc ACM SIGCOMM ACM Comput Commun Rev* 34:367–378
- Roughgarden T (2005) Selfish routing and the price of anarchy. MIT Press, Cambridge
- Saad W, Han Z, Debbah M, Hjørungnes A, Basar T (2009) Coalitional game theory for communication networks: a tutorial. *IEEE Signal Process Mag* 26(5): 77–97
- Shakkottai S, Srikant R (2007) Network optimization and control. NoW Publishers, Boston-Delft
- Yang S, Hajek B (2007) VCG-Kelly mechanisms for allocation of divisible goods: adapting VCG mechanisms to one-dimensional signals. *IEEE J Sel Areas Commun* 25:1237–1243

Networked Control Systems: Architecture and Stability Issues

Linda Bushnell¹ and Hong Ye²

¹Department of Electrical and Computer Engineering, University of Washington, Seattle, WA, USA

²The Mathworks, Inc., Natick, MA, USA

Abstract

When shared, band-limited, real-time communication networks are employed in a control system to exchange information between spatially distributed components, such as controllers, actuators, and sensors, it is categorized as a networked control system (NCS). The primary advantages of a NCS are reduced complexity and wiring, reduced design and implementation cost, ease of system maintenance and modification, and efficient data sharing. In addition, this unique architecture creates a way to connect the cyberspace to the physical space for remote operation of systems. The NCS architecture allows for performing more complex tasks but also requires taking the network effects into account when designing control laws and stability conditions. In this entry, we review significant results on the architecture and stability analysis of a NCS. The results presented address communication network-induced challenges such as time delays, scheduling, and information packet dropouts.

Keywords

Band-limited channels · Network-induced delays · Control network scheduler

Introduction

From the washing machine, air conditioner, and microwave oven to the telephone, stereo, and automobile, embedded computers are present in

the modern home. In a factory environment, there are thousands of networked smart sensors and actuators with embedded processors, working to complete a coordinated task. The trend in manufacturing plants, homes, buildings, aircraft, and automobiles is toward distributed networking. This trend can be inferred from many proposed or emerging network standards, such as controller area network (CAN) for automotive and industrial automation, BACnet for building automation, PROFIBUS and WorldFIP fieldbus for process control, and IEEE 802.11 and Bluetooth wireless standards for applications such as mobile sensor networks, HVAC systems, and unmanned aerial vehicles.

The traditional dedicated point-to-point wired connection in control systems has been successfully implemented in industry for decades. With the advance of communication network and hardware technologies, it is common to integrate the communication network into the control system to replace the dedicated point-to-point connection to achieve reduced weight and power, lower cost, simpler installation and maintenance, and higher reliability, to name a few advantages. For example, a typical new automobile has two controller area networks (CANs): a high-speed one in front of the firewall for the engine, transmission, and traction control and a low-speed one for locks, windows, and other devices (Johansson et al. 2005).

The conventional definition of a networked control system (NCS) is as follows: When a feedback control system is closed via a communication channel, which may be shared with other nodes outside the control system, then the control system is called a NCS. A NCS can also be described as a feedback control system where the control loops are closed through a real-time communication network.

Architecture of Networked Control Systems

The architecture of a NCS consists of a band-limited, digital communication network

physically and electronically integrated with a spatially distributed control system, operated on a given plant. Digital information, such as controller signals, actuator signals, sensor signals, and operator input, is transmitted via the network. The components connected by the network include all nodes of the control system, such as the supervisory (or “network owner”) computer, controller software and hardware, actuators, and sensors. In this structure, the feedback control system’s loops are closed over the shared communication network.

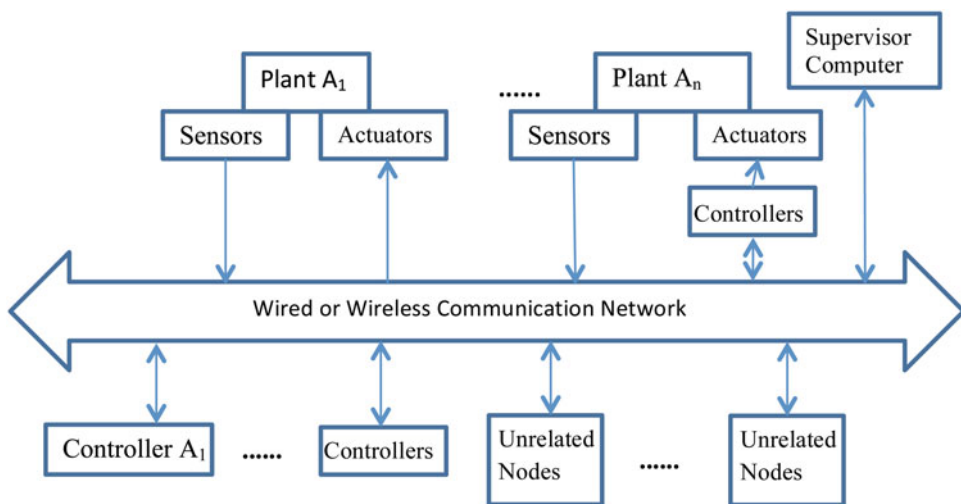
The communication network can be wired or wireless and may be shared with other unrelated nodes outside the control system. As illustrated in Fig. 1, the shared communication channel, which multiplexes signals from the sensors to the controllers and/or from the controllers to the actuators, serves many other uses besides control. Each of the system components directly connected to the network via a network interface is denoted a physical node. Besides the network interface, the sensors and actuator nodes are typically smart nodes with embedded microprocessors. Sometimes, the controller is colocated with the smart actuator. Several key issues make networked control systems distinct from traditional control systems (Hespanha et al. 2007; Yang 2006).

Band-Limited Channels

Bandwidth limitation of the shared communication channel requires that all nodes in the network must share (e.g., time sharing or frequency sharing, etc.) the common network resource without interfering with each other.

Sampling and Delays

In a NCS, the plant outputs are sampled by the sensors, which can convert continuous-time analog signals to digital signals; perform preprocessing, filtering, and encoding; and package the data signal so that it is ready for transmission. After winning the medium access control and being transmitted over the network, the package containing the sampled data signal arrives at the receiver side, which could be a controller or a smart actuator with a controller collocated with it. The receiver unpacks and decodes the signal. This process is quite different from the traditional periodic sampling in digital control. The overall delay between sampling and the eventual decoding of the transmitted packet at the receiver can be time varying and random due to both the network access delay (i.e., the time it takes for a shared network to accept the data) and the transmission delay (i.e., the time during which data are in transit inside the network). This also depends on the highly variable network



Networked Control Systems: Architecture and Stability Issues, Fig. 1 A general networked control system (NCS) architecture

conditions, such as congestion and channel quality. In some NCSs, the data transmitted are time stamped, which means that the receiver may have an estimate of the delay's duration and could take appropriate corrective action. Given the rapid advance of embedded computation and communication hardware technology today, the transmission delay in many embedded systems can be neglected when compared with the magnitude of network access delay.

Packet Dropouts

It is possible in a NCS that a packet may be lost while it is in transit through the network. The packet that contains important sampling data or control signals may drop occasionally due to transmission errors of the physical network link, message collision, or node failures, to name a few. Overflow in queue or buffer can lead to network congestion and package loss. Thus, the use of queues is not favored by NCSs in general. Packet dropouts also happen if the receiver discards outdated arrivals that have long delays. Most network protocols are equipped with transmission-retry mechanisms, such as TCP, that guarantee the eventual delivery of packets. These protocols, unfortunately, are not appropriate for a NCS since the retransmission of old sensor data or calculated control signals is generally not very useful when new, time-critical data are available. Using selected old data for estimation or prediction is an exception, where old data may be packaged with the new data in one packet. It is advantageous to discard the old, un-transmitted data and transmit a new packet if and when it becomes available. In this way, the controller always receives fresh data for its control calculation, and the actuator always executes the up-to-date command to control the plant.

Modeling Errors, Uncertainties, and Disturbances

In a distributed NCS, modeling errors, uncertainties, and disturbances always exist when using the mathematical model to describe the physical process. These factors may lead to a major impact on the overall system performance and cause

failure in fulfilling the desired objectives. Wang and Hovakimyan (2013) proposed a reference model-based architecture to decouple the design of controller and communication schemes. A reference model is introduced in each subsystem as a bridge to build the connection between the real system and an ideal model, free of uncertainties. The closeness between the real system and the reference model is associated only with plant uncertainties, and the difference between the reference model and the ideal model is only in the communication constraints.

Stability of Networked Control Systems

The stability of a control system is often extremely important and is generally a safety requirement. Examples include the control of rockets, robots, airplanes, automobiles, or ships. Instability in any one of these systems can result in an unimaginable accident and loss of life. The stability of a general dynamical system with no input can be described with the Lyapunov stability criteria, which is stated as follows: A linear system is stable if its impulse response approaches zero as time approaches infinity or if every bounded input produces a bounded output.

When sensors, controllers, and actuators are not colocated and use a shared network to communicate, the feedback loop of a NCS is closed over the network. Network-induced, variable delays and packet dropouts can degrade the performance of a NCS. For example, the NCS may have a longer settling time or bigger overshoot in the step response. Furthermore, the NCS may become unstable when delays and/or packet dropouts exceed a certain range. Designers choosing to use a NCS architecture, however, are motivated not by performance but by cost, maintenance, and reliability gains.

Band-Limited Channels

Inspired by Shannon's results on the maximum bit rate that a communication channel can carry reliably, a significant research effort has been devoted to the problem of determining the

minimum bit rate that is needed to stabilize a system through feedback over a finite capacity channel (Baillieul 1999; Nair and Evans 2000; Tatikonda and Mitter 2004; Wong and Brockett 1999; Baillieul and Antsaklis 2007). This has been solved exactly for linear plants, but only conservative results have been obtained for nonlinear plants. The data-rate theorem that quantifies a fundamental relationship between unstable physical systems and the rate at which information must be processed in order to stably control them was proved independently under a variety of assumptions. Minimum bit rate and quantization becomes especially important for networks designed to carry very small packets with little overhead, because encoding measurements or actuation signals with less bits can save network bandwidth.

Most of the NCS stability results presented here, however, are based on the observation that the channel can transmit a finite number of packets per unit of time (packet rate) and each packet can carry certain number of bits in the data field. The packets on a real-time control network typically are frequent and have small data segments compared to their headers. For example, a CAN II packet with a single 16-bit data sample has fixed 64 bits of overhead associated with identifier, control field, CRC, ACK field, and frame delimiter, resulting in 25% utilization, and this utilization can never exceed 50% (data field length is limited to 64 bits). Thus, the quantization effects imposed by the communication networks are generally ignored.

Network-Induced Delays

A significant number of results have attempted to characterize a maximum upper bound on the sampling or transmission interval for which stability of the NCS can be guaranteed. The upper bound is sometimes called the maximum allowable transfer interval (MATI) (Walsh 2001a). These results implicitly attempt to minimize the packet rate or schedule the traffic of the control network that is needed to stabilize a system through feedback. The general approach is to design the controller using established techniques, considering the network to be

transparent, and then to analyze the effect of the network on closed-loop system performance and stability.

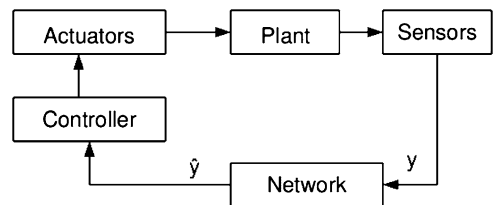
The NCS with a linear time-invariant (LTI) plant/controller pair and one-channel feedback (see Fig. 2) can be modeled by the following continuous-time system, where x includes the states of the plant and the controller, $x(t) = [x_p(t), x_c(t)]^T$:

$$\dot{x} = Ax + B\hat{y}, y = C(x) \quad (1)$$

$$\hat{y}(t) = \begin{cases} \hat{y}_{k-1}, & t \in [t_k, t_k + \tau_k) \\ \hat{y}_k, & t \in [t_k + \tau_k; t_{k+1}) \end{cases} \quad (2)$$

The signal y is a vector of sensor measurements, and $\hat{y}$ is the input to a continuous-time controller collocated with the actuators. Alternatively, $\hat{y}$ can be viewed as the input to the actuators and y as the desired control signal computed by a controller collocated with the sensors. The signal $y(t)$ is sampled at times $\{t_k : k \in N\}$ and the samples $y(k) := y(t_k)$ are sent through the network. But the samples arrive at the destination after a (possibly variable) delay of τ_k , where we assume that the network delays are always smaller than one sampling interval. For periodic sampling and constant delays, a sufficient and necessary condition for exponential stability of the NCS (Eqs. 1 and 2) was derived (Zhang et al. 2001). By using the augmented state space model and based on the stability of nonlinear hybrid systems, they also proved the sufficient condition for stability of the NCS in the time-invariant case.

If we now assume the sampling intervals are constant and the computation and transmission delays are negligible, then the variable network



Networked Control Systems: Architecture and Stability Issues, Fig. 2 A NCS architecture with one-channel feedback (controller collocated with actuator)

access delays serve as the main source of delays in a NCS (Lin et al. 2003, 2005). Using average dwell time results for discrete switched systems, Zhai et al. (2002) provided conditions such that NCS stability is guaranteed. Also, the authors consider robust disturbance attenuation analysis for this class of NCSs.

When the network delay is not constant or when the signal $y(t)$ is sampled in a nonperiodic fashion, the system (1) and (2) is not time invariant and one needs a Lyapunov-based argument to prove its stability. Zhang and Branicky (2001) derived the sufficient condition to ensure the NCS in Fig. 2 is exponentially stable. They also proposed a randomized algorithm to find the largest value of sampling interval for which stability can be guaranteed.

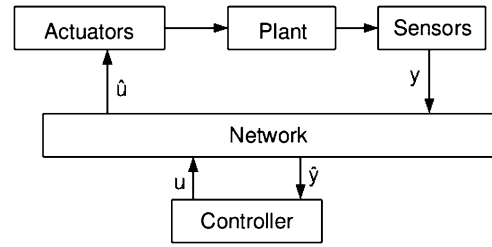
For a model-based NCS with state and output feedback, an explicit model of the plant is used to produce an estimate of the plant state behavior between transmission times (Montestruque and Antsaklis 2004). Sufficient conditions for Lyapunov stability are derived for a model-based NCS when the controller/actuator is updated with the sensor information at nonconstant time intervals. A NCS with transmission times that are driven by a stochastic process with identically independently distributed and Markov-chain-driven transmission times almost sure stability and mean-square sufficient conditions for stability are introduced. Onat et al. (2011) adapted above stability results to model-based predictive NCSs with realistic structure assumptions.

Control Network Scheduler

In general a multi-input/multi-output (MIMO) NCS with two-channel feedback, both the sampled plant output and controller output are transmitted via a network (see Fig. 3). Because of the network, only the reported output $y(t)$ is available to the controller and its prediction processes; similarly, only $\hat{u}(t)$ is available to the actuators on the plant. We label the network-induced error

$$e(t) := [\hat{y}(t), \hat{u}(t)]^T - [y(t), u(t)]^T$$

and the combined state of controller and plant $x(t) = [x_p(t), x_c(t)]^T$. The state of the entire



Networked Control Systems: Architecture and Stability Issues, Fig. 3 A NCS architecture with two-channel feedback

NCS is given by $z(t) = [x(t), e(t)]^T$. Following this general approach, the controller is designed using established techniques without considering the presence of the network.

The behavior of the network-induced error $e(t)$ is mainly determined by the architecture of the NCS and the scheduling strategy. In the special case of one-package transmission, there is only one node transmitting data on the network; therefore, the entire vector $e(t)$ is set to zero at each transmission time. For multiple nodes transmitting measured outputs $y(t)$ and/or computed inputs $u(t)$, the transmission order of the nodes depends on the scheduling strategy chosen for the NCS. In other words, the scheduling strategy decides which components of $e(t)$ are set to zero at the transmission times.

Static and dynamic schedulers (a.k.a. protocols) are two main categories used in a NCS. When the network resource or transmission order are pre-allocated or determined before runtime, it is called a static scheduler, such as round-robin scheduling. A dynamic scheduler determines the network allocation while the system runs. A novel dynamic network scheduler, try-once-discard (TOD), and several variations were introduced for wired and wireless NCSs (Walsh and Ye 2001; Ye et al. 2001). For linear and nonlinear NCSs with the new dynamic and commonly used static schedulers, an analytic proof of global exponential stability of a MIMO NCS was provided (Walsh 2001a; Walsh et al. 2001b). Simulation and experiment results showed that the dynamic schedulers outperform static schedulers

in terms of NCS performance, e.g., a bigger MATI.

Nesic and Teel (2004a, b) generalize the above results by considering a nonlinear NCS with external disturbances and more general class of protocols (or schedulers). They considered a new class of Lyapunov uniformly globally asymptotically stable (UGAS) protocols in a NCS. It is shown that if the controller is designed without taking into account the network, it yields input-to-state stability (ISS) with respect to external disturbances (not necessarily with respect to the network-induced error), and then the same controller will achieve semi-global practical ISS for the NCS when implemented via the network with a Lyapunov UGAS protocol. Moreover, the ISS gain is preserved. The adjustable parameter with respect to which semi-global practical ISS is achieved is the MATI between transmission times. The authors also studied the input–output L_p stability of a NCS for a large class of network scheduling protocols. It is shown that polling, static protocols, and dynamic protocol such as TOD belong to this class. Results in Nesic and Teel (2004a) provide a unifying framework for generating new scheduling protocols that preserve L_p stability properties of the system, if a design parameter is chosen to be sufficiently small. The most general version of these results can also be used to model a NCS with data packet dropouts. The proof technique used is based on the small gain theorem and lends itself to an easy interpretation.

A framework for analyzing the stability of a general nonlinear NCS with disturbances in the setting of L_p stability was provided by Tabbara et al. (2007). Their presentation provides sharper results for both gain and MATI than previously obtainable and details the property of uniformly persistently exciting scheduling protocols. This class of protocols was shown to lead to stability for high enough transmission rates. This was a natural property to demand, especially in the design of wireless scheduling protocols. The property is used directly in a novel proof technique based on the notions of vector comparison and (quasi)-monotone systems. Via simulations, analytical, and numerical comparison, it

is verified that the uniform persistence of excitation property of protocols is, in some sense, the “finest” property that can be extracted from wireless scheduling protocols.

Delays and Packet Dropouts

Packet dropouts can be modeled as either stochastic or deterministic phenomena. For a one-channel feedback NCS, Zhang and Branicky (2001) consider a deterministic dropouts model, with packet dropouts occurring at an asymptotic rate. Stability conditions were studied for a NCS with deterministic and stochastic dropouts (Seiler and Sengupta 2005).

Sometimes, the NCS was characterized as a continuous-time delayed differential equation (DDE) with the time-varying delay $\tau(t)$. One important advantage is that the equations are still valid even when the delays exceed the sampling interval. Researchers successfully used the Lyapunov–Krasovskii (Yue et al. 2004) and the Razumikhin theorems (Yu et al. 2004) to study the stability of a NCS that is modeled as DDEs.

Summary and Future Directions

This article introduced the concept of a networked control system and its general architecture. Several key issues specific to a NCS, such as band-limited channels, network-induced delays, and information packet dropouts, were explained. The stability condition of a NCS with various network effects was discussed with several common modeling techniques.

In terms of future directions, there has been significant effort in analyzing networked control systems with variable sampling rate, but most results investigate the stability for a given worst-case interval between consecutive sampling times, leading to conservative results. An open area of research would be to look at methods that take into account a stochastic characterization for the inter-sampling times. Substantial work has also been devoted to determining the stability of a NCS, as described in this article. Possible open areas of research would be to consider design issues related to the joint stability and

performance of the system. The design and development of controllers for a NCS is also an open area of research. In designing a controller for a NCS, one has to take into account the challenges introduced by the communication network. Only afterward can analysis of the whole system take place.

Cross-References

- [Data Rate of Nonlinear Control Systems and Feedback Entropy](#)
- [Information and Communication Complexity of Networked Control Systems](#)
- [Networked Control Systems: Estimation and Control over Lossy Networks](#)
- [Networked Systems](#)
- [Quantized Control and Data Rate Constraints](#)

Bibliography

- Baillieul J (1999) Feedback designs for controlling device arrays with communication channel bandwidth constraints. In: Lecture notes of the fourth ARO workshop on smart structures, Penn State University
- Baillieul J, Antsaklis PJ (2007) Control and communication challenges in networked control systems. *Proc IEEE* 95(1):9–28
- Hespanha JP, Naghshtabrizi P, Xu Y (2007) A survey of recent results in networked control systems. *Proc IEEE* 95(1):138–162
- Johansson KH, Torngrén M, Nielsen L (2005) Vehicle applications of controller area network. In: Levine WS, Hristu-Varsakelis D (eds) *Handbook of networked and embedded control systems*. Birkhäuser, Boston, pp 741–765
- Lin H, Zhai G, Antsaklis PJ (2003) Robust stability and disturbance attenuation analysis of a class of networked control systems. In: 42nd IEEE conference on decision and control, Maui, vol 2, pp 1182–1187
- Lin H, Zhai G, Fang L, Antsaklis PJ (2005) Stability and H_1 performance preserving scheduling policy for networked control systems. In: *Proceedings 16th IFAC world congress*, Prague
- Montestruque LA, Antsaklis PJ (2004) Stability of model-based networked control systems with time-varying transmission times. *IEEE Trans Autom Control* 49(9):1562–1572
- Nair GN, Evans RJ (2000) Stabilization with data-rate-limited feedback: tightest attainable bounds. *Syst Control Lett* 41(1):49–56. Elsevier
- Nesic D, Teel A (2004a) Input-output stability properties of networked control systems. *IEEE Trans Autom Control* 49(10):1650–1667
- Nesic D, Teel A (2004b) Input-to-state stability of networked control systems. *Automatica* 40(12):2121–2128
- Onat A, Naskali T, Parlakay E, Mutluer O (2011) Control over imperfect networks: model-based predictive networked control systems. *IEEE Trans Ind Electron* 58:905–913
- Seiler P, Sengupta R (2005) An H_∞ approach to networked control. *IEEE Trans Autom Control* 50(3):356–364
- Tabbara M, Nesic D, Teel A (2007) Stability of wireless and wireline networked control systems. *IEEE Trans Autom Control* 52(9):1615–1630
- Tatikonda S, Mitter S (2004) Control under communication constraints. *IEEE Trans Autom Control* 49(7):1056–1068
- Walsh G, Ye H (2001) Scheduling of networked control systems. *IEEE Control Syst Mag* 21(1): 57–65
- Walsh G, Ye H, Bushnell L (2001a) Stability analysis of networked control systems. *IEEE Trans Control Syst Technol* 10(3):438–446
- Walsh G, Beldiman O, Bushnell L (2001b) Asymptotic behavior of nonlinear networked control systems. *IEEE Trans Autom Control* 44:1093–1097
- Wang X, Hovakimyan N (2013) Distributed control of uncertain networked systems: a decoupled design. *IEEE Trans Autom Control* 58(10):2536–2549
- Wong WS, Brockett RW (1999) System with finite communication bandwidth constraints-II: stabilization with limited information feedback. *IEEE Trans Autom Control* 44(5):1049–1053
- Yang TC (2006) Networked control system: a brief survey. *IEE Proc Control Theory Appl* 153(4):403–412
- Ye H, Walsh G, Bushnell L (2001) Real-time mixed-traffic wireless networks. *IEEE Trans Ind Electron* 48(5): 883–890
- Yu M, Wang L, Chu T, Hao F (2004) An LMI approach to networked control systems with data packet dropout and transmission delays. In: 43rd IEEE conference on decision and control, Paradise Island, Bahamas, vol 4, pp 3545–3550
- Yue D, Han QL, Peng C (2004) State feedback controller design for networked control systems. *IEEE Trans Circuits Syst* 51(11):640–644
- Zhai G, Hu B, Yasuda K, Michel A (2002) Qualitative analysis of discrete-time switched systems. In: *Proceedings of the American control conference*, vol 3, pp 1880–1885
- Zhang W, Branicky MS (2001) Stability of networked control systems with time-varying transmission period. In: *Allerton conference on communication, control, and computing*, Monticello
- Zhang W, Branicky MS, Phillips SM (2001) Stability of networked control systems. *IEEE Control Syst Mag* 21(1):84–99

Networked Control Systems: Estimation and Control over Lossy Networks

João P. Hespanha¹ and Alexandre R. Mesquita²

¹Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA, USA

²Department of Electronics Engineering, Federal University of Minas Gerais, Belo Horizonte, Brazil

Abstract

This entry discusses optimal estimation and control for lossy networks. Conditions for stability are provided both for two-link and multiple-link networks. The online adaptation of network resources (controlled communication) is also considered.

Keywords

Automatic control · Communication networks · Controlled communication · Estimation · Networked control systems · Stability

Introduction

Network Control Systems (NCSs) are spatially distributed systems in which the communication between sensors, actuators, and controllers occurs through a shared band-limited digital communication network. In this entry, we consider the problem of estimation and control over such networks.

A significant difference between NCSs and standard digital control is the possibility that data may be lost while in transit through the network. Typically, *packet dropouts* result from transmission errors in physical network links (which is far more common in wireless than in wired networks) or from buffer overflows

due to congestion. Long transmission delays sometimes result in packet reordering, which essentially amounts to a packet dropout if the receiver discards “outdated” arrivals. Reliable transmission protocols, such as TCP, guarantee the eventual delivery of packets. However, these protocols are not appropriate for NCSs since the retransmission of old data is generally not useful. Another important difference between NCSs and standard digital control systems is that, due to the nature of network traffic, delays in the control loop may be time varying and nondeterministic.

In this entry, we concentrate on the problem of control and estimation in the presence of packet losses, leaving other important features of NCSs (such as quantization and random delays) to be addressed in other entries of this encyclopedia. Consequently, we assume that the network can be viewed as a channel that can carry real numbers without distortion, but that some of the messages may be lost. This network model is appropriate when the number of bits in each data packet is sufficiently large so that quantization effects can be ignored, but packet dropouts cannot. For more general channel models, see, for example, Imer and Basar (2005).

This entry also does not address network transmission delays explicitly. In general, network delays have two components: one that is due to the time spent transmitting packets and another due to the time packets wait in buffers waiting to be transmitted. Delays due to packet transmission present little variation and may be modeled as constants. For control design purposes, these delays may be incorporated into the plant model. Delays due to buffering depend on the network traffic and are typically random; they can be analyzed using the techniques developed in Antunes et al. (2012).

Notation and Basic Definitions. Throughout the entry, $\mathbb{R}$ stands for real numbers and $\mathbb{N}$ for nonnegative integers. For a given matrix $A \in \mathbb{R}^{n \times n}$ and vector $x \in \mathbb{R}^n$, $\|x\| := \sqrt{x'x}$ denotes the Euclidean norm of x , and $\lambda(A)$ the set of eigenvalues of A . Random variables are generally denoted in boldface. For a random variable $\mathbf{y}$, $E[\mathbf{y}]$ stands for the expectation of $\mathbf{y}$.

Two-Link Networks

Here, we consider a control/estimation problem when all network effects can be modeled using two erasure channels: one from the sensor to the controller and the other from the controller to the actuator (see Fig. 1).

We restrict our attention to a linear time-invariant (LTI) plant with intermittent observation and control packets:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + v_k B\mathbf{u}_k + \mathbf{w}_k, \quad (1a)$$

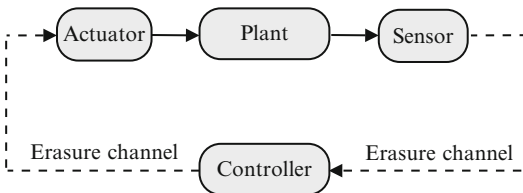
$$\mathbf{y}_k = \theta_k C\mathbf{x}_k + \mathbf{v}_k, \quad (1b)$$

$\forall k \in \mathbb{N}$, $\mathbf{x}_k, \mathbf{w}_k \in \mathbb{R}^n$, $\mathbf{y}_k, \mathbf{v}_k \in \mathbb{R}^p$, where $(\mathbf{x}_0, \mathbf{w}_k, \mathbf{v}_k)$ are mutually independent, zero-mean Gaussian with covariance matrices (P_0, R_w, R_v) , and $\theta_k, v_k \in \{0, 1\}$ are i.i.d. Bernoulli random variables with $\Pr\{\theta_k = 1\} = \bar{\theta}$ and $\Pr\{v_k = 1\} = \bar{v}$. The variable θ_k models the packet loss between sensor and controller, whereas v_k models the packet loss between controller and actuator. When there is a packet drop from controller to actuator, we set the actuator's output to zero. Different strategies, such as holding the control input, could still be modeled using (1) by augmenting of the state vector.

The information available to the controller up to time k is given by the information set:

$$\mathcal{I}_k = \{P_0\} \cup \{\mathbf{y}_\ell, \theta_\ell : \ell \leq k\} \cup \{v_\ell : \ell \leq k-1\}.$$

Here, we make an important assumption that acknowledgment packets from the actuator are always received by the controller so that $v_\ell, \ell \leq k-1$ is available at time k to the remote estimator.



Networked Control Systems: Estimation and Control over Lossy Networks, Fig. 1 Control system with two network links

Optimal Estimation with Remote Computation

The optimal mean-square estimate of $\mathbf{x}_k$, given the information known to the remote estimator at time k , is given by

$$\hat{\mathbf{x}}_{k|k} := E[\mathbf{x}_k | \mathcal{I}_k].$$

This estimate can be computed recursively using the following time-varying Kalman filter (TVKF) (Sinopoli et al. 2004):

$$\hat{\mathbf{x}}_{0|-1} = 0, \quad (2a)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \theta_k F_k (\mathbf{y}_k - C\hat{\mathbf{x}}_{k|k-1}), \quad (2b)$$

$$\hat{\mathbf{x}}_{k+1|k} = A\hat{\mathbf{x}}_{k|k} + v_k B\mathbf{u}_k, \quad (2c)$$

with the gain matrix F_k calculated recursively as follows

$$F_k = P_k C' (C P_k C' + R_v)^{-1},$$

$$P_{k+1} = A P_k A' + R_w - \theta_k A F_k (C P_k C' + R_v) F_k' A'.$$

Each P_k corresponds to the estimation error covariance matrix

$$P_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})'].$$

For this estimator, there exists a critical value θ_c for the dropout rate $\bar{\theta}$, above which the estimation error covariance becomes unbounded:

Theorem 1 (Sinopoli et al. 2004) Assume that $(A, R_w^{1/2})$ is controllable, (A, C) is observable, and A is unstable. Then there exists a critical value $\theta_c \in (0, 1]$ such that

$$E[P_k] \leq M, \forall k \in \mathbb{N} \Leftrightarrow \bar{\theta} \geq \theta_c$$

where M is a positive definite matrix that may depend on P_0 . Furthermore, the critical value θ_c satisfies $\theta_{\min} \leq \theta_c \leq \theta_{\max}$, where the lower bound is given by

$$\theta_{\min} = 1 - \frac{1}{(\max\{|\lambda(A)|\})^2}, \quad (3)$$

and the upper bound is given by the solution to the following (quasi-convex) optimization problem:

$$\theta_{\max} = \min\{\theta \geq 0 : \Psi_{\theta}(Y, Z) > 0, \\ 0 \leq Y \leq I \text{ for some } Y, Z\},$$

where

$$\Psi_{\theta}(Y, Z) = \begin{bmatrix} Y & \sqrt{\theta}(YA + ZC) & \sqrt{1-\theta}YA \\ \sqrt{\theta}(A'Y + C'Z') & Y & 0 \\ \sqrt{1-\theta}A'Y & 0 & Y \end{bmatrix}.$$

Remark 1 In some special cases, the upper bound in (3) is tight in the sense that $\theta_c - \theta_{\min}$. The largest class of systems known for which this occurs is that of *nondegenerate systems* defined in Mo and Sinopoli (2012). Examples of systems in this class include (1) those for which the matrix C is invertible and (2) those with a detectable pair (A, C) and such that the matrix A is diagonalizable with unstable eigenvalues having distinct absolute values.

Optimal Control with Remote Computation

From a control perspective, one may also be interested in finding control sequences $\mathbf{u}^N = \{\mathbf{u}_1, \dots, \mathbf{u}_{N-1}\}$, as functions of the information set $\mathcal{I}_N$, which minimize cost functions of the form

$$J = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbb{E} \left[\sum_{k=0}^{N-1} (\mathbf{x}'_k W \mathbf{x}_k + v_k \mathbf{u}'_k U \mathbf{u}_k) | \mathcal{I}_k \right].$$

Theorem 2 (Schenato et al. 2007) Assume that (A, B) and $(A, R_w^{1/2})$ are controllable, (A, C) and $(A, W^{1/2})$ are observable, and A is unstable. Then, finite control costs J are achievable if and only if $\bar{\theta} > \theta_c$ and $\bar{v} > v_c$, where the critical value v_c is given by the (quasi-convex) optimization problem

$$v_c = \min\{v \geq 0 : \Psi_v(Y, Z) > 0, \\ 0 \leq Y \leq I \text{ for some } Y, Z\},$$

where

$$\Psi_v(Y, Z) = \begin{bmatrix} Y & Y & \sqrt{\bar{v}}ZU^{1/2} & \sqrt{\bar{v}}(YA' + ZB') & \sqrt{1-\bar{v}}YA' \\ Y & W^{-1} & 0 & 0 & 0 \\ \sqrt{\bar{v}}U^{1/2}Z' & 0 & I & 0 & 0 \\ \sqrt{\bar{v}}(AY + BZ') & 0 & 0 & Y & 0 \\ \sqrt{1-\bar{v}}AY & 0 & 0 & 0 & Y \end{bmatrix}.$$

Moreover, under the above conditions, the separation principle holds in the sense that the optimal control is given by

$$\mathbf{u}_k = -(B'SB + U)^{-1}B'SA\hat{\mathbf{x}}_{k|k},$$

where $\hat{\mathbf{x}}_{k|k}$ is an optimal state estimate given by (2) and the matrix S is the solution to the modified algebraic Riccati (MARE) equation

$$S = A'SA + W - \bar{v}A'SB(B'SB + U)^{-1}B'SA.$$

Solutions to the MARE may be obtained iteratively when $\bar{v} > v_c$.

Estimation with Local Computation

To reduce the gap between the bounds $\theta_{\min}$ and $\theta_{\max}$ on the critical value of the drop probability in Theorem 1 and to allow for larger probabilities of drop, one may choose to compute state estimates at the sensor and transmit those to the controller/actuator. This scheme is motivated by the growing number of *smart sensors* with

embedded processing units that are capable of local computation. For the LTI plant

$$\begin{aligned}\mathbf{x}_{k+1} &= A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{w}_k, \\ \mathbf{y}_k &= C\mathbf{x}_k + \mathbf{v}_k,\end{aligned}$$

the smart sensor can compute locally an optimal state estimate using a standard stationary Kalman filter and transmits this estimate to the controller. We model packet dropouts as before using the process θ_k and assume that the process θ_k is known to the smart sensor by means of an perfect acknowledgment mechanism. This allows the sensor to know $\mathbf{u}_k$ exactly and to use it in the Kalman filter.

Let $\tilde{\mathbf{x}}_{k|k} = E[\mathbf{x}_k | \mathbf{y}_\ell, \theta_\ell, \ell \leq k]$ denote the local estimates transmitted by the sensor. Using the messages successfully received up to time k , the remote estimator computes the optimal estimate

$$\hat{\mathbf{x}}_{k|k-1} = E[\mathbf{x}_k | \theta_\ell, \tilde{\mathbf{x}}_{\ell|k}, \ell \leq k-1].$$

recursively by

$$\begin{aligned}\hat{\mathbf{x}}_{0|-1} &= 0, \\ \hat{\mathbf{x}}_{k|k} &= (1 - \theta_k)\hat{\mathbf{x}}_{k|k-1} + \theta_k\tilde{\mathbf{x}}_{k|k}, \quad k \in \mathbb{N}, \\ \hat{\mathbf{x}}_{k+1|k} &= A\hat{\mathbf{x}}_{k|k} + B\mathbf{u}_k\end{aligned}$$

Notice that now we are applying the (TVKF) to estimate $\tilde{\mathbf{x}}_k$, which is fully observable. Since $\theta_{\min}$ and $\theta_{\max}$ in Theorem 2 are equal for fully observable processes (Schenato et al. 2007), the local computation scheme grants a minimal critical value θ_c as stated in the theorem below.

Theorem 3 Assume that $(A, R_w^{1/2})$ is controllable, (A, C) is OBSERVABLE, and A is unstable. Then the critical value θ_c is given by $\theta_{\min}$ in (3), i.e.,

$$E[P_k] \leq M, \forall k \in \mathbb{N} \quad \Leftrightarrow \quad \bar{\theta} \geq \theta_{\min}$$

where M is a positive definite matrix that may depend on P_0 .

Drops in the Acknowledgement Packets

When there are drops in the acknowledgment channel from the actuator to the controller, the controller does not always know v_k , and therefore, it might not always have access to the control inputs that are actually applied to the plant. In this case, the posterior state probability becomes a Gaussian mixture distribution with infinitely many components, and the separation principle no longer holds (Schenato et al. 2007). This makes the estimation and control problems computationally more difficult, and, due to the smaller information set, some performance degradation in the control performance should be expected. For this reason, it is generally a good design choice to keep controller and actuator collocated when drops in the acknowledgment channels are significant.

Buffering

As an alternative to the approach described in section “Estimation with Local Computation” to use local computation at a smart sensor to allow for larger probabilities of drop, the designer may also consider the transmission of a sequence of previous measurements $\mathbf{y}_k, \mathbf{y}_{k-1}, \dots, \mathbf{y}_{k-N}$ in each packet. This approach is motivated by the fact that often data packets can carry much more than one vector of measured outputs. When N is reasonably large, one should expect similar estimation/control performances as in the approach described in section “Estimation with Local Computation”, but with a reduced computational effort at the sensor.

Analogously, an improvement to zeroing or simply holding the control input in case of packet drops between controller and actuator is for the controller to transmit a control sequence $\mathbf{u}_k, \mathbf{u}_{k+1}, \dots, \mathbf{u}_{k+N}$ that contains not only the control $\mathbf{u}_k$ to be used at the current time instant but also a few future controls $\mathbf{u}_{k+1}, \mathbf{u}_{k+2}, \dots, \mathbf{u}_{k+N}$. In the case of packet drops between controller and actuator, the actuator can use previously received “future” control inputs in lieu of the one contained in the lost packet. The sequence of future control inputs

may be obtained, e.g., by an optimal receding horizon control strategy (Gupta et al. 2006).

Estimation with Markovian Drops

When θ_k is a Markov process, we no longer have a separation principle, and the optimal controller may depend on the drops sequence. Yet, optimal state estimates are obtained using the same TVKF presented earlier. Below, we give conditions for the stability of the error covariance when drops are governed by the Gilbert-Elliot model: $\Pr\{\theta_{k+1} = j | \theta_k = i\} = p_{ij}$, $i, j \in \{0, 1\}$.

Theorem 4 (Mo and Sinopoli 2012) *Assume that $(A, R_w^{1/2})$ is controllable, A is unstable, and the system given by the pair (A, C) is nondegenerate as discussed in Remark 1. Moreover, suppose that the transition probabilities for the Gilbert-Elliot model satisfy $p_{01}; p_{10} > 0$. Then the expected error covariance $E[P_k]$ is uniformly bounded if*

$$p_{01} > \theta_{\min}$$

and it is unbounded for some initial condition if $p_{01} < \theta_{\min}$.

Networks with Multiple Links

We now consider feedback loops that are closed over a network of communication links, each of which drops packets according to a Bernoulli process. The sensor communicates with a controller across the network, and we assume that controller and actuator are collocated. The network may be represented by a graph $\mathcal{G}$ with nodes in the set $\mathcal{V}$ and edges in the set $\mathcal{E}$, where edges are drawn between two communicating nodes. We denote by p_{ij} the probability of a drop when node i transmits to node j . Drops are assumed to be independent across links and time.

To maximize robustness with respect to drops, sensors use a Kalman filter to compute an optimal estimate for the state of the process based on their measurements and transmit this estimate across the network. When the sensors do not have access to the process input, they can take advantage of the linearity of the Kalman filter:

as the output of a Kalman filter is the sum of a term due to measurements with another term due to control inputs, sensors may compute only the contribution due to measurements and transmit it to the controller, which can subsequently add the contribution due to the control inputs. This guarantees that optimal state estimates can still be computed at the control node, even when the sensors do not know the control input (Gupta et al. 2009).

The communication in the network goes as follows. Sensors time stamp their estimates and broadcast them to all nodes in their communication ranges. After receiving information from their neighbors, nodes compare time stamps and keep only the most recent estimates. These estimates are broadcasted to all neighboring nodes. When the controller receives new information, the optimal Kalman estimate is reconstructed, taking into account the total transmission delay (learned from the packet time stamps), and a standard LQG control can be used (Gupta et al. 2009).

To determine whether or not this procedure results in a stable closed loop, one defines a *cut* $\mathcal{C} = (\mathcal{S} \Leftrightarrow \mathcal{T})$ to be a partition of the node set $\mathcal{V}$ such that the sensor node is in $\mathcal{S}$ and the controller node is in $\mathcal{T}$. The cut-set is then defined as the set of edges $(i, j) \in \mathcal{E}$ such that $i \in \mathcal{S}$ and $j \in \mathcal{T}$, i.e., the set of edges that connect the sets $\mathcal{S}$ and $\mathcal{T}$. The *max-cut probability* is then defined as

$$p_{\max\text{-cut}} = \max_{\text{all cuts } (\mathcal{S} \Leftrightarrow \mathcal{T})} \prod_{(i,j) \in \mathcal{S} \times \mathcal{T}} p_{ij}.$$

The above maximization can be rewritten as a minimization over the sums of $-\log p_{ij}$, which leads to a linear program known as the minimum cut problem in network optimization theory (Cook 1995).

Theorem 5 (Gupta et al. 2009) *Assume that $R_w, R_v > 0$, that (A, B) is stabilizable, that (A, C) is observable, and that A is unstable. Then the control and communication policy described above is optimal for quadratic costs, and the expected state covariance is bounded if and only if*

$$p_{\max\text{-cut}} \cdot (\max\{|\lambda(A)|\})^2 < 1.$$

Estimation with Controlled Communication

To actively reduce network traffic and power consumption, sensor measurements may not be sent to the remote estimator at every time step. In addition, one may have the ability to somewhat control the probability of packet drops by varying the transmit power or by transmitting copies of the same message through multiple channel realizations. This is known as *controlled communication*, and it allows the designer to establish a trade-off between communication and estimation performance.

We consider the local estimation scenario described in section “[Estimation with Local Computation](#)” with the difference that the Bernoulli drops are now modulated as follows

$$\theta_k = \begin{cases} 1 & \text{with prob. } \Lambda_k \\ 0 & \text{with prob. } 1 - \Lambda_k \end{cases}$$

where the sensor is free to choose $\Lambda_k \in [0, p_{\max}]$ as a function of the information available up to time k . With its choice, the sensor incurs on a communication cost $c(\Lambda_k)$ at time k , where $c(\cdot)$ is some increasing function that may represent, for example, the energy needed in order to transmit with a probability of drop equal to Λ_k . Note that transmission scheduling, where Λ_k is either 0 or $p_{\max}$, is a special case of this framework.

In order to choose Λ_k , the sensor considers the estimation error $\tilde{\mathbf{e}}_k := \tilde{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1}$ between the local and the remote estimators. This error evolves according to

$$\tilde{\mathbf{e}}_{k+1} = \begin{cases} \mathbf{d}_k & \text{with prob. } \Lambda_k \\ A\tilde{\mathbf{e}}_k + \mathbf{d}_k & \text{with prob. } 1 - \Lambda_k \end{cases}$$

where $\mathbf{d}_k$ is the innovations process arising from the standard Kalman filter in the smart sensor.

Our objective is to find a “communication policy” that minimizes the long-term average cost

$$\tilde{J} := \lim_{K \rightarrow \infty} \frac{1}{K} \mathbb{E} \left[\sum_{k=0}^{K-1} \|\tilde{\mathbf{e}}_k\|^2 + \lambda c(\Lambda_k) \right], \quad \lambda > 0, \quad (4)$$

which penalizes a linear combination of the remote estimation error variance $\mathbb{E}[\|\tilde{\mathbf{e}}_k\|^2]$ and the average communication cost $\mathbb{E}[c(\Lambda_k)]$. In this context, a *communication policy* should be understood as a rule that selects Λ_k as a function of the information available to the sensor.

When

$$(1 - p_{\max}) \max\{|\lambda(A)|\}^2 < 1,$$

there exists an optimal communication policy that chooses Λ_k as a function of $\tilde{\mathbf{e}}_k$, which may be computed via dynamic programming and value iteration (Mesquita et al. 2012). While this procedure can be computationally difficult, it is often possible to obtain suboptimal but reasonable performance with rollout policies such as the following one:

$$\Lambda_k = \arg \min_{\Lambda \in [0, p_{\max}]} [(p_{\max} - \Lambda) \tilde{\mathbf{e}}_k' A' H A \tilde{\mathbf{e}}_k + \lambda c(\Lambda)] \quad (5)$$

where H is the positive semidefinite solution to the Lyapunov equation $(1 - p_{\max})A'HA - H = -I$ (Mesquita et al. 2012).

When computing $\tilde{\mathbf{e}}_k$ and Λ_k in (5) is computationally too costly for the sensor, one may prefer to make Λ_k a function of the number of consecutive dropped packets ℓ_k . In this case, minimizing $\tilde{J}$ in (4) is equivalent to minimizing the cost

$$\bar{J} := \lim_{K \rightarrow \infty} \frac{1}{K} \mathbb{E} \left[\sum_{k=0}^{K-1} \text{trace}(\Sigma_{\ell_k}) + \lambda c(\Lambda_k) \right],$$

where

$$\Sigma_{\ell} := \sum_{m=0}^{\ell} A^m R_w A^m.$$

Since ℓ_k belongs to a countable set, one can very efficiently solve this optimization using dynamic programming (Mesquita et al. 2012).

Summary and Future Directions

Most positive results in the subject rely on the assumption of perfect acknowledgments and on actuators and controllers being collocated. Future research should address ways of circumventing these assumptions.

Cross-References

- ▶ [Data Rate of Nonlinear Control Systems and Feedback Entropy](#)
- ▶ [Information and Communication Complexity of Networked Control Systems](#)
- ▶ [Networked Control Systems: Architecture and Stability Issues](#)
- ▶ [Networked Systems](#)
- ▶ [Quantized Control and Data Rate Constraints](#)

Bibliography

- Antunes D, Hespanha JP, Silvestre C (2012) Volterra integral approach to impulsive renewal systems: application to networked control. *IEEE Trans Autom Control* 57:607–619
- Cook WJ (1995) Combinatorial optimization: papers from the DIMACS special year, vol 20. American Mathematical Society, Providence
- Gupta V, Sinopoli B, Adlakha S, Goldsmith A, Murray R (2006) Receding horizon networked control. In: *Proceedings of the allerton conference on communication, control, and computing*, Monticello
- Gupta V, Dana AF, Hespanha JP, Murray RM, Hassibi B (2009) Data transmission over networks for estimation and control. *IEEE Trans Autom Control* 54(8):1807–1819
- Imer OC, Basar T (2005) Optimal estimation with limited measurements. In: *Proceedings of the 44th IEEE conference on decision and control, 2005 and 2005 European control conference (CDC-ECC'05)*, Seville
- Mesquita AR, Hespanha JP, Nair GN (2012) Redundant data transmission in control/estimation over lossy networks. *Automatica* 48(8):1612–1620
- Mo Y, Sinopoli B (2012) Kalman filtering with intermittent observations: tail distribution and critical value. *IEEE Trans Autom Control* 57(3):677–689
- Schenato L, Sinopoli B, Franceschetti M, Poolla K, Sastry SS (2007) Foundations of control and estimation over lossy networks. *Proc IEEE* 95(1):163–187
- Sinopoli B, Schenato L, Franceschetti M, Poolla K, Jordan MI, Sastry SS (2004) Kalman filtering with intermittent observations. *IEEE Trans Autom Control* 49(9):1453–1464

Networked Systems

Jorge Cortés

Department of Mechanical and Aerospace Engineering, University of California, La Jolla, CA, USA

Abstract

This article provides a brief overview on networked systems from a systems and control perspective. We pay special attention to the nature of the interactions among agents; the critical role played by information sharing, dissemination, and aggregation; and the distributed control paradigm to engineer the behavior of networked systems.

Keywords

Multiagent systems · Autonomous networks · Cooperative control · Swarms

Introduction

Networked systems appear in numerous scientific and engineering domains, including communication networks (Toh 2001), multi-robot networks (Arkin 1998; Balch and Parker 2002), sensor networks (Santi 2005; Schenato et al. 2007), water irrigation networks (Cantoni et al. 2007), power and electrical networks (Chow 1982; Chiang et al. 1995; Dörfler et al. 2013), camera networks (Song et al. 2011), transportation networks (Ahuja et al. 1993), social networks (Jackson 2010), chemical and biological networks (Kuramoto 1984; Strogatz 2003), and brain networks (Bullmore and Sporns 2009; Meunier et al. 2010; Gu et al. 2015). Their applications are pervasive, ranging from environmental monitoring, ocean sampling, and marine energy systems through search and rescue missions, high-stress deployment in disaster recovery, health monitoring of critical infrastructure to science imaging, the smart grid, and cybersecurity.

The rich nature of networked systems makes it difficult to provide a definition that, at the same time, is comprehensive enough to capture their variety and simple enough to be expressive of their main features. With this in mind, we loosely define a networked system as a “system of systems,” i.e., a collection of agents that interact with each other. These groups might be heterogeneous, composed by human, biological, or engineered agents possessing different capabilities regarding mobility, sensing, actuation, communication, and computation. Individuals may have objectives of their own or may share a common objective with others – which in turn might be adversarial with respect to another subset of agents.

In a networked system, the evolutions of the states of individual agents are coupled. Coupling might be the result of the physical interconnection among the agents, the consequence of the implementation of coordination algorithms where agents use information about each other, or a combination of both. There is diversity too in the nature of agents themselves and the interactions among them, which might be cooperative and adversarial or belong to the rich range between the two. Due to changes in the state of the agents, the network, or the environment, interactions among agents may be changing and dynamic. Such interactions may be structured across different layers, which themselves might be organized in a hierarchical fashion. Networked systems may also interact with external entities that specify high-level commands that trickle down through the system all the way to the agent level.

A defining characteristic of a networked system is the fact that information, understood in a broad sense, is sparse and distributed across the agents. As such, different individuals have access to information of varying degrees of quality. As part of the operation of the networked system, mechanisms are in place to share, transmit, and/or aggregate this information. Some information may be disseminated throughout the whole network, or in some cases, all information can be made centrally available at a reasonable cost. In other scenarios, however, the latter might turn

out to be too costly, unfeasible, or undesirable because of privacy and security considerations. Individual agents are the basic unit for decision-making but decisions might be made from intermediate levels of the networked system all the way to a central planner. The combination of information availability and decision-making capabilities gives rise to an ample spectrum of possibilities between the centralized control paradigm, where all information is available at a central planner who makes the decisions, and the fully distributed control paradigm, where individual agents only have access to the information shared by their neighbors in addition to their own.

Perspective from Systems and Control

There are many aspects that come into play when dealing with networked systems regarding computation, processing, sensing, communication, planning, motion control, and decision-making. This complexity makes their study challenging and fascinating and explains the interest that, with different emphases, they generate in a large number of disciplines. In biology, scientists analyze synchronization phenomena and self-organized swarming behavior in groups with distributed agent-to-agent interactions (Okubo 1986; Parrish et al. 2002; Conradt and Roper 2003; Couzin et al. 2005). In robotics, engineers design algorithmic solutions to help multi-vehicle networks and embedded systems coordinate their actions and perform challenging spatially distributed tasks (Arkin 1998; Committee on Networked Systems of Embedded Computers 2001; Balch and Parker 2002; Howard et al. 2006; Kumar et al. 2008). Graph theorists and applied mathematicians study the role played by the interconnection among agents in the emergence of phase transition phenomena (Bollobás 2001; Meester and Roy 2008; Chung 2010). This interest is also shared in communication and information theory, where researchers strive to design efficient communication protocols and examine the effect of topology control on group connectivity and information dissemination (Zhao and Guibas

2004; Giridhar and Kumar 2005; Lloyd et al. 2005; Santi 2005; Franceschetti and Meester 2007). Game theorists study the gap between the performance achieved by global, network-wide optimizers and the configurations that result from selfish agents interacting locally in social and economic systems (Roughgarden 2005; Nisan et al. 2007; Easley and Kleinberg 2010; Marden and Shamma 2013). In mechanism design, researchers seek to align the objectives of individual self-interested agents with the overall goal of the network. Static and mobile networked systems and their applications to the study of natural phenomena in oceans (Paley et al. 2008; Graham and Cortés 2012; Zhang and Leonard 2010; Das et al. 2012; Ouimet and Cortés 2013), rivers (Ru and Martínez 2013; Tinka et al. 2013), and the environment (DeVries and Paley 2012) also raise exciting challenges in estimation theory, computational geometry, and spatial statistics.

The field of systems and control brings a comprehensive approach to the modeling, analysis, and design of networked systems. Emphasis is put on the understanding of the general principles that explain how specific collective behaviors emerge from basic interactions; the establishment of models, abstractions, and tools that allow us to reason rigorously about complex interconnected systems; and the development of systematic methodologies that help engineer their behavior. The ultimate goal is to establish a science for integrating individual components into complex, self-organizing networks with predictable behavior. To realize the “power of many” and expand the realm of what is possible to achieve beyond the individual agent capabilities, special care is taken to obtain precise guarantees on the stability properties of coordination algorithms, understand the conditions and constraints under which they work, and characterize their performance and robustness against a variety of disturbances and disruptions.

Research Issues and How the Articles in the Encyclopedia Address Them

Given the key role played by agent-to-agent interactions in networked systems, several

encyclopedia articles, “► [Graphs for Modeling Networked Interactions](#)”, “► [Dynamic Graphs, Connectivity of](#)”, and “► [Controllability of Network Systems](#)”, deal with how their nature and effect can be modeled through graphs. This includes diverse aspects such as deterministic and stochastic interactions, static and dynamic graphs, state-dependent and time-dependent neighboring relationships, connectivity, and controllability. The importance of maintaining a certain level of coordination and consistency across the networked system is manifested in the various articles that deal with coordination tasks that are, in some way or another, related to some form of agreement. These include consensus “► [Averaging Algorithms and Consensus](#)”, formation control “► [Vehicular Chains](#)”, flocking “► [Flocking in Networked Systems](#)”, synchronization “► [Oscillator Synchronization](#)”, and distributed optimization “► [Distributed Optimization](#)”. A great deal of work, “► [Optimal Deployment and Spatial Coverage](#)” and “► [Multi-vehicle Routing](#)”, is also devoted to the design of cooperative strategies that achieve spatially distributed tasks such as optimal coverage, space partitioning, vehicle routing, and servicing. These articles explore the optimal placement of agents, the optimal tuning of sensors, and the distributed optimization of network resources. The articles “► [Estimation and Control Over Networks](#)” and “► [Distributed Estimation in Networks](#)” explore the impact that communication channels may have on the execution of estimation and control tasks over networks of sensors and actuators. The article “► [Nash Equilibrium Seeking Over Networks](#)” deals with game-theoretic scenarios that involve distributed interactions among multiple players to learn Nash equilibria under partial individual information. The growing importance in developing intelligent infrastructure of preserving the privacy of data obtained from individuals or privacy-sensitive entities in the operation of large-scale network systems is examined in “► [Privacy in Network Systems](#)”. Finally, “► [Routing in Transportation Networks](#)” provides an overview of the basics of routing control design to optimize network-level transportation objectives taking into account

complex traffic flow dynamics and driver preferences. A strong point of commonality among the contributions is the precise characterization of the scalability of coordination algorithms, together with the rigorous analysis of their correctness and stability properties. Another focal point is the analysis of the performance gap between centralized and distributed approaches in regard to the ultimate network objective.

Further information about other relevant aspects of networked systems can be found in other sections throughout this encyclopedia. Among these, we highlight the synthesis of cooperative strategies for data fusion, distributed estimation, and adaptive sampling, the analysis of the network operation under communication constraints (e.g., limited bandwidth, message drops, delays, and quantization), the distributed model predictive control, and the handling of uncertainty, imprecise information, and events via discrete event systems and triggered control.

Summary and Future Directions

In conclusion, this article has illustrated ways in which systems and control can help us design and analyze networked systems. We have focused on the role that information and agent interconnection play in shaping their behavior. We have also made emphasis on the increasingly rich set of methods and techniques that allow to provide correctness and performance guarantees. The field of networked systems is vast and the amount of work impossible to survey in this brief article. The reader is invited to further explore additional topics beyond the ones mentioned here. The monographs (Ren and Beard 2008; Bullo et al. 2009; Mesbahi and Egerstedt 2010; Alpcan and Başar 2010; Bullo 2019), edited volumes (Kumar et al. 2004; Shamma 2008; Saligrama 2008), and manuscripts (Olfati-Saber et al. 2007; Baillieul and Antsaklis 2007; Leonard et al. 2007; Kim and Kumar 2012), together with the references provided in the encyclopedia articles mentioned above, are a good starting point to undertake this enjoyable effort. Given the big impact that networked systems have, and will continue to have, in our society,

from energy and transportation through human interaction and healthcare to biology and the environment, there is no doubt that the coming years will witness the development of more tools, abstractions, and models that allow to reason rigorously about intelligent networks and for techniques that help design truly autonomous and adaptive networks.

Cross-References

- ▶ [Averaging Algorithms and Consensus](#)
- ▶ [Controllability of Network Systems](#)
- ▶ [Distributed Estimation in Networks](#)
- ▶ [Distributed Optimization](#)
- ▶ [Dynamic Graphs, Connectivity of](#)
- ▶ [Estimation and Control over Networks](#)
- ▶ [Flocking in Networked Systems](#)
- ▶ [Graphs for Modeling Networked Interactions](#)
- ▶ [Multi-vehicle Routing](#)
- ▶ [Nash Equilibrium Seeking over Networks](#)
- ▶ [Optimal Deployment and Spatial Coverage](#)
- ▶ [Oscillator Synchronization](#)
- ▶ [Privacy in Network Systems](#)
- ▶ [Routing in Transportation Networks](#)
- ▶ [Vehicular Chains](#)

Bibliography

- Ahuja RK, Magnanti TL, Orlin JB (1993) Network flows: theory, algorithms, and applications. Prentice Hall, Englewood Cliffs
- Alpcan T, Başar T (2010) Network security: a decision and game-theoretic approach. Cambridge University Press, Cambridge
- Arkin RC (1998) Behavior-based robotics. MIT Press, London
- Baillieul J, Antsaklis PJ (2007) Control and communication challenges in networked real-time systems. *Proc IEEE* 95(1):9–28
- Balch T, Parker LE (eds) (2002) Robot teams: from diversity to polymorphism. A. K. Peters, Natick
- Bollobás B (2001) Random graphs, 2nd edn. Cambridge University Press, Cambridge
- Bullmore E, Sporns O (2009) Complex brain networks: graph theoretical analysis of structural and functional systems. *Nat Rev Neurosci* 10(3):186–198
- Bullo F (2019) Lectures on network systems, 1st edn. Kindle Direct Publishing, with contributions by J. Cortes, F. Dorfler, and S. Martinez, Princeton, NJ

- Bullo F, Cortés J, Martinez S (2009) Distributed control of robotic networks. Applied mathematics series. Princeton University Press, Princeton
- Cantoni M, Weyer E, Li Y, Ooi SK, Mareels I, Ryan M (2007) Control of large-scale irrigation networks. *Proc IEEE* 95(1):75–91
- Chiang HD, Chu CC, Cauley G (1995) Direct stability analysis of electric power systems using energy functions: theory, applications, and perspective. *Proc IEEE* 83(11):1497–1529
- Chow JH (1982) Time-scale modeling of dynamic networks with applications to power systems. Springer, Berlin/New York
- Chung FRK (2010) Graph theory in the information age. *Not AMS* 57(6):726–732
- Committee on Networked Systems of Embedded Computers (2001) Embedded, everywhere: a research agenda for networked systems of embedded computers. National Academy Press, Washington, DC
- Conradt L, Roper TJ (2003) Group decision-making in animals. *Nature* 421(6919):155–158
- Couzin ID, Krause J, Franks NR, Levin SA (2005) Effective leadership and decision-making in animal groups on the move. *Nature* 433(7025):513–516
- Das J, Py F, Maughan T, O'Reilly T, Messié M, Ryan J, Sukhatme GS, Rajan K (2012) Coordinated sampling of dynamic oceanographic features with AUVs and drifters. *Int J Robot Res* 31(5):626–646
- DeVries L, Paley D (2012) Multi-vehicle control in a strong flowfield with application to hurricane sampling. *AIAA J Guid Control Dyn* 35(3):794–806
- Dörfler F, Chertkov M, Bullo F (2013) Synchronization in complex oscillator networks and smart grids. *Proc Natl Acad Sci* 110(6):2005–2010
- Easley D, Kleinberg J (2010) Networks, crowds, and markets: reasoning about a highly connected world. Cambridge University Press, Cambridge
- Franceschetti M, Meester R (2007) Random networks for communication. Cambridge University Press, Cambridge
- Giridhar A, Kumar PR (2005) Computing and communicating functions over sensor networks. *IEEE J Sel Areas Commun* 23(4):755–764
- Graham R, Cortés J (2012) Adaptive information collection by robotic sensor networks for spatial estimation. *IEEE Trans Autom Control* 57(6):1404–1419
- Gu S, Pasqualetti F, Cieslak M, Telesford QK, Yu AB, Kahn AE, Medaglia JD, Vettel JM, Miller MB, Grafton ST, Bassett DS (2015) Controllability of structural brain networks. *Nat Commun* 6:8414
- Howard A, Parker LE, Sukhatme GS (2006) Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment, and detection. *Int J Robot Res* 25(5–6):431–447
- Jackson MO (2010) Social and economic networks. Princeton University Press, Princeton, NJ
- Kim KD, Kumar PR (2012) Cyber-physical systems: a perspective at the centennial. *Proc IEEE* 100(Special Centennial Issue):1287–1308
- Kumar V, Leonard NE, Morse AS (eds) (2004) Cooperative control. Lecture notes in control and information sciences, vol 309. Springer, Berlin
- Kumar V, Rus D, Sukhatme GS (2008) Networked robots. In: Siciliano B, Khatib O (eds) Springer handbook of robotics. Springer, Berlin, pp 943–958
- Kuramoto Y (1984) Chemical oscillations, waves, and turbulence. Springer, Berlin/New York
- Leonard NE, Paley D, Lekien F, Sepulchre R, Fratantoni DM, Davis R (2007) Collective motion, sensor networks and ocean sampling. *Proc IEEE* 95(1):48–74
- Lloyd EL, Liu R, Marathe MV, Ramanathan R, Ravi SS (2005) Algorithmic aspects of topology control problems for ad hoc networks. *Mob Netw Appl* 10(1–2):19–34
- Marden JR, Shamma JS (2013) Game theory and distributed control. In: Young P, Zamir S (eds) Handbook of game theory, vol 4. Elsevier, Oxford
- Meester R, Roy R (2008) Continuum percolation. Cambridge University Press, Cambridge
- Mesbahi M, Egerstedt M (2010) Graph theoretic methods in multiagent networks. Applied mathematics series. Princeton University Press, Princeton
- Meunier D, Lambiotte R, Bullmore ET (2010) Modular and hierarchically modular organization of brain networks. *Front Neurosci* 4:200
- Nisan N, Roughgarden T, Tardos E, Vazirani VV (2007) Algorithmic game theory. Cambridge University Press, Cambridge/New York
- Okubo A (1986) Dynamical aspects of animal grouping: swarms, schools, flocks and herds. *Adv Biophys* 22: 1–94
- Olfati-Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. *Proc IEEE* 95(1):215–233
- Ouimet M, Cortés J (2013) Collective estimation of ocean nonlinear internal waves using robotic underwater drifters. *IEEE Access* 1:418–427
- Paley D, Zhang F, Leonard N (2008) Cooperative control for ocean sampling: the glider coordinated control system. *IEEE Trans Control Syst Technol* 16(4): 735–744
- Parrish JK, Viscido SV, Grunbaum D (2002) Self-organized fish schools: an examination of emergent properties. *Biol Bull* 202:296–305
- Ren W, Beard RW (2008) Distributed consensus in multi-vehicle cooperative control. Communications and control engineering. Springer, London
- Roughgarden T (2005) Selfish routing and the price of anarchy. MIT Press, Cambridge, MA
- Ru Y, Martínez S (2013) Coverage control in constant flow environments based on a mixed energy-time metric. *Automatica* 49(9):2632–2640
- Saligrama V (ed) (2008) Networked sensing information and control. Springer, Boston
- Santi P (2005) Topology control in wireless ad hoc and sensor networks. Wiley, New York
- Schenato L, Sinopoli B, Franceschetti M, Poolla K, Sastry SS (2007) Foundations of control and estimation over lossy networks. *Proc IEEE* 95(1):163–187

- Shamma JS (ed) (2008) Cooperative control of distributed multi-agent systems. Wiley, Chichester
- Song B, Ding C, Kamal AT, Farrel JA, Roy-Chowdhury AK (2011) Distributed camera networks. *IEEE Signal Process Mag* 28(3):20–31
- Strogatz SH (2003) *SYNC: the emerging science of spontaneous order*. Hyperion, New York
- Tinka A, Rafiee M, Bayen A (2013) Floating sensor networks for river studies. *IEEE Syst J* 7(1):36–49
- Toh CK (2001) Ad hoc mobile wireless networks: protocols and systems. Prentice Hall, Englewood Cliffs, NJ
- Zhang F, Leonard NE (2010) Cooperative filters and control for cooperative exploration. *IEEE Trans Autom Control* 55(3):650–663
- Zhao F, Guibas L (2004) *Wireless sensor networks: an information processing approach*. Morgan-Kaufmann, San Francisco

Neuro-inspired Control

Frank L. Lewis¹ and Kyriakos G. Vamvoudakis²

¹University of Texas at Arlington, Automation and Robotics Research Institute, Fort Worth, TX, USA

²The Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Abstract

There has been great interest recently in “universal model-free controllers” that do not need a mathematical model of the controlled plant but mimic the functions of biological processes to learn about the systems they are controlling online, so that performance improves automatically. Neural network (NN) control has had two major thrusts: approximate dynamic programming, which uses NN to approximately solve the optimal control problem, and NN in closed-loop feedback control.

Keywords

Neural networks · Neuro-inspired control · Approximate dynamic programming · Model-free control · Neuro-adaptive control

Neural Feedback Control

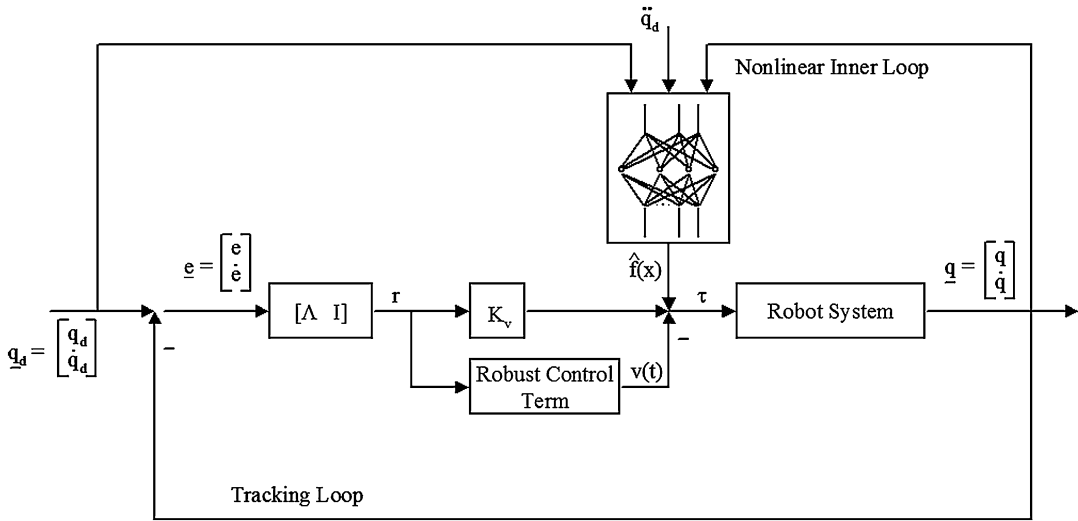
The objective is to design NN feedback controllers that cause a system to follow, or track, a prescribed trajectory or path (see, e.g., Alessandri et al. (1999), Ioannou and Fidan (2006), Kosmatopoulos et al. (1995), and Werbos (1989)). Consider the dynamics of an n -link robot manipulator

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau, \quad (1)$$

with $q(t) \in \mathbb{R}^n$ the joint variable vector, $M(q)$ an inertia matrix, V_m a centripetal/coriolis matrix, $G(q)$ a gravity vector, and $F(\cdot)$ representing friction terms. Bounded unknown disturbances and modeling errors are denoted by τ_d , and the control input torque is $\tau(t)$. The sliding mode control approach (Slotine and Li 1987) can be generalized to NN control systems. Given a desired trajectory $q_d \in \mathbb{R}^n$ define the tracking error $e(t) = q_d(t) - q(t)$ and the sliding variable error $r = \dot{e} + \lambda e$ with $\lambda = \lambda^T > 0$. Define the nonlinear robot function, $f(x) = M(q)(\ddot{q}_d + \lambda\dot{e}) + V_m(q, \dot{q})(\dot{q}_d + \lambda e) + G(q) + F(\dot{q})$, where the known vector $x(t)$ of measured signals is selected as, $x := [e^T \ \dot{e}^T \ q_d^T \ \dot{q}_d^T \ \ddot{q}_d^T]^T$.

NN Controller for Continuous-Time Systems

The NN controller is designed based on *functional approximation properties* of NN as shown in Lewis et al. (1999). Thus assume that $f(x)$ can be approximated by $\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x)$ with $\hat{V}$, $\hat{W}$ the estimated NN weights. Select the control input, $\tau = \hat{W}^T \sigma(\hat{V}^T x) + K_v r - v$ with K_v a symmetric positive definite gain and $v(t)$ a robustifying function. This NN control structure is shown in Fig. 1. The outer proportional-derivative (PD) tracking loop guarantees robust behavior. The inner loop containing the NN is known as a feedback linearization loop, and the NN effectively learns the unknown dynamics online to cancel the nonlinearities of the system. Let the estimated sigmoid Jacobian be $\hat{\sigma}' \equiv \frac{d\sigma(z)}{dz}|_{z=\hat{V}^T x}$. Then the NN weight tuning laws are provided by



Neuro-inspired Control, Fig. 1 Neural network robot controller

$$\begin{aligned}\dot{\hat{W}} &= F\hat{\sigma}r^T - F\hat{\sigma}'\hat{V}_{xr}^T - kF\|r\|\hat{W}, \\ \dot{\hat{V}} &= Gx(\hat{\sigma}\hat{W}r)^T - kG\|r\|\hat{V},\end{aligned}$$

with any constant symmetric matrices $F, G > 0$, and scalar tuning parameter $k > 0$.

NN Controller for Discrete-Time Systems

Most feedback controllers today are implemented on digital computers. This requires the specification of control algorithms in discrete-time or digital form (Lewis et al. 1999). To design such controllers, one may consider the discrete-time dynamics $x_{k+1} = f(x_k) + g(x_k)u_k$ with unknown functions $f(\cdot), g(\cdot)$. The digital NN controller derived in this situation has the form of a feedback linearization controller shown in Fig. 1. One can derive tuning algorithms, for a discrete-time neural network controller with L layers, that guarantee system stability and robustness (Lewis et al. 1999). For the i -th layer the weight updates are of the form,

$$\begin{aligned}\hat{W}_i(k+1) &= \hat{W}_i(k) - \alpha_i \hat{\phi}_i(k) \hat{y}_i^T(k) - \Gamma \\ &\quad \left\| I - \alpha_i \hat{\phi}_i(k) \hat{\phi}_i(k)^T \right\| \hat{W}_i(k)\end{aligned}$$

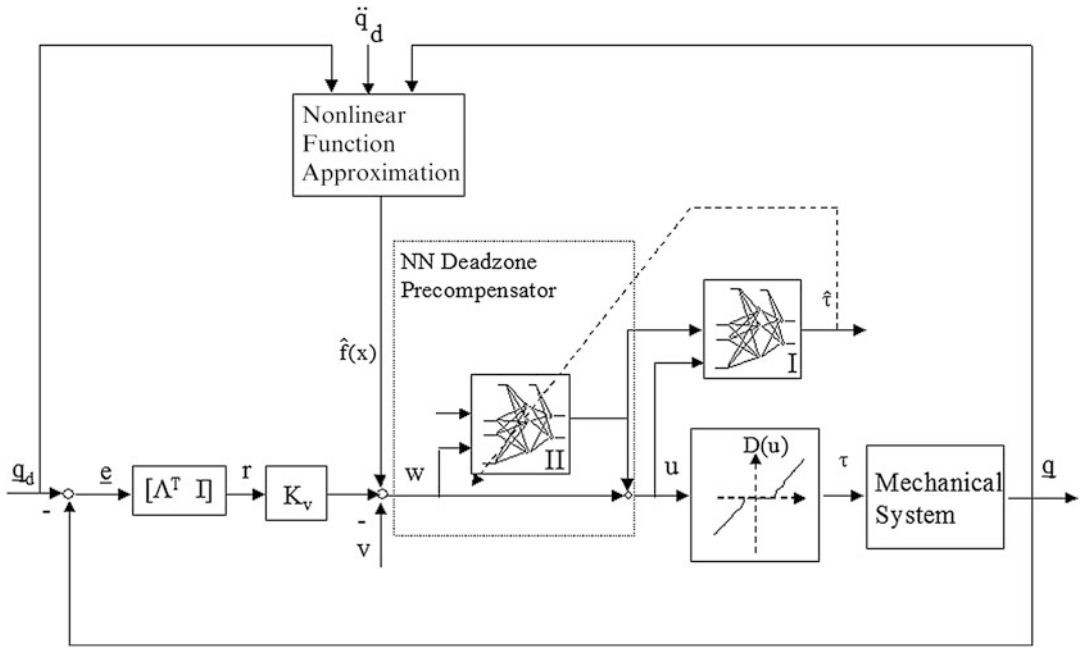
where $\hat{\phi}_i(k)$ are the output functions of layer i , $0 < \Gamma < 1$ is a design parameter and

$$\hat{y}_i(k) = \begin{cases} \hat{W}_i^T \hat{\phi}_i(k) + K_v r(k) & \text{for } i = 1, \dots, L-1, \\ r(k+1) & \text{for } i = L \end{cases}$$

with $r(k)$ a filtered error.

Feed-Forward Neurocontroller

Industrial, aerospace, DoD, and MEMS assembly systems have actuators that generally contain deadzone, backlash, and hysteresis. Since these actuator nonlinearities appear in the feed-forward loop, the NN compensator must also appear in the feedforward loop. This design is significantly more complex than for feedback NN controllers. Details are given in Lewis et al. (2002). Feed-forward controllers can offset the effects of deadzone if properly designed. It can be shown that a NN deadzone compensator has the structure shown in Fig. 2. The NN compensator consists of two NNs. NN II is in the direct feed-forward control loop, and NN I is not directly in the control loop but serves as an observer to estimate the (unmeasured) applied torque $\tau(t)$. The feedback stability and performance of the NN deadzone compensator have been rigorously proven using nonlinear stability proof techniques. The two NN were each selected as having one tunable layer, namely, the output weights. The activation functions were set as a basis by selecting fixed random values for the first-layer weights. To guarantee



Neuro-inspired Control, Fig. 2 Feed-forward NN for deadzone compensation

stability, the output weights of the inversion NN II (subscript i denotes weights and sigmoids of the inversion) and the estimator NN I should be tuned, respectively, as

$$\begin{aligned}\dot{\hat{W}}_i &= T\sigma_i(V_i w)r^T \hat{W}^T \sigma'_i(V^T u)V^T \\ &\quad - k_1 T \|r\| \hat{W}_i - k_2 T \|r\| \|\hat{W}_i\| \hat{W}_i, \\ \dot{\hat{W}} &= -S\sigma'(V^T u)V^T \hat{W}_i \sigma_i(V_i^T w)r^T \\ &\quad - k_1 S \|r\| \hat{W},\end{aligned}$$

with design matrices $T, S \succ 0$ and tuning gains k_1, k_2 .

A continuous Robust Integral of the Sign of the Error structure has been used in Patre et al. (2008) to compensate for the unstructured uncertainties.

Approximate Dynamic Programming for Feedback Control

The current status of work in approximate dynamic programming (ADP) for feedback

control is given in Lewis and Liu (2012). ADP is a form of reinforcement learning (RL) based on an actor/critic structure (Bertsekas and Tsitsiklis 1996). RL is a class of methods used in machine learning to methodically modify the actions of an agent based on observed responses from its environment (Sutton and Barto 1998). The actor/critic structures are RL systems that have two learning structures: A critic network evaluates the performance of a current action policy, and based on that evaluation, an actor structure updates the action policy. Adaptive optimal controllers (Lewis et al. 2012) have been proposed by adding optimality criteria to an adaptive controller or adding adaptive characteristics to an optimal controller.

Optimal Adaptive Control of Discrete-Time Nonlinear Systems

Consider a class of discrete-time systems described by the deterministic nonlinear dynamics in the affine state space difference equation form

$$x_{k+1} = f(x_k) + g(x_k)u_k, \quad (2)$$

with state $x_k \in \mathbb{R}^n$ and control input $u_k \in \mathbb{R}^m$. A deterministic control policy is defined as a function from state space to control space $\mathbb{R}^n \rightarrow \mathbb{R}^m$. That is, for every state x_k , the policy defines a control action $u_k = h(x_k)$ as a feedback controller. Define a deterministic cost function that yields the value function $V(x_k) = \sum_{i=k}^{\infty} \gamma^{i-k} r(x_i, u_i)$, with $0 < \gamma \leq 1$ a discount factor, $Q(x_k)$, $R >$ and $u_k = h(x_k)$ a prescribed feedback control policy. The optimal value is given by Bellman's optimality equation, $V^*(x_k) = \min_{h(\cdot)} (r(x_k, h(x_k)) + \gamma V^*(x_{k+1}))$, which is the discrete-time Hamilton-Jacobi-Bellman (HJB) equation. Two forms of RL can be based on policy iteration (PI) and value iteration (VI). For temporal difference learning, PI is written as follows in terms of the deterministic Bellman equation. where ϵ_{ac} is a small number that checks the algorithm convergence. Value iteration is similar, but the policy evaluation procedure is performed as $V_{i+1}(x_k) = r(x_k, h_i(x_k)) + \gamma V_i(x_{k+1})$. In value iteration we can select any initial control policy, not necessarily admissible or stabilizing. In the control system, the critic and the actor NNs are tuned online using the observed data $(x_k, x_{k+1}, r(x_k, h_i(x_k)))$ along the system trajectory. The critic and actor are tuned sequentially in both the PI and the VI. That is, the weights of one neural network are held constant, while the weights of the other are tuned until convergence. This procedure is repeated until both neural networks have converged. Thus, the controller learns the optimal controller online. The convergence of value iteration using two neural networks for the discrete-time nonlinear

system (2) is proven in Al-Tamimi et al. (2008). Design of an ADP controller that uses only output feedback is given in Lewis and Vamvoudakis (2011).

Optimal Adaptive Control of Continuous-Time Nonlinear Systems

RL is considerably more difficult for continuous-time systems than for discrete-time systems, and fewer results are available. This subsection will provide the formulation of optimal control problem followed by an offline PI algorithm provided in Abu-Khalaf and Lewis (2005) that will give us the structure for the proposed online algorithms that follow. Consider the following nonlinear time-invariant affine in the input dynamical system given by

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t); x(0) = x_0,$$

where $x(t) \in \mathbb{R}^n$, $f(x(t)) \in \mathbb{R}^n$, $g(x(t)) \in \mathbb{R}^{n \times m}$ and control input $u(t) \in \mathbb{R}^m$. We assume that $f(0) = 0$, $f(x) + g(x)u$ is Lipschitz continuous on a set $\Omega \subset \mathbb{R}^n$ that contains the origin and that the system is stabilizable on Ω , that is, there exists a continuous control function $u(t) \in U$ such that the system is asymptotically stable on Ω . Define the infinite horizon integral cost $\forall t \geq 0$

$$V(x_t) = \int_t^{\infty} r(x(\tau), u(\tau)) d\tau, \quad (3)$$

with $Q(x)$ positive definite and $R \in \mathbb{R}^{m \times m}$ a symmetric positive definite matrix. For any admissible control policy if the associated cost (3) is C^1 , then an infinitesimal version is the Bellman equation, and the optimal cost function $V^*(x)$ is defined by $V^*(x_0) = \min_u \int_0^{\infty} r(x, u) d\tau$, which satisfies the HJB equation. By employing the stationarity condition, the optimal control function for the given problem is

$$u^*(x) = -\frac{1}{2} R^{-1} g^T(x) \frac{\partial V^*(x)}{\partial x}. \quad (4)$$

Algorithm 1 PI for discrete-time systems

- 1: **procedure**
 - 2: Given admissible policies $h_0(x_k)$.
 - 3: **while** $\|V^{h_i} - V^{h_{i-1}}\| \geq \epsilon_{ac}$ **do**
 - 4: Solve for the value $V_{(i)}(x)$ using $V_{i+1}(x_k) = r(x_k, h_i(x_k)) + \gamma V_i(x_{k+1})$.
 - 5: Update the control policy $h_{(i+1)}(x_k)$ using $h_{i+1}(x_k) = \arg \min_{h(\cdot)} (r(x_k, h(x_k)) + \gamma V_{i+1}(x_{k+1}))$.
 - 6: $i := i + 1$
 - 7: **end while**
 - 8: **end procedure**
-

Inserting the optimal control (4) into the Bellman equation one obtains the formulation for HJB equation in terms of $\frac{\partial V^*(x)}{\partial x}$ and with boundary condition $V^*(0) = 0$,

$$0 = r(x, u^*) + \frac{\partial V^*(x)}{\partial x} (f(x) + g(x)u^*), \quad (5)$$

which for the linear case becomes the well-known Riccati equation. In order to find the optimal control solution for the problem, one needs to solve the HJB equations (5) for the value function and then substitute in (4) to obtain the optimal control. However, due to the nonlinear nature of the HJB equation finding, its solution is generally difficult or impossible. The following PI algorithm is an iterative algorithm for solving optimal control problems and will give us the structure for the online learning algorithm: A PI algorithm that solves online

Algorithm 2 PI for continuous-time systems

- 1: **procedure**
 - 2: Given admissible policies $u^{(0)}$.
 - 3: **while** $\|V^{u^{(i)}} - V^{u^{(i-1)}}\| \geq \epsilon_{ac}$ **do**
 - 4: Solve for the value $V^{(i)}(x)$ using Bellman's equation, $Q(x) + \frac{\partial V^{u^{(i)}}}{\partial x} (f(x) + g(x)u^{(i)}) + u^{(i)T} R u^{(i)} = 0$, $V^{u^{(i)}}(0) = 0$.
 - 5: Update the control policy $u^{(i+1)}$ using, $u^{(i+1)} = -\left(\frac{1}{2}R^{-1}g^T(x)\frac{\partial V^{u^{(i)}}}{\partial x}\right)^T$.
 - 6: $i := i + 1$
 - 7: **end while**
 - 8: **end procedure**
-

the HJB equation without full information of the plant dynamics is proposed in Vrabie et al. (2009) where the Bellman equation is proved to be equivalent to the *integral RL form* with an optimal value given for some $T > 0$ as, $V^*(x(t)) = \arg \min_u \int_t^{t+T} r(x(\tau), u(\tau))d\tau + V^*(x(t+T))$. Therefore, the temporal difference error for continuous-time systems can be defined as $e(t : t+T) = \rho(t : t+T) + V(x(t+T)) - V(x(t))$, with $\rho(t : t+T) \equiv \int_t^{t+T} r(x(\tau), u(\tau))d\tau$ without any information of the plant dynamics. The IRL controller just given tunes the critic neural

network to determine the value while holding the control policy fixed. The IRL algorithm can be implemented online by RL techniques using value function approximation $\hat{V}(x) = \hat{W}_1^T \phi(x)$ in a critic approximator network. Using that approximation in the PI algorithm, one can use batch least squares or recursive least squares to update the value function, and then on convergence of the value parameters, the action is updated. The work in Vamvoudakis and Lewis (2010) presents a way of finding the optimal control solution in a synchronous manner along with stability and convergence guarantees but with known dynamics. This procedure is more nearly in line with accepted practice in adaptive control. Synchronous online learning algorithms that avoids the knowledge of the dynamics are proposed in Kamalapurkar et al. (2018) and Vamvoudakis et al. (2014).

On-Policy Versus Off-Policy Regulation and Tracking Methods

Finally, algorithms that are used to solve optimal control problems are categorized in two classes of neuro-control methods, namely, on-policy and off-policy methods. On-policy methods evaluate or improve the same policy as the one that is used to make decisions. In off-policy methods, these two functions are separated. The policy used to generate data, called the behavior policy, may in fact be unrelated to the policy that is evaluated and improved, called the estimation policy or target policy. The learning process for the target policy is online, but the data used in this step can be obtained offline by applying the behavior policy to the system dynamics. The off-policy methods are data efficient and fast since a stream of experiences obtained from executing a behavior policy is reused to update several value functions corresponding to different estimation policies. Moreover, off-policy algorithms (Jiang and Jiang 2017) take into account the effect of probing noise needed for exploration. A survey on on-policy and off-policy methods for regulation and tracking is provided in our recent work in Kiumarsi et al. (2018) and Vamvoudakis et al. (2015).

Games

RL techniques have been applied to design adaptive controllers that converge to the solution of two-player zero-sum games in Vamvoudakis and Lewis (2012) and Vrabie et al. (2012), of multi-player nonzero-sum games in Vamvoudakis and Lewis (2011a, b) and Vamvoudakis et al. (2012), and of Stackelberg games in Vamvoudakis et al. (2019). In these cases, the adaptive control structure has multiple loops, with action networks and critic networks for each player. The adaptive controller for zero-sum games finds the solution to the H-infinity control problem online in real time. This adaptive controller does not require any systems dynamics information. The magazine article (Vamvoudakis et al. 2017) presents a family of algorithms that show how to solve multiplayer games online by adaptive learning in real time using data measured along the trajectories of the players. The full dynamics of the players do not need to be known for these online solution techniques. This is a truly dynamic framework for team decision-making, since players or teams can change their objectives or optimality criteria on the fly, and the new strategies for all players, appropriate to the new situation, are then recomputed in real time.

Summary and Future Directions

This book entry discusses some neuro-inspired control techniques. These controllers have multi-loop, multi-timescale structures and can learn the solutions to Hamilton-Jacobi design equations such as the Riccati equation online without knowing the full dynamical model of the system. An interesting and important open research direction is to develop novel neuro-inspired approaches for feedback control of nonlinear systems with high-dimensional inputs and unstructured input data, i.e., by using deep neural networks. Deep neural networks can approximate a more accurate structure of the value function and avoid divergence of the algorithm and consequently instability of the feedback control system.

Cross-References

- [Adaptive Control: Overview](#)
- [Multiagent Reinforcement Learning](#)
- [Optimal Control and Mechanics](#)

Acknowledgments This material is based upon the work supported by NSF CPS-1851588, ECCS-1839804, SATC-1801611, by Minerva Research Initiative N00014-18-1-2160, and by ONR N00014-17-1-2239, N00014-18-1-2221.

Bibliography

- Abu-Khalaf M, Lewis FL (2005) Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. *Automatica* 41(5):779–791
- Al-Tamimi A, Lewis FL, Abu-Khalaf M (2008) Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. *IEEE Trans Syst Man Cybern Part B* 38(4):943–949
- Alessandri A, Baglietto M, Parisini T, Zoppoli R (1999) A neural state estimator with bounded errors for nonlinear systems. *IEEE Trans Autom Control* 44(11):2028–2042
- Bertsekas DP, Tsitsiklis JN (1996) *Neuro-dynamic programming*, vol 5. Athena Scientific, Belmont
- Ioannou PA, Fidan B (2006) *Adaptive control tutorial*. Advances in design and control. SIAM
- Jiang Y, Jiang ZP (2017) *Robust adaptive dynamic programming*. Wiley
- Kamalapurkar R, Walters PS, Rosenfeld JA, Dixon WE (2018) Reinforcement learning for optimal feedback control: a Lyapunov-based approach. Springer
- Kiumarsi B, Vamvoudakis KG, Modares H, Lewis FL (2018) Optimal and autonomous control using reinforcement learning: a survey. *IEEE Trans Neural Netw Learn Syst* 29(6):2042–2062
- Kosmatopoulos EB, Polycarpou MM, Christodoulou MA, Ioannou PA (1995) High-order neural network structures for identification of dynamical systems. *IEEE Trans Neural Netw* 6(2):422–431
- Lewis FL, Liu D (2012) Reinforcement learning and approximate dynamic programming for feedback control. IEEE Press computational intelligence series
- Lewis FL, Vamvoudakis KG (2011) Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data. *IEEE Trans Syst Man Cybern Part B* 41(1):14–25
- Lewis FL, Jagannathan S, Yesildirek A (1999) *Neural network control of robot manipulators and nonlinear systems*. Taylor and Francis, London

- Lewis FL, Campos J, Selmic R (2002) Neuro-fuzzy control of industrial systems with actuator nonlinearities. Society of Industrial and Applied Mathematics Press, Philadelphia
- Lewis FL, Vrabie D, Vamvoudakis KG (2012) Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers. *IEEE Control Syst Mag* 32(6):76–105
- Patre PM, MacKunis W, Kaiser K, Dixon WE (2008) Asymptotic tracking for uncertain dynamic systems via a multi layer neural network feedforward and RISE feedback control structure. *IEEE Trans Autom Control* 53(9):2180–2185
- Slotine JJE, Li W (1987) On the adaptive control of robot manipulators. *Int J Robot Res* 6(3):49–59
- Sutton RS, Barto AG (1998) Reinforcement learning an introduction. MIT Press, Cambridge, MA
- Vamvoudakis KG, Lewis FL (2010) Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica* 46(5):878–888
- Vamvoudakis KG, Lewis FL (2011a) Multi-player non zero sum games: online adaptive learning solution of coupled Hamilton-Jacobi equations. *Automatica* 47(8):1556–1569
- Vamvoudakis KG, Lewis FL (2011b) Policy iteration algorithm for distributed networks and graphical games. In: *Proceedings of 50th IEEE conference on decision and control*, Orlando, pp 128–135
- Vamvoudakis KG, Lewis FL (2012) Online solution of nonlinear two-player zero-sum games using synchronous policy iteration. *Int J Robust Nonlinear Control* 22(13):1460–1483
- Vamvoudakis KG, Lewis FL, Hudas GR (2012) Multi-agent differential graphical games: online adaptive learning solution for synchronization with optimality. *Automatica* 48(8):1598–1611
- Vamvoudakis KG, Vrabie D, Lewis FL (2014) Online adaptive algorithm for optimal control with integral reinforcement learning. *Int J Robust Nonlinear Control* 24(17):2686–2710
- Vamvoudakis KG, Antsaklis PJ, Dixon WE, Hespanha JP, Lewis FL, Modares H, Kiumarsi B (2015) Autonomy and machine intelligence in complex systems: a tutorial. In: *Proceedings of American control conference*, Chicago, pp 5062–5079
- Vamvoudakis KG, Modares H, Kiumarsi B, Lewis FL (2017) Game theory-based control system algorithms with real-time reinforcement learning: how to solve multiplayer games online. *IEEE Control Syst Mag* 37(1):33–52
- Vamvoudakis KG, Lewis FL, Dixon WE (2019) Open-loop Stackelberg learning solution for hierarchical control problems. *Int J Adapt Control Signal Process* 33(2):285–299
- Vrabie D, Pastravanu O, Lewis FL, Abu-Khalaf M (2009) Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica* 45(2):477–484

Vrabie DL, Vamvoudakis KG, Lewis FL (2012) Optimal adaptive control and differential games by reinforcement learning principles. *Control engineering series*. IET Press

Werbos PJ (1989) Neural networks for control and system identification. In: *Proceedings of IEEE conference decision and control*, Orlando

Noise Spectra in MEMS Coriolis Vibratory Gyros

Robert T. M'Closkey

Mechanical and Aerospace Engineering,
Samueli School of Engineering and Applied
Science, University of California, Los Angeles,
CA, USA

Abstract

Coriolis vibratory gyros possess vibrating elements whose motion is modified due to rotation of the sensor. Maturation of wafer-scale fabrication techniques has led to the production of a diverse family of sensors based around miniature resonators with integrated pick-offs, forcers, and signal conditioning electronics. This entry summarizes the common features found in the noise spectra of the angular rate estimates obtained from such devices. The angle random walk and angle white noise figures of merit are described, and the noise spectra relation to the Allan variance is also discussed.

Keywords

Microelectromechanical systems ·
Resonators · Solid state gyro · Coriolis
vibratory gyro · Noise spectra

Introduction

Wafer processing techniques have matured to the point where miniature, multi-degree-of-freedom, precision mechanical structures can be routinely fabricated. Due to the challenge

of fabricating structures with freely rotating elements at small scales, sensitivity to angular motion is achieved through the coupling to two resonant modes in an elastic structure. Thus, all microelectromechanical (MEMS) gyros fall into a class of angular rate sensors called Coriolis vibratory gyros, or CVGs. A variety of MEMS CVG designs have emerged, from traditional tuning forks fabricated from crystalline quartz to lumped-mass/stiffness structures and distributed-mass/stiffness axisymmetric resonators fabricated from silicon and fused silica. A great deal of effort is involved in designing the structures to tailor the mode shapes and modal frequencies of the pair of modes which are exploited for angular rate sensing while simultaneously minimizing the influence of other modes which inevitably appear. Examples of resonators that have been produced by the research community include planar rings (Ayazi and Najafi 2001; Shcheglov and Challoner 2006; Yang et al. 2015; Su et al. 2014; Ge and M'Closkey 2017), hemispheres and hemi-toroids (Bernstein et al. 2013; Prikhodko et al. 2011; Cho et al. 2014; Horning et al. 2014), and tuning forks (Weinberg and Kourepenis 2006; Trusov et al. 2011). Axisymmetric designs typically produce modal degeneracy, whereas tuning fork-type resonators exhibit some separation in the modal frequencies of the modes of interest.

CVG performance is determined by the noise spectra of the extracted angular rate signals. This entry focuses on case where external noise sources exist at the input and output of the resonator's electromechanical transfer function. This captures many sources of noise including contributions from buffering electronics, which are naturally modeled as appearing at the transfer function output, and mechanical-thermal noise (Gabrielson 1993), digital-to-analog converter noise, and vibration of the sensor case, which are examples of sources appearing at the transfer function input. Thus, the presentation is somewhat generic. One aspect which is not included in the analysis is the modeling of sources of long-term drift in the sensor offset (bias) signals. This is subject of intense research and must be approached separately

for each technology since it can depend on the resonator packaging, resonator material, thermal environment, residual stress, package vacuum level and stability, and additional factors.

Resonator Model

Ignoring all but the modes of interest, the differential equation written in sensor case-fixed coordinates is described by

$$M\ddot{x} + (D + \Omega\alpha S)\dot{x} + Kx = f \quad (1)$$

where M , D , and K are 2×2 positive definite mass, damping, and stiffness matrices, respectively,

$$M = \begin{bmatrix} m_1 & m_3 \\ m_3 & m_2 \end{bmatrix}, \quad D = \begin{bmatrix} d_1 & d_3 \\ d_3 & d_2 \end{bmatrix}, \quad K = \begin{bmatrix} k_1 & k_3 \\ k_3 & k_2 \end{bmatrix},$$

Ω is the angular rotation rate of the sensor case, S is the skew-symmetric matrix

$$S = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

α captures the "strength" of the Coriolis coupling, which depends on the resonator design, and $f = [f_1, f_2]^T$ are the forces acting on the two degrees of freedom x_1 and x_2 that comprise the vector x . Noise sources will be added to the model later. In actual resonators, measurements of each coordinate are available and, depending on the transduction method, may be proportional to x or $\dot{x}$. Furthermore, it is assumed that the forces f are co-located with the coordinates. In an ideal CVG, the mass, stiffness, and damping matrices in (1) are diagonal, and so the two coordinates are only coupled through the Coriolis term. In practice, the off-diagonal terms in these matrices are non-zero and are the source of various *in-phase* and *quadrature* biases, or offsets, in the angular rate estimate extracted from the sensor. Nevertheless, it can be assumed that the coordinates are nearly decoupled, with the exception of the Coriolis term, and so that it makes sense to refer to *the* resonant mode associated

with a given channel, i.e., x_1/f_1 channel has only one resonant mode with frequency $\omega_1 := \sqrt{k_1/m_1}$, the x_2/f_2 channel has another resonant mode with frequency $\omega_2 := \sqrt{k_2/m_2}$, and the “cross-channels” x_1/f_2 and x_2/f_1 are essentially zero.

MEMS CVGs operate in one of two modes: an open-loop configuration in which the rotation-induced response is measured in one degree of freedom or a closed-loop configuration in which the rotation-induced response is attenuated with feedback. In the latter case, the control effort is related to the rotation rate. Whichever configuration is used, though, the companion degree of freedom is forced into a fixed amplitude sinusoidal response. The excitation of this degree of freedom is necessary for the sensor to function since its motion acts as a carrier signal whose amplitude is modulated by the sensor’s angular rate. It will be assumed that the x_1 coordinate is used for the sensor excitation. In order to build an appreciable amplitude in this coordinate, a feedback loop is used to drive the x_1/f_1 channel resonant mode at its natural frequency. The controllers for this task are typically phase-locked loops or automatic gain control filters. The design of these controllers is standard and do not present any particular challenges. It will be assumed that the excitation controller specifies f_1 such that $x_1 = a \cos(\omega_1 t)$, where ω_1 is close to the natural frequency of the x_1/f_1 channel resonant mode. This effectively removes the x_1 coordinate from the analysis. The differential equation for the second coordinate is

$$m_2 \ddot{x}_2 + d_2 \dot{x}_2 + k_2 x_2 = f_2 + d \quad (2)$$

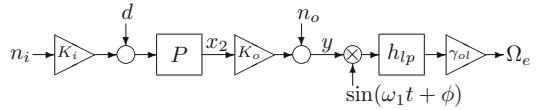
where d is the “disturbance” to the x_2 coordinate created by the excitation loop driving the x_1 coordinate,

$$d = a(\omega_1^2 m_3 - k_3) \cos(\omega_1 t) + a\omega_1 (d_3 - \alpha \Omega(t)) \sin(\omega_1 t). \quad (3)$$

The transfer function P is defined as $P := 1/(m_2 s^2 + d_2 s + k_2)$.

Open-Loop CVG

The open-loop CVG does not use feedback on the x_2 coordinate so $f_2 = 0$. The block diagram is shown below:



where noise sources n_i and n_o , which are assumed to be independent, have been added at the input and output, respectively, of P , and the transduction gains K_o and K_i merely convert the resonator motion into a voltage (unit V) or a voltage into a force. Without loss of generality, the noise sources are located at points where they are expressed in volts. The scale factor is determined assuming the noises are zero and the angular rate is constant, $\bar{\Omega}$,

$$x_2(t) = |P(j\omega_1)|a(\omega_1^2 m_3 - k_3) \cos(\omega_1 t + \angle P(j\omega_1)) + |P(j\omega_1)|a\omega_1 (d_3 - \alpha \bar{\Omega}) \sin(\omega_1 t + \angle P(j\omega_1)) \quad (4)$$

The angular rate is recovered by synchronous demodulation of $y = K_o x_2$: it is multiplied by $\sin(\omega_1 t + \phi)$, where the phase $\phi = \angle P(j\omega_1)$ is chosen to maximize the component with $\bar{\Omega}$ and then low-pass filtered with h_{lp} to produce the “DC” signal, $(K_o x_2(t) \sin(\omega_1 t + \phi))$

$$= \frac{1}{2} K_o |P(j\omega_1)|a\omega_1 (d_3 - \alpha \bar{\Omega})$$

where $*$ denotes convolution. The term multiplying $\bar{\Omega}$ is the inverse of the *open-loop scale factor*,

$$\gamma_{ol}^{-1} = -\frac{1}{2} K_o |P(j\omega_1)|a\omega_1 \alpha.$$

Multiplication of the demodulated signal by γ_{ol} produces the constant rate $\bar{\Omega}$ plus the *in-phase* bias $-d_3/\alpha$. This bias term, which is due to dissipation in the resonator, is indistinguishable

from an applied angular rate. The quadrature bias is (ideally) perfectly rejected by the synchronous demodulation. In practice, the bias terms slowly drift with time due to slowly changing resonator dynamics. The root causes of such drift, and how to mitigate them, is an active area of research in MEMs CVG development. The synchronous demodulation can also be subject to phase errors which means that the quadrature bias can leak into the in-phase measurement, thereby adding further sources of drift. In terms of noise analysis, though, these slow drifts impact the low frequency portion of the angular rate noise spectrum and are not the subject of analysis in this entry.

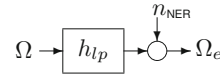
When a sinusoidal angular rate $\Omega(t) = \Omega_0 \cos(\tilde{\omega}t)$ is applied to the sensor, the modulation of Ω by $\sin(\omega_1 t)$ in (3) produces sidebands at frequencies $\omega_1 \pm \tilde{\omega}$. The amplitude and phase response of x_2 should be essentially equal at these frequencies to ensure that the scale factor is not dependent on the frequency of the angular rate input over the range of interest. In other words it is desired $P(j\omega_1 + \tilde{\omega}) \approx P(j\omega_1 - \tilde{\omega}) \approx P(j\omega_1)$; therefore, ω_1 should be somewhat smaller than ω_2 so that ω_1 resides in the “flat” portion of the frequency response of P . This is the motivation in open-loop CVGs to design the modal frequencies so that there is a substantial difference between them and with the frequencies ordered such that $\omega_1 < \omega_2$. The bandwidth of estimated angular rate signal is then established by the low-pass filter h_{lp} that is used in the demodulator. The bandwidth of h_{lp} is denoted ω_{lp} .

The mean-square spectral density of y , denoted S_y , due to noise sources n_i and n_o is given by $S_y = S_o + |K_o P K_i|^2 S_i$, where S_i and S_o are the spectral densities of n_i and n_o , respectively. It is assumed that S_i and S_o are constant in the frequency band $[\omega_1 - \omega_{lp}, \omega_1 + \omega_{lp}]$, thus, S_y is also constant in this band since $|K_o P K_i| \approx |K_o P(j\omega_1) K_i|$ over this band. In practice, the spectrum of y is directly measured with the effect of the noise sources aggregated into a single value for the root spectral density which will be denoted v_o . Demodulating y produces the spectrum $\frac{1}{4} (S_y(\omega_1 + \omega) + S_y(\omega_1 - \omega)) = \frac{1}{2} v_o^2$ at the output of h_{lp} and the subsequent multiplication

by the scale factor yields the following expression for the *noise-equivalent rate*, or S_{NER} ,

$$S_{\text{NER}}(\omega) = \frac{1}{2} \gamma_{ol}^2 v_o^2 \quad \omega \in [0, \omega_{lp}] \quad (5)$$

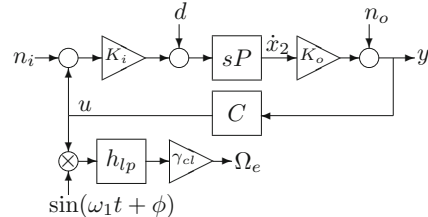
The open-loop CVG model can be summarized as follows:



where the spectral density of n_{NER} is given by (5).

Closed-Loop CVG

Unlike open-loop CVGs in which $\omega_1 < \omega_2$, closed-loop CVGs are designed so that $\omega_1 = \omega_2$ (a condition of *modal degeneracy*). The motivation is to create a relatively large mechanical response of the x_2 coordinate due to angular rate inputs. This occurs because the angular rate-modulated term $\Omega(t) \sin(\omega_1 t)$ in (3) drives the x_2 coordinate with a spectrum that is in a neighborhood of the modal frequency, ω_2 , which has the effect of increasing the signal-to-noise ratio of y with respect to the output noise source, n_o . The response of x_2 in a neighborhood of ω_2 is frequency dependent, though, especially in resonators with high quality factors, so some kind of equalization is necessary – the equalizing filter is essentially an inverse of the plant, which is most conveniently achieved using feedback. Thus, the closed-loop CVG operates by feeding back y to null the effect of the angular rate on the response of the x_2 coordinate as shown in the block diagram below



For sufficiently high loop gain, the control effort u cancels d ; thus, u is demodulated in order to

recover a signal proportional to Ω . The feedback also keeps the x_2 coordinate in a linear regime with respect to electrostatic forces and the elastic material properties, which may not be the case if the modally degenerate resonator is operated open-loop. Another difference with the open-loop CVG is associated with the plant model itself. The open-loop CVG often uses a charge amplifier if the sensing pick-offs are capacitive or piezoelectric in nature. This configuration yields a measurement that is proportional to displacement. While it is possible to use a charge amplifier in a closed-loop CVG, a common technique for buffering the capacitive pick-offs is to employ a transresistance amplifier, which produces a measurement proportional to velocity. This is an advantage in closed-loop CVGs because the feedback filter is designed to dampen the x_2 coordinate resonance; if displacement is measured, then the controller has to provide the necessary phase-lead filtering; however, if velocity is measured, then the controller is a simpler filter since the phase-lead is obtained via the measurement. Hence, in the closed-loop CVG, it is assumed that sP is the plant to be controlled. In terms of analyzing the closed-loop system, the controller can be assumed to be a simple gain, C .

Ignoring the noise sources for the moment and assuming a constant angular rate, $\bar{\Omega}$, $u = -K_i^{-1}d$

for sufficiently large loop gain so demodulating u yields

$$\begin{aligned} h_{lp}(t) * (u(t) \sin(\omega_1 t + \phi)) \\ = -\frac{1}{2} K_i^{-1} a \omega_1 (d_3 - \alpha \bar{\Omega}) \end{aligned}$$

where the demodulation phase ϕ is chosen to maximize the $\bar{\Omega}$ component and reject the quadrature term. The closed-loop scale factor is

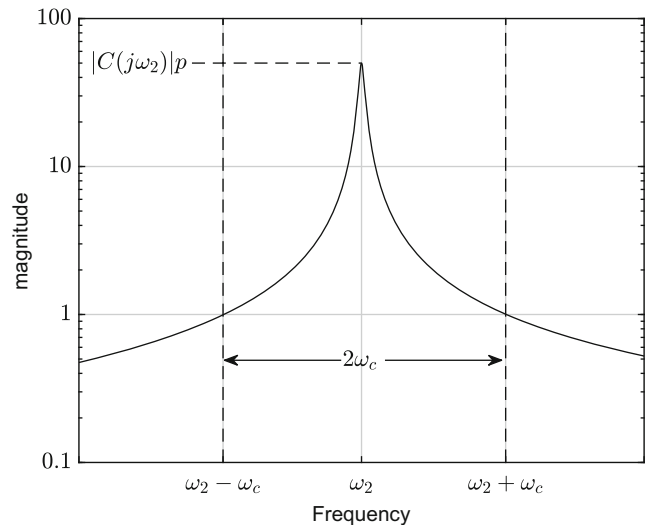
$$\gamma_{cl}^{-1} = \frac{1}{2} K_i^{-1} a \omega_1 \alpha$$

The sensor bandwidth, denoted ω_c , is not determined by the demodulation low-pass filter h_{lp} as it is in the open-loop CVG. Instead it is determined by the frequency interval over which the loop gain $K_o s P K_i C$ has magnitude greater than one (see Fig. 1). The sensor model in this case is $\omega_c/(s + \omega_c)$, where ω_c is defined in Fig. 1.

With the noise sources present, it is assumed again that S_i and S_o are constants; however, it is necessary to distinguish the input and output noise intensities (they cannot be aggregated into a single constant) because of the frequency dependence caused by P ; thus, $S_i(\omega) = v_i^2$ and $S_o(\omega) = v_o^2$, both given in V^2/Hz units. The mean-square spectral density of u , denoted S_u , is given by

Noise Spectra in MEMS Coriolis Vibratory Gyros,

Fig. 1 Loop gain magnitude, $|K_o s P K_i C|$. The closed-loop bandwidth, ω_c , is one half of the interval over which the loop gain magnitude is greater than one



$$S_u = \left| \frac{\tilde{P}C}{1 - \tilde{P}C} \right|^2 v_i + \left| \frac{C}{1 - \tilde{P}C} \right|^2 v_o, \quad (6)$$

where $\tilde{P} = K_o s P K_i C$ denotes the product of the forward path elements. It is possible to rewrite (6) as follows:

$$S_u(\omega) = |C(j\omega_2)|^2 v_o^2 \frac{(\omega_2^2 - \omega^2)^2 + (2\omega_e \omega)^2}{(\omega_2^2 - \omega^2)^2 + (2\omega_c \omega)^2}, \quad (7)$$

where ω_e is defined as the effective bandwidth

$$\omega_e = \omega_m \sqrt{1 + (p v_i / v_o)^2}$$

and ω_m is the mechanical bandwidth of the x_2 mode (the mechanical bandwidth is defined as the product of the damping ratio and modal frequency and so is simply the exponential decay rate of the x_2 mode). The parameter p is the maximum gain of $\tilde{P}$. The effective bandwidth captures the relative importance of the two noise sources: if $\omega_e \rightarrow \omega_m$, then n_i can be ignored in the analysis, whereas if $\omega_e \rightarrow \infty$, then n_o can be ignored in the analysis. Although it v_i does not appear explicitly in the way S_u is written in (7), if $\omega_e \rightarrow \infty$, it can be shown that $S_u \rightarrow v_i^2$ because the closed-loop bandwidth, mechanical bandwidth, maximum gain of $\tilde{P}$, and controller gain $|C(j\omega_2)|$ satisfy the relationship

$\omega_c = \omega_m p |C(j\omega_2)|$. Introducing this additional notation cannot be avoided because if the modal damping goes to zero, then there is always a frequency band around ω_2 where the input noise dominates the noise spectrum of angular rate estimate.

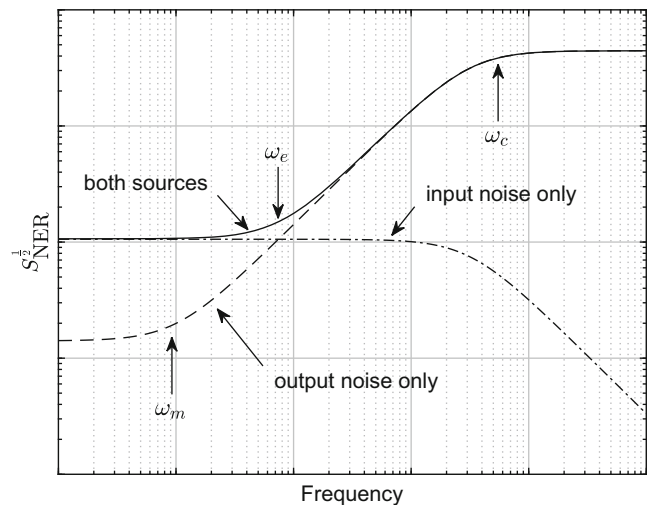
The spectrum of the demodulated signal at the output of h_{lp} is $\frac{1}{4} (S_u(\omega_1 + \omega) + S_u(\omega_1 - \omega))$. In practice, the frequency of the demodulator differs from the modal frequency of the x_2 coordinate, i.e., $\omega_1 \neq \omega_2$; however, this complicates the expression for the noise-equivalent rate S_{NER} ; thus, the special case will be assumed in which $\omega_1 = \omega_2$,

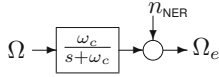
$$\begin{aligned} S_{\text{NER}}(\omega) &= \frac{1}{4} \gamma_{cl}^2 (S_u(\omega_2 + \omega) + S_u(\omega_2 - \omega)) \\ &\approx \frac{1}{2} \gamma_{cl}^2 |C(j\omega_2)|^2 v_o^2 \frac{\omega^2 + \omega_e^2}{\omega^2 + \omega_c^2} \end{aligned} \quad (8)$$

The approximation is valid assuming $\omega_c \ll \omega_1$, which is always satisfied in practice. Figure 2 shows $S_{\text{NER}}^{\frac{1}{2}}$ for the cases when only one noise source is present and also when both noise sources are present and contribute trends to the noise spectrum. The figure also elucidates the roles of ω_m , ω_e and ω_c . Thus, the closed-loop CVG model can be summarized as follows:

Noise Spectra in MEMS Coriolis Vibratory Gyros,

Fig. 2 Noise equivalent rate spectrum in the closed-loop CVG when the modal frequencies are matched

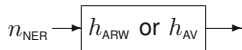




where the spectral density of n_{NER} is given by (8).

Angle Random Walk and Allan Variance Representation

The mean-square spectral density of the angular rate noise defines the CVG performance with regard to the “external” noise sources n_i and n_o . As mentioned earlier, this is not a complete description of the angular rate noise since changes in the resonator itself create low frequency (long period) variations in the angular rate bias/offset signal. Modeling these signals is much more resonator-specific as opposed to the analysis covered here which is agnostic with regard to the particular resonator design. Ignoring the offset variation, it is still useful to convert S_{NER} into alternative metrics that find common use in the navigation community. One such metric is the sensor’s angle random walk (ARW) figure, and another metric is the Allan variance (AV), which is often presented in graphical form. Both ARW and AV can be analyzed using the following block diagram as an aid,



where the impulse response for the ARW computation is a “moving integral” and the impulse response for the AV computation is a time-averaged difference of the rate signal and its time-delayed copy (the averaging interval and time delay being equal),

$$h_{\text{ARW}}(t) = \begin{cases} 1 & t \in [0, \tau] \\ 0 & \text{otherwise} \end{cases},$$

$$h_{\text{AV}}(t) = \begin{cases} \frac{1}{\tau} & t \in [0, \tau] \\ -\frac{1}{\tau} & t \in [\tau, 2\tau] \\ 0 & \text{otherwise} \end{cases}.$$

Here, τ is the integration or averaging interval. The quantity of interest is the variance (denoted

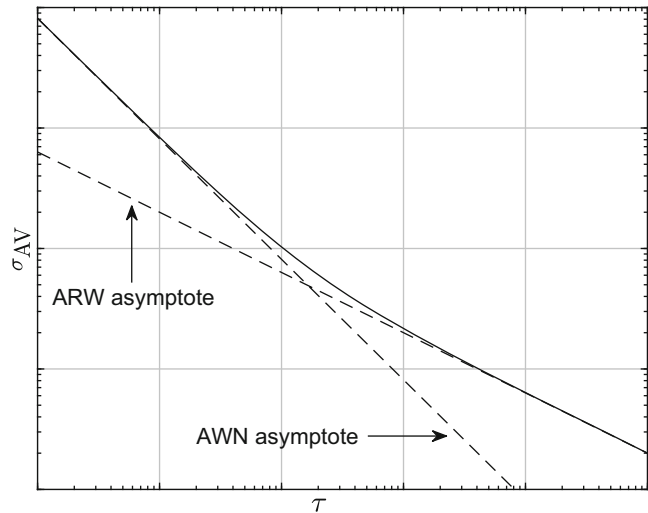
σ_{AV}^2 and σ_{ARW}^2) of the signal produced by such filtering as a function of τ . For the ARW computation, the output has units of angle, whereas the AV computation has units of angular rate. The mean-square value of the output in the ARW computation is given by

$$\sigma_{\text{ARW}}^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{\text{NER}}(\omega) |h_{\text{ARW}}(\omega)|^2 d\omega. \quad (9)$$

The expression for σ_{AV}^2 is obtained by simply replacing h_{ARW} with h_{AV} . Evaluating (9) for the open-loop CVG yields $\sigma_{\text{ARW}}^2 = S_{\text{NER}}(0)\tau$. The ARW figure associated with the CVG is the value of σ_{ARW} when τ is equal to 1 h and represents the standard deviation (uncertainty) associated with computed change in angle of the sensor obtain by integrating its angular rate measurement for 1 h. The expression for σ_{ARW}^2 in the closed-loop CVG has an additional term, $\sigma_{\text{ARW}}^2 = S_{\text{NER}}(0)\tau + S_{\text{NER}}(0)\omega_c/(\omega_e)^2$. The constant term dominates the variance value for integration times $\tau < \omega_c/(\omega_e)^2$ and for this reason is the *angle white noise* (AWN) figure associated with the CVG. AWN is a consequence of the output noise source n_o creating the “phase-lead” profile of S_{NER} . Nevertheless, for sufficiently long integration times, $\sigma_{\text{ARW}}^2 \approx S_{\text{NER}}(0)\tau$, and so an ARW figure can also be quoted the closed-loop CVG.

The Allan variance computation for the open-loop CVG yields $\sigma_{\text{AV}}^2 = S_{\text{NER}}(0)/\tau$, and the closed-loop AV is closely approximated by $\sigma_{\text{AV}}^2 = S_{\text{NER}}(0)\frac{1}{\tau} + \frac{3}{2}S_{\text{NER}}(0)\frac{\omega_c}{\omega_e^2}\frac{1}{\tau^2}$. Common practice is to graph σ_{AV} (Allan deviation) versus τ on log-log axes. This yields the familiar “ $-1/2$ ” slope associated with the ARW for both open-loop and closed-loop CVGs and the “ -1 ” slope associated with the AWN at smaller integration times for the closed-loop CVG. These trends are shown in Fig. 3. In the closed-loop CVG, the asymptotes cross at an integration time of $\tau = \frac{3}{2}\frac{\omega_c}{\omega_e^2}$. These graphs assume that the input noise does not dominate the S_{NER} spectrum in the closed-loop sensor, i.e., the S_{NER} has a phase-lead characteristic and $\omega_e < \omega_c$ in particular. If the input noise does dominate the angular rate noise spectrum, then S_{NER} looks like the spectrum associated with the

Noise Spectra in MEMS Coriolis Vibratory Gyros,
Fig. 3 ARW and AWN trends in the root Allan variance, σ_{AV}



open-loop analysis, and the AWN asymptote is not present in the AV graph.

Summary and Future Directions

This entry focused on the analysis of noise spectra for open- and closed-loop solid state gyros, also called Coriolis vibratory gyros. The resonators in open-loop CVGs are designed so that the two modes exploited for angular rate sensing have separated modal frequencies. The consequence is that the noise-equivalent-rate spectrum is essentially constant across the bandwidth of the sensor. Integrating the angular noise produces an angle estimate whose variance grows linearly with the integration interval – this is the classical angle random walk figure. In order to take advantage of the “mechanical” amplification afforded by a lightly damped resonance, CGV resonators with degenerate modal frequencies require feedback to equalize the response of the sensing degree of freedom due to angular rate inputs. Although the input and output noise sources can be considered to have constant spectral densities, noise-equivalent-rate spectrum typically exhibits a phase-lead characteristic in closed-loop CVGs. In this case, the variance of the integrated rate signal features both angle white noise and angle random walk trends. The

Allan variance is another common tool for capturing CVG performance, and it is shown how the angular noise spectra produce trends in the Allan variance graphs. A major issue in MEMS CVGs is the presence of fairly large zero-input measurement offsets. The magnitude of the offsets is driven by dissipation in the resonator so a direct means of reducing the offset is to fabricate the resonator from low-loss materials and mitigate so-called anchor loss, e.g., Trusov et al. (2011) and Prikhodko et al. (2011). Other approaches augment resonators with additional measurements that correlate with the long-term drift in the offset, e.g., Tatar et al. (2017) and Ge and M’Closkey (2017).

Cross-References

- [Control Systems for Nanopositioning](#)
- [Inertial Navigation](#)

Recommended Reading

Mechanical-thermal noise can play a significant role in limiting the performance of microscale sensors as pointed out by Gabrielson (1993). CVG-specific analysis is given in Leland (2005) and Mohd-Yasin et al. (2009), and a general

overview of noise in MEMS sensors is presented in Annovazzi-Lodi and Merlo (1999). The analysis of the closed-loop configuration in this entry focused on the modally degenerate case; however, more general analysis must also include the effects of inevitable detuning that occurs in physical devices. These details are derived in Kim and M'Closkey (2013).

Bibliography

- Annovazzi-Lodi V, Merlo S (1999) Mechanical-thermal noise in micromachined gyros. *Microelectron J* 30:1227–1230
- Ayazi F, Najafi K (2001) A HARPSS polysilicon vibrating ring gyroscope. *J Microelectromech Syst* 10:169–179
- Bernstein J, Bancu M, Cook E, Chaparala M, Teynor W, Weinberg M (2013) A MEMS diamond hemispherical resonator. *J Micromech Microeng* 23(12):1–8
- Cho JY, Woo J-K, Yan J, Peterson R, Najafi K (2014) Fused-silica micro birdbath resonator gyroscope (μ -BRG). *J Microelectromech Syst* 23(1):66–77
- Gabrielson TB (1993) Mechanical-thermal noise in micro-machined acoustic and vibration sensors. *IEEE Trans Electron Devices* 40(5):903–909
- Ge H, M'Closkey R (2017) Simultaneous angular rate estimates extracted from a single axisymmetric resonator. *IEEE Sensors J* 17:7460–7469
- Horning R, Johnson B, Compton R, Cabuz E (2014) Hemitoroidal resonator gyroscope. U.S. Patent 8 631 702 B2
- Kim D, M'Closkey R (2013) Spectral analysis of vibratory gyro noise. *IEEE Sensors J* 13:4361–4374
- Leland RP (2005) Mechanical-thermal noise in MEMS gyroscopes. *IEEE Sensors J* 5:493–500
- Mohd-Yasin F, Nagel DJ, Korman CE (2009) Noise in MEMS. *Meas Sci Technol* 21:1–22
- Prikhodko I, Zotov S, Trusov A, Shkel A (2011) Microscale glass-blown three-dimensional spherical shell resonators. *J Microelectromech Syst* 20(3):691–701
- Shcheglov KV, Challoner AD (2006) Isolated planar gyroscope with internal radial sensing and actuation. US Patent US7 040 163B2, 9 May 2006
- Su TH, Nitzan SH, Taheri-Tehrani P, Kline MH, Boser BE, Horsley DA (2014) Silicon MEMS disk resonator gyroscope with an integrated CMOS analog front-end. *IEEE Sensors J* 14(10):3426–3432
- Tatar E, Mukherjee T, Fedder GK (2017) Stress effects and compensation of bias drift in a MEMS vibratory-rate gyroscope. *J Microelectromech Syst* 26(3):569–579
- Trusov AA, Prikhodko IP, Zotov SA, Shkel AM (2011) Low-dissipation silicon tuning fork gyroscopes for rate and whole angle measurements. *IEEE Sensors J* 11(11):2763–2770

- Weinberg MS, Kourepenis A (2006) Error sources in in-plane silicon tuning-fork MEMS gyroscopes. *J Microelectromech Syst* 15(3):479–491
- Yang Y, Ng EJ, Chen Y, Flader IB, Kenny TW (2015) Mode-matching of wineglass mode disk resonator gyroscope in (100) single crystal silicon. *J Microelectromech Syst* 24(2):343–350

Noisy Channel Effects on Multi-agent Consensus

Lihua Xie

School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore,
Singapore

Abstract

This article briefly discusses the topic of multi-agent consensus under imperfect communication channels including limited data rate, additive noises, and packet dropouts. The focus is on the design of encoders, decoders, and distributed control protocols so that the multi-agent system can reach consensus.

Keywords

Quantized consensus · Distributed control · Noisy communication channels

Introduction

Multi-agent consensus aims to reach an agreement of certain interested variables within a multi-agent system, where each agent is able to perform autonomous decision-making with sensing, computation, and communication capabilities. Posing as a fundamental problem in cooperative control of multi-agent systems, it has found wide applications in sensor networks, distributed estimation, formation control, smart grid, etc. Based on the communication with only local neighbors, distributed consensus protocols are much more preferable than centralized ones

due to its scalability with respect to the network size and robustness to the node or link failure. In practice, the communication is not perfect and several major issues need to be considered. In this survey, we will introduce the basics of multi-agent consensus under imperfect communication channels including limited data rate, packet dropouts, and additive channel noises. For the sake of an easy illustration, we shall focus on multi-agent systems with integrator dynamics.

To start with, let us briefly recall graph theory which has been an important tool for studying multi-agent systems. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is often used to model the interconnections of agents, where $\mathcal{V} = \{1, 2, \dots, N\}$ is the node set of N agents with $i \in \mathcal{V}$ representing the i -th agent; $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the edge set with each edge representing the information flow among agents. Throughout this survey, we consider the simple case that $\mathcal{G}$ is undirected, i.e., an edge $(j, i) \in \mathcal{E}$ implies that agent i and agent j can communicate with each other. The neighborhood set $\mathcal{N}_i$ of agent i is defined as $\mathcal{N}_i = \{j \neq i \mid (j, i) \in \mathcal{E}\}$. The adjacency weight matrix $\mathcal{A}$ is defined as $\mathcal{A} = [a_{ij}]_{N \times N}$, where $a_{ij} > 0$ iff $j \in \mathcal{N}_i$, and $a_{ij} = 0$ otherwise. The Laplacian matrix $\mathcal{L} = [\mathcal{L}_{ij}]_{N \times N}$ is defined as $\mathcal{L}_{ii} = \sum_{j \in \mathcal{N}_i} a_{ij}$, $\mathcal{L}_{ij} = -a_{ij}$ for $i \neq j$. Therefore, for undirected graphs, $\mathcal{L}$ is symmetric and $\mathcal{L}\mathbf{1} = \mathbf{1}'\mathcal{L} = \mathbf{0}$, where $\mathbf{1}$ is the vector of ones. Moreover, if $\mathcal{G}$ is undirected and connected, zero is a simple eigenvalue of $\mathcal{L}$, and all the other eigenvalues of $\mathcal{L}$ are positive, i.e.,

$$0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_N,$$

where λ_i are eigenvalues of $\mathcal{L}$ sorted in an ascending order.

Suppose we have N agents with single integrator dynamics, i.e.,

$$\dot{x}_i(t) = u_i(t), \quad i = 1, \dots, N. \quad (1)$$

The consensus problem requires us to design distributed control inputs u_i , which relies on x_i and x_j , $j \in \mathcal{N}_i$ only, such that the agents can reach an agreement on the states asymptotically, as shown in the following definition.

Definition 1 The multi-agent system (1) achieves consensus if

$$\lim_{t \rightarrow \infty} \|x_i(t) - x_j(t)\| = 0, \quad \forall i, j = 1, \dots, N. \quad (2)$$

Moreover, if it holds that

$$\lim_{t \rightarrow \infty} \|x_i(t) - \frac{1}{N} \sum_{j=1}^N x_j(0)\| = 0, \quad i = 1, \dots, N \quad (3)$$

then average consensus is achieved.

A simple consensus protocol is the relative neighborhood state error-based consensus protocol, which is given by

$$\dot{u}_i(t) = a \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(t) - x_i(t)), \quad (4)$$

where $a > 0$ is the consensus gain to be selected. With the consensus protocol (4), the stacked agent dynamics of $x = [x_1, \dots, x_N]'$ can be written as

$$\dot{x}(t) = (I - a\mathcal{L})x(t). \quad (5)$$

To show the average consensus with (4), note that we can decompose $x(0)$ as $x(0) = \frac{\mathbf{1}'x(0)}{N}\mathbf{1} + z$, where $\frac{\mathbf{1}'x(0)}{N}\mathbf{1}$ is the orthogonal projection of $x(0)$ onto the subspace spanned by $\mathbf{1}$ and z is orthogonal to $\mathbf{1}$. Moreover, if a is selected to satisfy $0 < a < \frac{2}{\lambda_N}$, $\mathbf{1}$ is a simple eigenvalue of $I - a\mathcal{L}$, and all the other eigenvalues of $I - a\mathcal{L}$ are within the unit circle. Besides, $(I - a\mathcal{L})\mathbf{1} = \mathbf{1}$. In view of the properties of $I - a\mathcal{L}$, we have from the iteration (5) that $\lim_{t \rightarrow \infty} x(t) = \frac{\mathbf{1}'x(0)}{N}\mathbf{1}$. As a result, average consensus is achieved.

However, the information of neighboring agents' states in the protocol (4) is typically obtained through communication. In practice, communication channels are inevitably subject to quantization error, noise, and uncertainty, especially in wireless communications. Thus, when there exists quantization error or noise in communication, the information obtained from

agent j by agent i can be expressed as

$$y_{ij}(t) = x_j(t) + w_{ij}(t); \quad (6)$$

when there exist packet dropouts, the information obtained from agent j by agent i can be expressed as

$$y_{ij}(t) = \gamma_{ij}(t)x_j(t), \quad (7)$$

where $\gamma_{ij}(t)$ is a 0-1 binary random variable.

Quantized Consensus

In a digital network, communication channels have a finite channel capacity, thus, at each time step, agents can only transmit a finite amount of information to their neighbors. Specifically, the sender needs to first encode the quantized state and then send out the code. When a neighbor receives the code, it uses a decoding algorithm to obtain an estimate of the sender's state. Hence, the multi-agent consensus problem needs to be

solved with quantized information and finite bits of data. The quantized consensus problem has been studied in the early works (Carli et al. 2007a; Kar and Moura 2009) where a uniform quantizer of infinite levels was used to design distributed control protocols. However, only consensus with bounded errors is achieved. Based on dynamic quantization, Carli et al. (2007b) achieved exact average consensus, but the number of quantization levels increases with the number of agents.

In Li et al. (2010), a distributed quantized consensus protocol was proposed for single integrator dynamics which can achieve exact average consensus with merely 1-bit information exchange between neighboring agents. This result relies on the specific design of both the encoder-decoder scheme and the distributed control protocol.

In the encoder-decoder scheme, dynamic quantization is applied to the state innovation to save communication cost. More specifically, the encoder Φ_j of the j -th agent is given by

$$\begin{cases} \xi_j(0) = 0, \\ \xi_j(t) = g(t-1)\Delta_j(t) + \xi_j(t-1), \\ \Delta_j(t) = Q\left[\frac{1}{g(t-1)}(x_j(t) - \xi_j(t-1))\right], t = 1, 2, \dots, \end{cases} \quad (8)$$

where $\xi_j(t)$ is the internal state of the encoder, $\Delta_j(t)$ is the output of the encoder to be sent to the neighbors of agent j , $g(t)$ is a positive scaling

function, and $Q(\cdot)$ is a finite-level uniform quantizer:

$$Q(y) = \begin{cases} 0, & -1/2 < y < 1/2; \\ i, & \frac{2i-1}{2} \leq y < \frac{2i+1}{2}, i = 1, 2, \dots, K-1; \\ K, & y \geq \frac{2K-1}{2}; \\ -Q(-y), & y \leq -1/2. \end{cases} \quad (9)$$

Although the above quantizer has $2K+1$ levels, there is no need to transmit the zero encoder output if the communication channel is assumed

to be perfect. Hence, the encoder output can be represented by at most $\lceil \log_2(2K) \rceil$ bits. Observe that the encoder Φ_j is based on the

innovation $x_j(t) - \xi_j(t-1)$ which generally requires fewer bits for its representation than the state $x_j(t)$.

For each communication channel $(j, i) \in \mathcal{E}$, agent i receives $\Delta_j(t)$ and uses the following decoder Ψ_{ij} to estimate x_j :

$$\begin{cases} y_{ij}(0) = 0, \\ y_{ij}(t) = g(t-1)\Delta_j(t) + y_{ij}(t-1), t = 1, 2, \dots, \end{cases} \quad (10)$$

where $y_{ij}(t)$ is an estimate of $x_j(t)$. Clearly, the decoded message $y_{ij}(t)$ is identical to the quantized state $\xi_j(t)$.

To achieve the exact consensus, the distributed control protocol is based on the self-compensation method to overcome information asymmetry of agents. For agent i , it is given by

$$u_i(t) = a \sum_{j \in \mathcal{N}_i} a_{ij}(y_{ij}(t) - \xi_i(t)). \quad (11)$$

Note that agent i does not use the exact state $x_i(t)$ to construct the control input. Instead, it uses the estimated state $\xi_i(t)$ obtained from its encoder output. This enables that both agents i and j use the same information for state estimation which is the key to achieving the exact consensus under the limited data rate.

Recall the error expression $w_{ij}(t) = y_{ij}(t) - x_j(t)$ in (6), and note that $y_{ij}(t) = \xi_j(t)$ for $j \in \mathcal{N}_i$. Thus, we can denote $w_j(t) = w_{ij}(t)$ for any $i \in \mathcal{N}_j$. Moreover, if we denote $\epsilon_j(t)$ as the quantization error, then we have $w_j(t) = g(t-1)\epsilon_j(t)$. Consequently, the dynamics of the closed loop system with the protocol (11) is given by

$$x(t+1) = (I - a\mathcal{L})x(t) - a\mathcal{L}w(t) \quad (12)$$

where $w = [w_1, \dots, w_N]' = g(t-1)[\epsilon_1, \dots, \epsilon_N]'$. Comparing with (5), it can be seen that (12) has an additional error term $-a\mathcal{L}w(t)$. Indeed, the scaling function $g(t)$ can be selected to be exponentially decaying, e.g., $g(t) = g_0\gamma^t$ with properly chosen $g_0 > 0$ and $\gamma \in (0, 1)$, and the quantization error can be guaranteed to be uniformly bounded. Therefore, the error term will be exponentially decaying, and under a connected and undirected network, it is shown that the

multi-agent system under the protocol (11) with encoder (8) and decoder (10) is able to achieve exact consensus with merely 1-bit of information exchange. Furthermore, the asymptotic convergence rate in relation to data rate and network synchronizability can be established – see Li et al. (2010) for more details.

With similar quantization techniques and distributed control protocol, the above work was extended to address more general topology (Zhang and Zhang 2013; Li et al. 2014). The cases of double and higher order integrator dynamics were studied in Li and Xie (2012) and Qiu et al. (2016) respectively, where only the first state variable is measurable, and the encoder-decoder scheme was designed to serve as a state observer. Specifically, it is shown that under a connected and undirected network and properly selected parameters, n bits of information exchange suffice to ensure the exponentially fast consensus for multi-agent systems with n -th order integrator dynamics. Qiu et al. (2017) considered the dynamics of $2m$ -th order oscillator, and it was found that the sufficient data rate for consensus is an integer between m and $2m$, depending on the position of the poles of the oscillator. The general LTI dynamics was also studied in You and Xie (2011a), and it was shown that the quantized consensus can be achieved with a limited data rate.

Average Consensus with Additive Noises

In analog communications, the additive communication noise channel model is often adopted, where the received information from agent j takes the form of (6). For simplicity, we assume that $w_{ij}(t)$ are zero mean white Gaussian noises

with variance σ_w^2 and $w_{ij}(t)$ are independent of each other for any i, j, t .

In this case, the consensus protocol is adapted to

$$u_i(t) = a(t) \sum_{j \in \mathcal{N}_i} a_{ij}(y_{ij}(t) - x_i(t)), \quad (13)$$

where a time-varying gain $a(t)$ is utilized and its necessity will be demonstrated soon.

The stacked dynamics can be computed as

$$x(t+1) = (I - a(t)\mathcal{L})x(t) + a(t)D_{\mathcal{G}}W(t), \quad (14)$$

where $D_{\mathcal{G}} = \text{diag}(\alpha_1, \dots, \alpha_N)$ with α_i being the i -th row of $\mathcal{A}$; $W(t) = [w_1^T(t), \dots, w_N^T(t)]^T$ with $w_i(t) = [w_{i1}(t), \dots, w_{iN}(t)]^T$.

It is clear that when there exist additive noises, exact consensus cannot be achieved. Therefore, we need a new definition of consensus, e.g., the commonly used mean square consensus definition.

Definition 2 The multi-agent system (1) achieves mean square consensus if

$$\lim_{t \rightarrow \infty} \mathbb{E}\{\| (I - \frac{1}{N}\mathbf{1}\mathbf{1}^T) x(t) \|^2\} = 0. \quad (15)$$

Moreover, if there exists a random variable x^* , with $\mathbb{E}\{x^*\} = \frac{1}{N} \sum_i x_i(0)$ and $\text{Var}(x^*) < \infty$, such that

$$\lim_{t \rightarrow \infty} \mathbb{E}\{\|x_i(t) - x^*\|^2\} = 0, \quad i = 1, \dots, N,$$

then the mean square average consensus is achieved.

Due to the additive noise in (14), the state $x(t)$ will fluctuate randomly. The fluctuation will not vanish as required by (15) if $a(t)$ does not converge to 0. Moreover, the convergence rate of $a(t)$ is critical for consensus. When $a(t)$ converges to 0 in (13), the signal $x_j(t)$ (contained in $y_{ij}(t)$) is attenuated together with the noise. Therefore, $a(t)$ cannot decrease too fast. Otherwise, the agents may converge to different individual limits.

It has been proved (Li and Zhang 2010) that the necessary and sufficient condition on $a(t)$ for the mean square consensus is

$$\sum_{t=0}^{\infty} a(t) = \infty, \quad \sum_{t=0}^{\infty} a(t)^2 < \infty. \quad (16)$$

Below we will briefly outline the proof. Since

$$\begin{aligned} \frac{1}{N} \sum_i x_i(t+1) &= \frac{1}{N} \sum_i x_i(t) \\ &\quad + \frac{1}{N} \mathbf{1}^T D_{\mathcal{G}} a(t) W(t), \end{aligned}$$

that is

$$\begin{aligned} \frac{1}{N} \sum_i x_i(t) &= \frac{1}{N} \sum_i x_i(0) \\ &\quad + \frac{1}{N} \mathbf{1}^T D_{\mathcal{G}} \sum_{i=0}^{t-1} a(i) W(i), \end{aligned}$$

we can expect that each agent's state would converge to

$$\begin{aligned} x^* &= \frac{1}{N} \sum_i x_i(0) \\ &\quad + \frac{1}{N} \mathbf{1}^T D_{\mathcal{G}} \sum_{t=0}^{\infty} a(t) W(t), \end{aligned}$$

which has bounded variance if and only if

$$\sum_{t=0}^{\infty} a(t)^2 < \infty. \quad (17)$$

Moreover, let $\delta(t) = (I - \frac{1}{N}\mathbf{1}\mathbf{1}^T)x(t)$ and define the Lyapunov function $V(t) = \delta(t)'\delta(t)$, with some manipulations; we can show that there exist t_0 and positive constants c_1, c_2 , and c_3 , such that for $t > t_0$,

$$\begin{aligned} (1 - c_1 a(t)) \mathbb{E}\{V(t)\} + c_2 a(t)^2 &\geq \mathbb{E}\{V(t+1)\} \\ &\geq e^{-c_3 \sum_{t=t_0}^{\infty} a(t)} \mathbb{E}\{V(t_0)\}. \end{aligned}$$

Therefore, a necessary and sufficient condition to ensure $\lim_{t \rightarrow \infty} \mathbb{E}\{V(t)\} = 0$ is

$$\sum_{t=0}^{\infty} a(t) = \infty, \quad \lim_{t \rightarrow \infty} a(t) = 0,$$

which, together with (17), yields the gain condition (16).

As is clear from the above proof, $\sum_{t=0}^{\infty} a(t) = \infty$ is the convergence condition, which is to make all agents' states reach an agreement with a proper rate. On the other hand, $\sum_{t=0}^{\infty} a(t)^2 < \infty$ is the robustness condition, which is to attenuate the measurement noises, such that the consensus protocol is robust with respect to measurement noises.

The gain condition (16) has been shown to be sufficient in various settings: e.g., double integrator dynamics (Cheng et al. 2011), general LTI agent dynamics (Cheng et al. 2014), time-varying topology (Huang et al. 2010; Huang and Manton 2010; Li and Zhang 2010), and output feedback case (Wang et al. 2015). The convergence rate analysis has also been studied (Xu et al. 2012; Tang and Li 2015; Cheng et al. 2016).

Packet Dropouts

In the presence of packet dropouts, the received information from agent j takes the form of (7). Accordingly, the consensus protocol is adapted to

$$u_i(t) = a \sum_{j \in \mathcal{N}_i} a_{ij} \gamma_{ij}(t) (x_j(t) - x_i(t)). \quad (18)$$

Clearly, it implies that only when $\gamma_{ij}(t) = 1$, then the relative state between agent j and agent i will be used. Consequently, the stacked dynamics is given by

$$x(t+1) = (I - a\mathcal{L}(t))x(t), \quad (19)$$

where $\mathcal{L}(t) = [\mathcal{L}_{ij}(t)]_{N \times N}$ is defined as $\mathcal{L}_{ii}(t) = \sum_{j \in \mathcal{N}_i} a_{ij} \gamma_{ij}(t)$, $\mathcal{L}_{ij} = -a_{ij} \gamma_{ij}(t)$ for $i \neq j$. It was shown in (Tahbaz-Salehi and Jadbabaie 2008) that almost sure consensus can be achieved if the spectrum of the expected graph is smaller than 1, i.e., $|\lambda_2(E(I - \mathcal{L}(t)))| < 1$.

More research works have been conducted by modeling $\gamma_{ij}(t)$ as a Bernoulli process with $P(\gamma_{ij}(t) = 1) = p_{ij}$ which is the recovery rate on the communication channel (i, j) . It is found that the recovery rate plays an important role in ensuring consensus. In Zhang and Tian (2012) and You et al. (2013), it was shown that the recovery rate needs to be sufficiently large to ensure the mean square consensus.

Furthermore, it is also worth mentioning that when $\gamma_{ij}(t)$ is a general random variable, then the problem is closely related with the fading channel issue. See the recent works (Xu et al. 2016, 2018) for more details.

Summary and Future Research

This article described the multi-agent consensus under noisy environment with special focus on consensus where communication channels are of limited data rate or subject to additive/multiplicative noises. While the consensus problem has been extensively studied, there are several issues that remain to be further studied.

For the quantized consensus problem, most of the existing studies have been limited to integrator dynamics with a few exceptions; see e.g., You and Xie (2011b), where a lower bound on the required data rate for consensus is provided. The bound, however, is overly conservative for general dynamics. It is of interest to study the minimal data rate problem for consensus of general linear dynamics. On the other hand, Li et al. (2010) established a relationship between the asymptotic ($N \rightarrow \infty$) consensus convergence rate and network synchronizability (λ_2/λ_N). Similar convergence rate result is not available for higher order integrator dynamics. Moreover, (You and Xie 2011b) has provided a necessary and sufficient condition for consensusability in terms of the agent dynamics and network topology under the assumption that the communication channels have infinite capacities. It is interesting to further take data rate into consideration from both theoretical and practical points of view though the problem is challenging.

Cross-References

- [Estimation and Control over Networks](#)
- [Information-Based Multi-Agent Systems](#)

Bibliography

- Carli R, Fagnani F, Frasca P, Taylor T, Zampieri S (2007a) Average consensus on networks with transmission noise or quantization. In: 2007 European control conference (ECC). IEEE, pp 1852–1857
- Carli R, Fagnani F, Frasca P, Zampieri S (2007b) Efficient quantized techniques for consensus algorithms. In: NeCST workshop, Nancy, France Nancy
- Cheng L, Hou ZG, Tan M, Wang X (2011) Necessary and sufficient conditions for consensus of double-integrator multi-agent systems with measurement noises. *IEEE Trans Autom Control* 56(8):1958–1963
- Cheng L, Hou ZG, Tan M (2014) A mean square consensus protocol for linear multi-agent systems with communication noises and fixed topologies. *IEEE Trans Autom Control* 59(1):261–267
- Cheng L, Wang Y, Ren W, Hou ZG, Tan M (2016) On convergence rate of leader-following consensus of linear multi-agent systems with communication noises. *IEEE Trans Autom Control* 61(11):3586–3592
- Huang M, Manton JH (2010) Stochastic consensus seeking with noisy and directed inter-agent communication: fixed and randomly varying topologies. *IEEE Trans Autom Control* 55(1):235–241
- Huang M, Dey S, Nair GN, Manton JH (2010) Stochastic consensus over noisy networks with markovian and arbitrary switches. *Automatica* 46(10):1571–1583
- Kar S, Moura JM (2009) Distributed consensus algorithms in sensor networks: quantized data and random link failures. *IEEE Trans Signal Process* 58(3):1383–1400
- Li D, Liu Q, Wang X, Yin Z (2014) Quantized consensus over directed networks with switching topologies. *Syst Control Lett* 65:13–22
- Li T, Xie L (2012) Distributed coordination of multi-agent systems with quantized-observer based encoding-decoding. *IEEE Trans Autom Control* 57(12):3023–3037
- Li T, Zhang JF (2010) Consensus conditions of multi-agent systems with time-varying topologies and stochastic communication noises. *IEEE Trans Autom Control* 55(9):2043–2057
- Li T, Fu M, Xie L, Zhang JF (2010) Distributed consensus with limited communication data rate. *IEEE Trans Autom Control* 56(2):279–292
- Qiu Z, Xie L, Hong Y (2016) Quantized leaderless and leader-following consensus of high-order multi-agent systems with limited data rate. *IEEE Trans Autom Control* 61(9):2432–2447
- Qiu Z, Xie L, Hong Y (2017) Data rate for distributed consensus of multi-agent systems with high-order oscillator dynamics. *IEEE Trans Autom Control* 11(62):6065–6072
- Tahbaz-Salehi A, Jadbabaie A (2008) A necessary and sufficient condition for consensus over random networks. *IEEE Trans Autom Control* 53(3):791–795
- Tang H, Li T (2015) Continuous-time stochastic consensus: stochastic approximation and kalman–bucy filtering based protocols. *Automatica* 61:146–155
- Wang Y, Cheng L, Ren W, Hou ZG, Tan M (2015) Seeking consensus in networks of linear agents: communication noises and markovian switching topologies. *IEEE Trans Autom Control* 60(5):1374–1379
- Xu J, Zhang H, Xie L (2012) Stochastic approximation approach for consensus and convergence rate analysis of multiagent systems. *IEEE Trans Autom Control* 57(12):3163–3168
- Xu L, Xiao N, Xie L (2016) Consensusability of discrete-time linear multi-agent systems over analog fading networks. *Automatica* 71:292–299
- Xu L, Zheng J, Xiao N, Xie L (2018) Mean square consensus of multi-agent systems over fading networks with directed graphs. *Automatica* 95:503–510
- You K, Xie L (2011a) Network topology and communication data rate for consensusability of discrete-time multi-agent systems. *IEEE Trans Autom Control* 56(10):2262–2275
- You K, Xie L (2011b) Network topology and communication data rate for consensusability of discrete-time multi-agent systems. *IEEE Trans Autom Control* 56(10):2262–2275
- You K, Li Z, Xie L (2013) Consensus condition for linear multi-agent systems over randomly switching topologies. *Automatica* 49(10):3125–3132
- Zhang Q, Zhang JF (2013) Quantized data-based distributed consensus under directed time-varying communication topology. *SIAM J Control Optim* 51(1):332–352
- Zhang Y, Tian YP (2012) Maximum allowable loss probability for consensus of multi-agent systems over random weighted lossy networks. *IEEE Trans Autom Control* 57(8):2127–2132

Nominal Model-Predictive Control

Lars Grüne

Mathematical Institute, University of Bayreuth, Bayreuth, Germany

Abstract

Model-predictive control is a controller design method which synthesizes a sampled data feedback controller from the iterative solution of open-loop optimal control problems. We describe the basic functionality of

MPC controllers, their properties regarding feasibility, stability and performance, and the assumptions needed in order to rigorously ensure these properties in a nominal setting.

Keywords

Model predictive control · Receding horizon control · Piecewise optimal control

Introduction

Model-predictive control (MPC) is a method for the optimization-based control of linear and nonlinear dynamical systems. While the literal meaning of “model-predictive control” applies to virtually every model-based controller design method, nowadays the term commonly refers to control methods in which pieces of open-loop optimal control functions or sequences are put together in order to synthesize a sampled data feedback law. As such, it is often used synonymously with “receding horizon control.”

The concept of MPC was first presented in Propöř (1963) and was reinvented several times already in the 1960s. Due to the lack of sufficiently fast computer hardware, for a while these ideas did not have much of an impact. This changed during the 1970s when MPC was successfully used in chemical process control. At that time, MPC was mainly applied to linear systems with quadratic cost and linear constraints, since for this class of problems algorithms were sufficiently fast for real-time implementation – at least for the typically relatively slow dynamics of process control systems. The 1980s have then seen the development of theory and increasingly sophisticated concepts for linear MPC, while in the 1990s nonlinear MPC (often abbreviated as NMPC) attracted the attention of the MPC community. After the year 2000, several gaps in the analysis of nonlinear MPC without terminal constraints and costs were closed, and increasingly faster algorithms were developed. Together with the progress in hardware, this has considerably broadened the possible applications of both linear and nonlinear MPC.

In this entry, we explain the functionality of nominal MPC along with its most important properties and the assumptions needed to rigorously ensure these properties. We also give some hints on the underlying proofs. The term nominal MPC refers to the assumption that the mismatch between our model and the real plant is sufficiently small to be neglected in the following considerations. If this is not the case, methods from robust MPC must be used (► [Robust Model Predictive Control](#)). We describe all concepts for nonlinear discrete time systems, noting that the basic results outlined in this entry are conceptually similar for linear and for continuous-time systems.

Model-Predictive Control

In this entry, we discuss MPC for discrete time control systems of the form

$$x_{\mathbf{u}}(j+1) = f(x_{\mathbf{u}}(j), u(j)), \quad x_{\mathbf{u}}(0) = x_0 \quad (1)$$

with state $x_{\mathbf{u}}(j) \in X$, initial condition $x_0 \in \mathbb{X}$, and control input sequence $\mathbf{u} = (u(0), u(1), \dots)$ with $u(k) \in U$, where the state space X and the control value space U are normed spaces. For control systems in continuous time, one may either apply the discrete time approach to a sampled data model of the system. Alternatively, continuous-time versions of the concepts and results from this entry are available in the literature; see, e.g., Findeisen and Allgöwer (2002) or Mayne et al. (2000).

The core of any MPC scheme is an optimal control problem of the form

$$\text{minimize } J_N(x_0, \mathbf{u}) \quad (2)$$

w.r.t. $\mathbf{u} = (u(0), \dots, u(N-1))$ with

$$J_N(x_0, \mathbf{u}) : \sum_{j=0}^{N-1} \ell(x_{\mathbf{u}}(j), u(j)) + F(x_{\mathbf{u}}(N)) \quad (3)$$

subject to the constraints

$$\begin{aligned} u(j) \in \mathbb{U}, x_{\mathbf{u}}(j) \in \mathbb{X} \text{ for } j = 0, \dots, N-1 \\ x_{\mathbf{u}}(N) \in \mathbb{X}_0, \end{aligned} \quad (4)$$

for control constraint set $\mathbb{U} \subseteq U$, state constraint set $\mathbb{X} \subseteq X$, and terminal constraint set $\mathbb{X}_0 \subseteq X$. The function $\ell : \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{R}$ is called stage cost or running cost; the function $F : \mathbb{X} \rightarrow \mathbb{R}$ is referred to as terminal cost. We assume that for each initial value $x_0 \in \mathbb{X}$, the optimal control problem (2) has a solution and denote the corresponding minimizing control sequence by $\mathbf{u}^*$. Algorithms for computing $\mathbf{u}^*$ are discussed in ► [Optimization Algorithms for MPC](#) and ► [Explicit Model Predictive Control](#).

The key idea of MPC is to compute the values $\mu_N(x)$ of the MPC feedback law μ_N from the open-loop optimal control sequences $\mathbf{u}^*$. To formalize this idea, consider the closed-loop system

$$x_{\mu_N}(k+1) = f(x_{\mu_N}(k), \mu_N(x_{\mu_N}(k))). \quad (5)$$

In order to evaluate μ_N along the closed-loop solution, given an initial value $x_{\mu_N}(0) \in \mathbb{X}$, we iteratively perform the following steps.

Basic MPC Loop

1. Set $k := 0$.
2. Solve (2)–(4) for $x_0 = x_{\mu_N}(k)$; denote the optimal control sequence by $\mathbf{u}^* = (u^*(0), \dots, u^*(N-1))$.
3. Set $\mu_N(x_{\mu_N}(k)) : u^*(0)$, compute $x_{\mu_N}(k+1)$ according to (5), set $k := k+1$ and go to (1).

Due to its ability to handle constraints and possibly nonlinear dynamics, MPC has become one of the most popular modern control methods in the industry (► [Model Predictive Control in Practice](#)). While in the literature various variants of this basic scheme are discussed, here we restrict ourselves to this most widely used basic MPC scheme.

When analyzing an MPC scheme, three properties are important:

- **Recursive Feasibility**, i.e., the property that the constraints (4) can be satisfied in Step (ii) in each sampling instant
- **Stability**, i.e., in particular convergence of the closed-loop solutions $x_{\mu_N}(k)$ to a desired equilibrium x_* as $k \rightarrow \infty$
- **Performance**, i.e., appropriate quantitative properties of $x_{\mu_N}(k)$

Here we discuss these three issues for two widely used MPC variants:

1. MPC with terminal constraints and costs
2. MPC with neither terminal constraints nor costs

In (a), F and $\mathbb{X}_0$ in (3) and (4) are specifically designed in order to guarantee proper performance of the closed loop. In (b), we set $F \equiv 0$ and $\mathbb{X}_0 = \mathbb{X}$. Thus, the choice of ℓ and N in (3) is the most important part of the design procedure.

Recursive Feasibility

Since the ability to handle constraints is one of the key features of MPC, it is important to ensure that the constraints $x_{\mu_N}(k) \in \mathbb{X}$ and $\mu_N(x_{\mu_N}(k)) \in \mathbb{U}$ are satisfied for all $k \geq 0$. However, beyond constraint satisfaction, the stronger property $x_{\mu_N}(k) \in \mathbb{X}_N$ is required, where $\mathbb{X}_N$ denotes the *feasible set* for horizon N ,

$$\mathbb{X}_N := \{x \in \mathbb{X} \mid \text{there exists } \mathbf{u} \text{ such that (4) holds}\}.$$

The property $x \in \mathbb{X}_N$ is called *feasibility* of x . Feasibility of $x = x_{\mu_N}(k)$ is a prerequisite for the MPC feedback μ_N being well defined, because the nonexistence of such an admissible control sequence $\mathbf{u}$ would imply that solving (2) under the constraints (4) in Step (ii) of the MPC iteration is impossible.

Since for $k \geq 0$ the state $x_{\mu_N}(k+1) = f(x_{\mu_N}(k), u^*(0))$ is determined by the solution of the previous optimal control problem, the usual way to address this problem is via the notion of

recursive feasibility. This property demands the existence of a set $A \subseteq \mathbb{X}$ such that:

- For each $x_0 \in A$, the problem (2)–(4) is feasible.
- For each $x_0 \in A$ and the optimal control u^* from (2) to (4), the relation $f(x_0, u^*(0)) \in A$ holds.

It is not too difficult to see that this property implies $x_{\mu_N}(k) \in A$ for all $k \geq 1$ if $x_{\mu_N}(0) \in A$.

For terminal-constrained problems, recursive feasibility is usually established by demanding that the terminal constraint set $\mathbb{X}_0$ is *viable* or *controlled forward invariant*. This means that for each $x \in \mathbb{X}_0$, there exists $u \in \mathbb{U}$ with $f(x, u) \in \mathbb{X}_0$. Under this assumption, it is quite straightforward to prove that the feasible set $A = \mathbb{X}_N$ is also recursively feasible (Grüne and Pannek 2011, Lemma 5.11). Note that viability of $\mathbb{X}_0$ is immediate if $\mathbb{X}_0 = \{x_*\}$ and $x_* \in \mathbb{X}$ is an equilibrium, i.e., a point for which there exists $u_* \in \mathbb{U}$ with $f(x_*, u_*) = x_*$. This setting is referred to as *equilibrium terminal constraint*.

For MPC without terminal constraints, the most straightforward way to ensure recursive feasibility is to assume that the state constraint set $\mathbb{X}$ is viable (Grüne and Pannek 2011, Theorem 3.5). However, checking viability and even more constructing a viable state constraint set is in general a very difficult task. Hence, other methods for establishing recursive feasibility are needed. One method is to assume that the sequence of feasible sets $\mathbb{X}_N$, $N \in \mathbb{N}$ becomes *stationary* for some N_0 , i.e., that $\mathbb{X}_{N+1} = \mathbb{X}_N$ holds for all $N \geq N_0$. Under this assumption, recursive feasibility of $\mathbb{X}_{N_0}$ follows, see Kerrigan (2000, Theorem 5.3). However, like viability, stationarity is difficult to verify.

For this reason, a conceptually different approach to ensure recursive feasibility was presented in Grüne and Pannek (2011, Theorem 7.20); a similar approach for linear systems can be found in Primbs and Nevistić (2000). The approach is suitable for stabilizing MPC problems in which the stage cost ℓ penalizes the distance to a desired equilibrium x_* (cf. section “Stability”). Assuming the existence – but not

the knowledge – of a viable neighborhood $\mathcal{N}$ of x_* , one can show that any initial point x_0 for which the corresponding open-loop optimal solution satisfies $x_{u^*}(j) \in \mathcal{N}$ or some $j \leq N$ is contained in a recursively feasible set. The fact that ℓ penalizes the distance to x_* then implies $x_{u^*}(j) \in \mathcal{N}$ for suitable initial values. Together, these properties yield the existence of recursively feasible sets A_N which converge to set A_∞ as N increases.

Stability

Stability in the sense of this entry refers to the fact that a prespecified equilibrium $x_* \in \mathbb{X}$ – typically a desired operating point – is asymptotically stable for the MPC closed loop for all initial values in some set $\mathcal{S}$. This means that the solutions $x_{\mu_N}(k)$ starting in $\mathcal{S}$ converge to x_* as $k \rightarrow \infty$ and that solutions starting close to x_* remain close to x_* for all $k \geq 0$. Note that this setting can be extended to time-varying reference solutions; see ► [Tracking Model Predictive Control](#).

In order to enforce this property, we assume that the stage cost ℓ penalizes the distance to the equilibrium x_* in the following sense: ℓ satisfies

$$\ell(x_*, u_*) = 0 \text{ and } \alpha_1(|x|) \leq \ell(x, u) \quad (6)$$

for all $x \in \mathbb{X}$ and $u \in \mathbb{U}$. Here α_1 is a $\mathcal{K}_\infty$ function, i.e., a continuous function $\alpha_1 : [0, \infty) \rightarrow [0, \infty)$ which is strictly increasing, unbounded, and satisfies $\alpha_1(0) = 0$. With $|x|$, we denote the norm on X . In this entry, we exclusively discuss stage costs ℓ satisfying (6). More general settings using appropriate detectability conditions are discussed in Rawlings and Mayne (2009, Sect. 2.7) or Grimm et al. (2005) in the context of stabilizing MPC. Even more general ℓ are allowed in the context of economic MPC; see the ► [Economic Model Predictive Control](#) article.

In case of terminal constraints and terminal costs, a compatibility condition between ℓ and F is needed on $\mathbb{X}_0$ in order to ensure stability. More precisely, we demand that for each $x \in \mathbb{X}_0$ there exists a control value $u \in \mathbb{U}$ such that $f(x, u) \in \mathbb{X}_0$ and

$$F(f(x, u)) - F(x) \leq -\ell(x, u) \quad (7)$$

holds. Observe that the condition $f(x, u) \in \mathbb{X}_0$ is again the viability condition which we already imposed for ensuring recursive feasibility. Note that (7) is trivially satisfied for $F \equiv 0$ in case of $\mathbb{X}_0 = \{x_*\}$ by choosing $u = u_*$.

Stability is now concluded by using the optimal value function

$$V_N(x_0) := \inf_{\mathbf{u} \text{ s.t. (4)}} J_N(x_0, \mathbf{u})$$

as a Lyapunov function. This will yield stability on $\mathcal{S} = \mathbb{X}_N$, as $\mathbb{X}_N$ is exactly the set on which V_N is defined. In order to prove that V_N is a Lyapunov function, we need to check that V_N is bounded from below and above by $\mathcal{K}_\infty$ functions α_1 and α_2 and that V_N is strictly decaying along the closed-loop solution.

The first amounts to checking

$$\alpha_1(|x|) \leq V_N(x) \leq \alpha_2(|x|) \quad (8)$$

for all $x \in \mathbb{X}_N$. The lower bound follows immediately from (6) (with the same α_1), and the upper bound can be ensured by conditions on the problem data (see, e.g., Rawlings and Mayne 2009, Sect. 4.5; Grüne and Pannek 2011, Sect. 5.3).

For ensuring that V_N is strictly decreasing along the closed-loop solutions, we need to prove

$$V_N(f(x, \mu_N(x))) \leq V_N(x) - \ell(x, \mu_N(x)). \quad (9)$$

In order to prove this inequality, one uses on the one hand the dynamic programming principle stating that

$$V_{N-1}(f(x, \mu_N(x))) = V_N(x_0) - \ell(x, \mu_N(x)). \quad (10)$$

On the other hand, one shows that (7) implies

$$V_{N-1}(x) \geq V_N(x) \quad (11)$$

for all $x \in \mathbb{X}_N$. Inserting (11) with $f(x, \mu_N(x))$ in place of x into (10) then immediately yields (9). Details of this proof can be found, e.g., in Mayne et al. (2000), Rawlings and Mayne

(2009), or Grüne and Pannek (2011). The survey Mayne et al. (2000) is probably the first paper which develops the conditions needed for this proof in a systematic way; a continuous-time version of these results can be found in Fontes (2001).

Summarizing, for MPC with terminal constraints and costs, under the conditions (6)–(8), we obtain asymptotic stability of x_* on $\mathcal{S} = \mathbb{X}_N$.

For MPC without terminal constraints and costs, i.e., with $\mathbb{X}_0 = \mathbb{X}$ and $F \equiv 0$, these conditions can never be satisfied, as (7) will immediately imply $\ell(x, u) = 0$ for all $x \in \mathbb{X}$, contradicting (6). Moreover, without terminal constraints and costs, one cannot expect (9) to be true. This is because without terminal constraints, the inequality $V_{N-1}(x) \leq V_N(x)$ holds, which together with the dynamic programming principle implies that if (9) holds, then it holds with equality. This, however, would imply that μ_N is the infinite horizon optimal feedback law, which – though not impossible – is very unlikely to hold.

Thus, we need to relax (9). In order to do so, instead of (9), we assume the relaxed inequality

$$V_N(f(x, \mu_N(x))) \leq V_N(x) - \alpha \ell(x, \mu_N(x)) \quad (12)$$

for some $\alpha > 0$ and all $x \in \mathbb{X}$, which is still enough to conclude asymptotic stability of x_* if (6) and (8) hold. The existence of such an α can be concluded from bounds on the optimal value function V_N . Assuming the existence of constants $\gamma_K \geq 0$ such that the inequality

$$V_K(x) \leq \gamma_K \min_{u \in \mathbb{U}} \ell(x, u) \quad (13)$$

holds for all $K = 1, \dots, N$ and $x \in \mathbb{X}$, there are various ways to compute α from $\gamma_1, \dots, \gamma_N$, see Grüne (2012, Sect. 3). The best possible estimate for α , whose derivation is explained in detail in Grüne and Pannek (2011, Chap. 6), yields

$$\alpha = 1 - \frac{(\gamma_N - 1) \prod_{i=2}^N (\gamma_i - 1)}{\prod_{i=2}^N \gamma_i - \prod_{i=2}^N (\gamma_i - 1)}. \quad (14)$$

Though not immediately obvious, a closer look at this term reveals $\alpha \rightarrow 1$ as $N \rightarrow \infty$ if the γ_K are bounded. Hence, $\alpha > 0$ for sufficiently large N .

Summarizing the second part of this section, for MPC without terminal constraints and costs, under the conditions (6), (8), and (13), asymptotic stability follows on $\mathcal{S} = \mathbb{X}$ for all optimization horizons N for which $\alpha > 0$ holds in (14). Note that the condition (13) implicitly depends on the choice of ℓ . A judicious choice of ℓ can considerably reduce the size of the horizon N for which $\alpha > 0$ holds, see Grüne and Pannek (2011, Sect. 6.6) and thus the computational effort for solving (2)–(4).

Performance

Performance of MPC controllers can be measured in many different ways. As the MPC controller is derived from successive solutions of (2), a natural quantitative way to measure its performance is to evaluate the infinite horizon functional corresponding to (3) along the closed loop, i.e.,

$$J_{\infty}^{\ell}(x_0, \mu_N) := \sum_{k=0}^{\infty} \ell(x_{\mu_N}(k), \mu_N(x_{\mu_N}(k)))$$

with $x_{\mu_N}(0) = x_0$. This value can then be compared with the optimal infinite horizon value

$$V_{\infty}(x_0) := \inf_{\mathbf{u}: u(k) \in \mathbb{U}, x_{\mathbf{u}}(k) \in \mathbb{X}} J_{\infty}(x_0, \mathbf{u})$$

where

$$J_{\infty}(x_0, \mathbf{u}) := \sum_{k=0}^{\infty} \ell(x_{\mathbf{u}}(k), u(k)).$$

To this end, for MPC with terminal constraints and costs, by induction over (9) and using non-negativity of ℓ , it is fairly easy to conclude the inequality

$$J_{\infty}^{\ell}(x_0, \mu_N) \leq V_N(x_0)$$

for all $x \in \mathbb{X}_N$. However, due to the conditions on the terminal cost in (7), V_N may be considerably larger than V_{∞} and an estimate relating these two functions is in general not easy to derive (Grüne and Pannek 2011, Examples 5.18 and 5.19). However, it is possible to show that under the same assumptions guaranteeing stability, the convergence

$$V_N(x) \rightarrow V_{\infty}(x)$$

holds for $N \rightarrow \infty$ (Grüne and Pannek 2011, Theorem 5.21). Hence, we recover approximately optimal infinite horizon performance for sufficiently large horizon N .

For MPC without terminal constraints and costs, the inequality $V_N(x_0) \leq V_{\infty}(x_0)$ is immediate; however, (9) will typically not hold. As a remedy, we can use (12) in order to derive an estimate. Using induction over (12), we arrive at the estimate

$$J_{\infty}^{\ell}(x_0, \mu_N) \leq V_N(x_0)/\alpha \leq V_{\infty}(x_0)/\alpha.$$

Since $\alpha \rightarrow 1$ as $N \rightarrow \infty$, also in this case we obtain approximately optimal infinite horizon performance for sufficiently large horizon N .

Summary and Future Directions

MPC is a controller design method which uses the iterative solution of open-loop optimal control problems in order to synthesize a sampled data feedback controller μ_N . The advantages of MPC are its ability to handle constraints, the rigorously provable stability properties of the closed loop, and its approximate optimality properties. Assumptions needed in order to rigorously ensure these properties together with the corresponding mathematical arguments have been outlined in this entry, both for MPC with terminal constraints and costs and without. Among the disadvantages of MPC are the computational effort and the fact that the resulting feedback is a full state feedback, thus necessitating the use of a state estimator to reconstruct the state from output data (► [Moving Horizon Estimation](#)).

Future directions include the application of MPC to more general problems than set point stabilization or tracking, the development of efficient algorithms for large-scale problems including those originating from discretized infinite-dimensional control problems, and the understanding of the opportunities and limitations of MPC in increasingly complex environments; see also ► [Distributed Model Predictive Control](#).

Cross-References

- [Distributed Model Predictive Control](#)
- [Economic Model Predictive Control](#)
- [Robust Model Predictive Control](#)
- [Stochastic Model Predictive Control](#)
- [Tracking Model Predictive Control](#)

Recommended Reading

MPC in the form known today was first described in Propöř (1963) and is now covered in several monographs, two recent ones being Rawlings and Mayne (2009) and Grüne and Pannek (2011). More information on continuous-time MPC can be found in the survey by Findeisen and Allgöwer (2002). The nowadays standard framework for stability and feasibility of MPC with stabilizing terminal constraints is presented in Mayne et al. (2000); for a continuous-time version, see Fontes (2001). Stability of MPC without terminal constraints was proved in Grimm et al. (2005) under very general conditions; for a comparison of various such results, see Grüne (2012). Feasibility without terminal constraints is discussed in Kerrigan (2000) and Primbs and Nevistić (2000).

Bibliography

- Findeisen R, Allgöwer F (2002) An introduction to nonlinear model predictive control. In: 21st Benelux meeting on systems and control, Veldhoven, The Netherlands (see also <http://www.tue.nl/en/publication/ep/p/d/ep-uid/252788/>), pp 119–141
- Fontes FACC (2001) A general framework to design stabilizing nonlinear model predictive controllers. *Syst Control Lett* 42:127–143

- Grimm G, Messina MJ, Tuna SE, Teel AR (2005) Model predictive control: for want of a local control Lyapunov function, all is not lost. *IEEE Trans Autom Control* 50(5):546–558
- Grüne L (2012) NMPC without terminal constraints. In: Proceedings of the IFAC conference on nonlinear model predictive control – NMPC’12, pp 1–13
- Grüne L, Pannek J (2011) *Nonlinear model predictive control: theory and algorithms*. Springer, London
- Kerrigan EC (2000) *Robust constraint satisfaction: invariant sets and predictive control*. PhD thesis, University of Cambridge
- Mayne DQ, Rawlings JB, Rao CV, Scokaert POM (2000) Constrained model predictive control: stability and optimality. *Automatica* 36:789–814
- Primbs JA, Nevistić V (2000) Feasibility and stability of constrained finite receding horizon control. *Automatica* 36(7):965–971
- Propöř A (1963) Application of linear programming methods for the synthesis of automatic sampled-data systems. *Avtomat i Telemekh* 24:912–920
- Rawlings JB, Mayne DQ (2009) *Model predictive control: theory and design*. Nob Hill Publishing, Madison

Nonlinear Adaptive Control

Alessandro Astolfi

Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma Tor Vergata, Rome, Italy

Abstract

We consider the control of nonlinear systems in which parameters are uncertain and may vary. For such systems the control must adapt to the parameter change to deliver closed-loop performance, such as asymptotic stability or tracking. A concise description of available methods and basic adaptive stabilization results, which can be used as building blocks for complex adaptive control problems, are discussed.

Keywords

Adaptive control · Nonlinear systems · Adaptive stabilization · Estimation ·

Stability · Certainty equivalence · Output feedback

Introduction

The adaptive control problem, namely, the problem of designing a feedback controller which contains an *adaptation mechanism* to counteract changes in the parameters of the system to be controlled, is of significant importance in applications. In almost all systems, physical parameters are subject to changes. These may be triggered, for example, by changes in temperature (the volume of a liquid/gas), aging (the friction coefficient of a mechanical system), or normal operation (the mass of the fuel of an aircraft changes during flight; the center of mass of a vehicle is affected by its load).

While adaptive control is naturally associated with the notion of estimation – the parameters of a system have to be identified to design a controller – it may be possible to design adaptive controllers which do not rely on an *explicit* parameter estimation – it is sufficient to estimate the *role* of the parameters in the control signal.

Adaptive control is different from robust control. In the simplest possible scenario, the aim of robust control is to design a control law guaranteeing performance specifications for a given range of parameter values. Robust control thus requires some a priori information on the parameter. Adaptive control does not require any a priori information on the parameter, although any such information can be exploited in the controller design, but requires a parameterized model: a model which contains information on the way the parameters affect the dynamics of the system.

The adaptive control problem for general nonlinear systems can be formulated as follows. Consider a nonlinear system described by equations of the form (while we focus on continuous-time systems, similar considerations apply to discrete-time systems):

$$\dot{x} = F(x, u, \theta), \quad y = H(x, \theta), \quad (1)$$

where $x(t) \in \mathbb{R}^n$ denotes the state of the system, $u(t) \in \mathbb{R}^m$ denotes the input of the system,

$\theta \in \mathbb{R}^q$ denotes the constant unknown parameter, $y(t) \in \mathbb{R}^p$ denotes the measured output, and $F : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q \rightarrow \mathbb{R}^n$ and $H : \mathbb{R}^n \times \mathbb{R}^q \rightarrow \mathbb{R}^p$ are smooth mappings. In what follows, for simplicity, we mostly assume that $y = x$, that is, the whole state of the system is available for control design.

The adaptive control problem consists in finding, if possible, a dynamic control law described by equations of the form

$$\dot{\hat{\theta}} = w(x, \hat{\theta}, r), \quad (2)$$

$$u = v(x, \hat{\theta}, r), \quad (3)$$

with $r(t) \in \mathbb{R}^s$ an exogenous (reference) signal, and $w : \mathbb{R}^n \times \mathbb{R}^q \times \mathbb{R}^s \rightarrow \mathbb{R}^q$ and $v : \mathbb{R}^n \times \mathbb{R}^q \times \mathbb{R}^s \rightarrow \mathbb{R}^m$ smooth mappings, such that the closed-loop system, described by the equations

$$\dot{x} = F(x, v(x, \hat{\theta}, r), \theta), \quad \dot{\hat{\theta}} = w(x, \hat{\theta}, r), \quad (4)$$

has specific properties. For example one could require that all trajectories be bounded and the x -component of the state converge to a given value x^* (this is the so-called adaptive regulation requirement) or that the input-output behavior of the system from the input r to some user-defined output signal coincide with a given reference model (this is the so-called model reference adaptive control requirement).

A natural way to characterize design specifications for the adaptive control problem and to facilitate its solution is to assume the existence of a known parameter controller, described by the equation

$$u = v^*(x, \theta, r), \quad (5)$$

such that the nonadaptive closed-loop system $\dot{x} = F(x, v^*(x, \theta, r), \theta)$ satisfies given design specifications. In this perspective, the adaptive control problem boils down to the design of the *update law* (2) and of the *feedback law* (3) such that the behavior of the adaptive closed-loop system *matches* that of the nonadaptive closed-loop system $\dot{x} = F(x, v^*(x, \theta, r), \theta)$.

The above description suggests a design method for the *feedback law*: one could replace θ with $\hat{\theta}$ in Eq. (5). This design is often known

as certainty equivalence design and lends itself to the interpretation that $\hat{\theta}$ be an estimate for θ . Naturally, one could also modify the feedback law, adding further $\hat{\theta}$ -dependent and x -dependent terms: this is often called a redesign. Redesign may be guided by various considerations, for example, it may be based on the use of a specific Lyapunov function (yielding the so-called Lyapunov redesign), or by structural properties of the system, or by robustness constraints.

The interpretation of $\hat{\theta}$ as an estimate for θ leads to two somewhat diverse approaches for the design of the update law. The former, pursued in the so-called estimation-based (or indirect) adaptive control, relies on the design of a parameter estimator, for example, using recursive least-square methods. This approach has its roots in identification theory and has been studied in-depth for linear systems. The latter relies on the observation that the design of an update law is equivalent to the design of a (reduced-order) observer for the extended system

$$\dot{x} = F(x, u, \theta), \quad \dot{\theta} = 0,$$

with output $y = x$. This approach has its roots in the theory of nonlinear observer design.

The approaches described so far rely on a sort of separation principle: the update law and the feedback law are designed separately. While this approach may be adequate for linear systems, for nonlinear systems it is often necessary to design the update law and the feedback law in one step, i.e., the selection of the feedback law depends upon the selection of the update law and vice versa. To illustrate this design method and provide some explicit adaptive control design tools, we focus on a special class of nonlinear systems: systems which are linearly parameterized in the unknown parameter.

Linearly Parameterized Systems

Consider the system (1) and assume that the mapping F is affine in the parameter θ and in the control u , namely:

$$F(x, u, \theta) = f_0(x) + g(x)u + f_1(x)\theta, \quad (6)$$

with $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^m$, and $f_1 : \mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^q$ smooth mappings. For this class of systems, under additional assumptions, it is possible to provide systematic adaptive control design tools. We provide two formal results: additional results (depending on the specific assumptions imposed on the system) may be derived. In both cases the focus is on the adaptive stabilization problem: the goal of the adaptive controller is to render a given equilibrium stable, in the sense of Lyapunov, and to guarantee convergence of the x -component of the state (recall that the state of the adaptive closed-loop system is the vector $(x, \hat{\theta})$).

Theorem 1 Consider the system (6) and a point x^* . Assume there exist a known parameter controller

$$u = v_0(x) + v_1(x)\theta,$$

with $v_0 : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $v_1 : \mathbb{R}^n \rightarrow \mathbb{R}^m \times \mathbb{R}^q$ smooth mappings, and a positive definite and radially unbounded function $V : \mathbb{R}^n \rightarrow \mathbb{R}$, such that $V(x^*) = 0$ and

$$\frac{\partial V}{\partial x} f^*(x, \theta) < 0$$

for all $x \neq x^*$, where $f^*(x, \theta) = f_0(x) + g(x)(v_0(x) + v_1(x)\theta) + f_1(x)\theta$.

Then the update law

$$\dot{\hat{\theta}} = - \left(\frac{\partial V}{\partial x} g(x) v_1(x) \right)^\top$$

and the feedback law

$$u = v_0(x) + v_1(x)\hat{\theta}$$

are such that all trajectories of the closed-loop system are bounded and $\lim_{t \rightarrow \infty} x(t) = x^*$.

Theorem 2 Consider the system (6) and a point x^* . Assume there exists a known parameter controller $u = v(x, \theta)$ such that the closed-loop system

$$\dot{x} = f^*(x, \theta),$$

where $f^*(x, \theta) = f_0(x) + g(x)v(x, \theta) + f_1(x)\theta$, has a globally asymptotically stable equilibrium at x^* . Assume, in addition, that there exists a mapping $\beta : \mathbb{R}^n \rightarrow \mathbb{R}^q$ such that all trajectories of the system

$$\begin{aligned}\dot{z} &= -\left[\frac{\partial \beta}{\partial x} f_1(x)\right] z, \\ \dot{x} &= f^*(x) + g(x)(v(x, \theta + z) - v(x, \theta))\end{aligned}\quad (7)$$

are bounded and satisfy

$$\lim_{t \rightarrow \infty} [g(x(t))(v(x(t), \theta + z(t)) - v(x(t), \theta))] = 0.$$

Then the update law

$$\begin{aligned}\dot{\hat{\theta}} &= -\frac{\partial \beta}{\partial x} \left[f_0(x) + f_1(x)(\hat{\theta} + \beta(x)) \right. \\ &\quad \left. + g(x)v(x, \hat{\theta} + \beta(x)) \right]\end{aligned}$$

and the feedback law

$$u = v(x, \hat{\theta} + \beta(x))$$

are such that all trajectories of the closed-loop system are bounded and $\lim_{t \rightarrow \infty} x(t) = x^*$.

The stability properties of the adaptive closed-loop system in Theorem 1 can be studied with the Lyapunov function $W(x, \hat{\theta}) = V(x) + \frac{1}{2}\|\hat{\theta} - \theta\|^2$, whereas a Lyapunov analysis for the adaptive closed-loop system in Theorem 2 can be carried out, under additional assumptions, via a Lyapunov function of the form $W(x, \hat{\theta}) = V(x) + \frac{1}{2}\|\hat{\theta} - \theta + \beta(x)\|^2$. This suggests that in Theorem 1 $\hat{\theta}$ plays the role of the estimate of θ , whereas in Theorem 2 such a role is played by $\hat{\theta} + \beta(x)$. Note that in none of the two theorems, the parameter estimate is required to converge to the true value of the parameters, although in Theorem 2, the feedback law is required to converge, along trajectories, to the known parameter controller. This has a very important, possibly counterintuitive, consequence: the asymptotic nonadaptive controller $u = v(x, \hat{\theta}_\infty)$, where $\hat{\theta}_\infty = \lim_{t \rightarrow \infty} \hat{\theta}(t)$, provided the limit exists,

is not in general a stabilizing controller for system (6).

Example Consider the nonlinear system described by the equation $\dot{x} = u + \theta x^2$, with $x(t) \in \mathbb{R}$, $u(t) \in \mathbb{R}$, and $\theta \in \mathbb{R}$. A known parameter controller satisfying the assumptions of Theorem 1 (with $V(x) = x^2/2$) and of Theorem 2 (with $\beta(x) = x$) is $u = -x - \theta x^2$. The resulting update laws and feedback laws are (the subscripts “1” and “2” are used to refer to the construction in Theorems 1 and 2, respectively)

$$\dot{\hat{\theta}}_1 = x^3, \quad u_1 = -x - \hat{\theta} x^2,$$

and

$$\dot{\hat{\theta}}_2 = x, \quad u_2 = -x - (\hat{\theta} + x)x^2,$$

respectively. $\diamond$

The basic building blocks in Theorems 1 and 2 can be exploited repeatedly to design adaptive controllers for systems with a specific structure, for example, for systems described by the equations:

$$\begin{aligned}\dot{x}_1 &= x_2 + \varphi_1(x_1)^\top \theta, \\ \dot{x}_2 &= x_3 + \varphi_2(x_1, x_2)^\top \theta, \\ &\vdots \\ \dot{x}_i &= x_{i+1} + \varphi_i(x_1, \dots, x_i)^\top \theta, \\ &\vdots \\ \dot{x}_n &= u + \varphi_n(x_1, \dots, x_n)^\top \theta,\end{aligned}\quad (8)$$

with $x_i(t) \in \mathbb{R}$, for $i = 1, \dots, n$, $u(t) \in \mathbb{R}$, $\varphi_i : \mathbb{R}^i \rightarrow \mathbb{R}^q$, for $i = 1, \dots, n$, smooth mappings, and $\theta \in \mathbb{R}^q$. Note that the last of the Eqs. (8) can be replaced by

$$\dot{x}_n = \bar{\theta} u + \varphi_n(x_1, \dots, x_n)^\top \theta,$$

with $\bar{\theta} \in \mathbb{R}$, provided its sign is known (this latter requirement may be removed using the so-called Nussbaum gain). The parameter $\bar{\theta}$ is often referred to as the high-frequency gain of the system: a terminology borrowed from linear systems theory.

Output Feedback Adaptive Control

A key feature of the parameterized systems described so far is that these are linearly parameterized in θ . The linear parameterization allows developing systematic design tools, such as those given in Theorems 1 and 2. Such results, however, require full information on the state of the system. If only partial information on the state is available, one has to combine an estimator of the state with an update law. Such a combination requires either strong assumptions on the system or very specific structures. For example, it is feasible if the system is not only linearly parameterized in the parameter θ but it is also linearly parameterized in the unmeasured states, hence it is described by equations of the form:

$$\begin{aligned}\dot{x}_1 &= x_2 + \psi_1(x_1) + \varphi_1(x_1)^\top \theta, \\ \dot{x}_2 &= x_3 + \psi_2(x_1) + \varphi_2(x_1)^\top \theta, \\ &\vdots \\ \dot{x}_i &= x_{i+1} + \psi_i(x_1) + \varphi_i(x_1)^\top \theta + b_i u, \\ &\vdots \\ \dot{x}_{n-1} &= x_n + \psi_{n-1}(x_1) + \varphi_{n-1}(x_1)^\top \theta + b_{n-1} u, \\ \dot{x}_n &= \psi_n(x_1) + \varphi_n(x_1)^\top \theta + b_n u, \\ y &= x_1,\end{aligned}$$

with $x_i(t) \in \mathbb{R}$, for $i = 1, \dots, n$, $u(t) \in \mathbb{R}$, $y(t) \in \mathbb{R}$, $\varphi_i : \mathbb{R} \rightarrow \mathbb{R}^q$ and $\psi_i : \mathbb{R} \rightarrow \mathbb{R}$, for $i = 1, \dots, n$, smooth mappings, $\theta \in \mathbb{R}^q$ and $b = [b_1, \dots, b_{n-1}, b_n]^\top$ unknown, but such that the sign of b_n is known and the polynomial $b_n s^{n-i} + b_{n-1} s^{n-i-1} + \dots + b_i$ has all roots with negative real part: this implies that the system with input u and output y is minimum phase.

Nonlinear Parameterized Systems

Adaptive control of nonlinearly parameterized systems is an open area of research. The design of adaptive controllers relies often upon structural assumptions, for example the existence of a monotonic parameterization, as in the system described by the equation

$$\dot{x} = F(x, u) + \Phi(x, \theta),$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $\theta \in \mathbb{R}^q$, and $F : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $\Phi : \mathbb{R}^n \times \mathbb{R}^q \rightarrow \mathbb{R}^n$ smooth mappings, and such that, for all x , the mapping Φ satisfies the monotonicity condition

$$(\theta_a - \theta_b)^\top (\Phi(x, \theta_a) - \Phi(x, \theta_b)) > 0,$$

for all $\theta_a \neq \theta_b$. Alternatively, the design may exploit the so-called over-parameterization, for example, the equation of the system

$$\dot{x} = u + \psi_1(x) \sin \theta + \psi_2(x) \cos \theta,$$

with $x(t) \in \mathbb{R}$, $u(t) \in \mathbb{R}$, and $\theta \in \mathbb{R}$, may be rewritten in over-parameterized form as

$$\dot{x} = u + \psi_1(x)\theta_1 + \psi_2(x)\theta_2,$$

with $\theta_i \in \mathbb{R}$, for $i = 1, 2$. Note that the over-parameterized form *overlooks* the important information that $\theta_1^2 + \theta_2^2 = 1$.

Summary and Future Directions

The problem of adaptive stabilization for nonlinear systems has been discussed. Two conceptual building blocks for the design of stabilizing adaptive controllers have been discussed and classes of systems for which these blocks allow to explicitly design adaptive controllers have been given. The role of parameter convergence, or lack thereof, has been briefly discussed together with connections between adaptive control and observer designs. The difficulties associated with non-full state measurement and with nonlinear parameterization have been also briefly highlighted. Several problems have not been discussed, for example model reference adaptive control, robust adaptive control, *universal* adaptive controllers, the use of projections to incorporate prior knowledge on the parameter, and adaptive control problems for systems with time-varying parameters. Details on these can be found in the recommended reading list.

Cross-References

- [Adaptive Control: Overview](#)
- [History of Adaptive Control](#)
- [Stochastic Adaptive Control](#)
- [Switching Adaptive Control](#)

Recommended Reading and References

Classical references on adaptive control for nonlinear systems and on recent research directions are given below.

Bibliography

- Astolfi A, Karagiannis D, Ortega R (2008) Nonlinear and adaptive control with applications. Springer, London
- Hovakimyan N, Cao C (2010) L_1 Adaptive control theory. SIAM, Philadelphia
- Ilchmann A (1997) Universal adaptive stabilization of nonlinear systems. *Dyn Control* 7(3):199–213
- Ilchmann A, Ryan EP (1994) Universal λ -tracking for nonlinearly-perturbed systems in the presence of noise. *Automatica* 30(2):337–346
- Jiang Z-P, Praly L (1998) Design of robust adaptive controllers for nonlinear systems with dynamic uncertainties. *Automatica* 34(7):825–840
- Kanellakopoulos I, Kokotović PV, Morse AS (1991) Systematic design of adaptive controllers for feedback linearizable systems. *IEEE Trans Autom Control* 36(11):1241–1253
- Krstić M, Kanellakopoulos I, Kokotović P (1995) Nonlinear and adaptive control design. Wiley, New York
- Marino R, Tomei P (1995) Nonlinear control design: geometric, adaptive and robust. Prentice-Hall, London
- Nussbaum RD (1982) Some remarks on a conjecture in parameter adaptive control. *Syst Control Lett* 3(5):243–246
- Pomet J-B, Praly L (1992) Adaptive nonlinear regulation: estimation from the Lyapunov equation. *IEEE Trans Autom Control* 37(6):729–740
- Sastry SS, Isidori A (1989) Adaptive control of linearizable systems. *IEEE Trans Autom Control* 34(11):1123–1131
- Spooner JT, Maggiore M, Ordóñez R, Passino KM (2002) Stable adaptive control and estimation for nonlinear systems. Wiley, New York
- Townley S (1999) An example of a globally stabilizing adaptive controller with a generically destabilizing parameter estimate. *IEEE Trans Autom Control* 44(11):2238–2241

Nonlinear Filters

Frederick E. Daum
Raytheon Company, Woburn, MA, USA

Abstract

Nonlinear filters estimate the state of dynamical systems given noisy measurements related to the state vector. In theory, such filters can provide optimal estimation accuracy for nonlinear measurements with nonlinear dynamics and non-Gaussian noise. However, in practice, the actual performance of nonlinear filters is limited by the curse of dimensionality. There are many different types of nonlinear filters, including the extended Kalman filter, the unscented Kalman filter, and particle filters.

Keywords

Bayesian · Computational complexity · Curse of dimensionality · Estimation · Extended kalman filter · Non-Gaussian · Particle filter · Prediction · Smoothing · Stability · Unscented kalman filter

Description of Nonlinear Filters

Nonlinear filters are algorithms that estimate the state vector (x) of a nonlinear dynamical system given measurements of nonlinear functions of the state vector corrupted by noise. Such filters also quantify the uncertainty in the resulting estimate of the state vector (e.g., using the error covariance matrix). Some nonlinear filters compute the entire probability density of the state vector conditioned on the set of measurements available, rather than computing a point estimate of the state vector (e.g., conditional mean or maximum likelihood). For some applications the conditional probability density of x is highly non-Gaussian (e.g., strongly multimodal). Even if the measurement noise and the process noise and the initial uncertainty in

x are all Gaussian, the conditional density of x can be non-Gaussian, owing to the nonlinearities in the dynamics or measurements. The dynamical systems can evolve in continuous time or discrete time, and the measurements can be made in continuous time or at discrete times. The most popular nonlinear filter in practical applications is the extended Kalman filter (EKF), but there are many other families of nonlinear filters, including particle filters, unscented Kalman filters (UKFs), batch least squares, exact finite-dimensional filters, Gaussian sum filters, cubature Kalman filters, etc. Table 1 summarizes the most popular nonlinear filters. The theory for nonlinear filters is relatively simple (see Ho and Lee 1964), but the crucial practical issue is computational complexity, even today with fast modern inexpensive computers, e.g., graphical processing units (GPUs). See Ristic et al. (2004) for a book which is both accessible to engineers and thorough.

Bayesian Formulation of Filtering Problem

The Bayesian approach to nonlinear filters is by far the most popular formulation of the problem (see Ho and Lee 1964), and it has virtually eliminated all other competing theories, because it is simple, general, systematic, and useful. All ten nonlinear filters listed in Table 1 are Bayesian. The Bayesian approach uses a model of the dynamics of x as well as a model of the measurements. For example, discrete-time dynamics and measurement models are typically of the form

$$\begin{aligned}x(t_{k+1}) &= f(x(t_k), t_k) + w(t_k) \\z(t_k) &= h(x(t_k), t_k) + v(t_k)\end{aligned}$$

in which $x(t_k)$ is the d -dimensional state vector at time t_k , $z(t_k)$ is the m -dimensional measurement vector at time t_k , v is the measurement noise, and w is the so-called process noise. Both v and w are often modeled as Gaussian zero-mean random processes with statistically independent values at

distinct discrete times, but these models could be highly non-Gaussian with statistically correlated random values. The initial probability density of x before any measurements are available is also used in the Bayesian formulations. Real physical systems are most commonly modeled as evolving in continuous time using Itô stochastic differential equations:

$$dx = f(x(t), t)dt + dw$$

However, most engineers would rather think of the above Itô equation as an ordinary differential equation driven by Gaussian white noise:

$$dx/dt = f(x(t), t) + dw/dt$$

Mathematicians prefer the Itô equation to avoid the embarrassment that the time derivative of $w(t)$ does not exist. For details of stochastic calculus, see Jazwinski (1998). Such mathematical subtleties rarely cause any trouble in practical engineering applications. We emphasize, however, that it is important to correctly model continuous-time random processes for the evolution of the state vector (x) in many practical applications. Similarly, one can model measurements in continuous time using Itô calculus:

$$dz = h(x(t), t)dt + dv$$

Most engineers consider continuous-time measurement models as impractical and unnecessarily complicated mathematically, because digital computers always require discrete-time measurements and there are no practical analog computers that can be used for nonlinear filtering, owing to the overwhelming superiority of digital computers in terms of accuracy, stability, dynamic range, and flexibility. Nevertheless, there are many papers published by researchers using continuous-time measurement models. But the vast majority of practical papers on nonlinear filters use discrete time measurement models for obvious reasons. This contrasts sharply with the practical importance of correctly modeling continuous-time random processes for the evolution of the state vector (x).

Nonlinear Filters, Table 1 Summary of nonlinear filters

Nonlinear filter	Conditional probability density	Computational complexity	Comments	References
1. Extended Kalman filter (EKF)	Gaussian	d^3	Gives good accuracy for many practical applications but can be highly suboptimal in difficult problems	Gelb et al. (1974)
2. Unscented Kalman filter(UKF)	Gaussian	d^3	Often the UKF beats the EKF, but sometimes the EKF is better than the UKF; see Noushin (2008) for details	Julier and Uhlmann (2003)
3. Batch least squares	Gaussian	d^3	Often beats the EKF accuracy but can fail for multimodal or other strongly non-Gaussian densities	Sorenson (1980)
4. Particle filter	Arbitrary	Varies from d^3 to exponential in d , depending on many features of the problem	Often beats the EKF accuracy but can fail due to the curse of dimensionality and particle degeneracy and ill-conditioning	Doucet (2011)
5. Cubature Kalman filter	Gaussian	d^3	Sometimes beats the EKF and UKF for difficult nonlinear non-Gaussian problems, but not always	Haykin (2010)
6. Gaussian sum	Arbitrary	Varies from d^3 to exponential in d , depending on many features of the problem	Beats the EKF for certain difficult nonlinear non-Gaussian problems	Sorenson (1988)
7. Exact finite-dimensional filters	Exponential family	d^3	Beats the EKF for certain difficult nonlinear non-Gaussian problems	Daum (2005)
8. Implicit particle filters	Arbitrary	Suffers from the curse of dimensionality (i.e., computation time grows exponentially in d)	Only low-dimensional numerical examples have been published so far	Chorin (2009)
9. Particle flow filter	Arbitrary	Faster than standard particle filters by many orders of magnitude for high-dimensional problems (but unfortunately there is no explicit formula for computation time)	Beats the EKF by orders of magnitude for certain difficult nonlinear non-Gaussian problems	Daum (2013)
10. Numerical solution of Fokker-Planck equation	Arbitrary	Suffers from the curse of dimensionality (i.e., computation time grows exponentially in d)	Beats the EKF by orders of magnitude for certain difficult nonlinear non-Gaussian problems	Ristic (2004)

Nonlinear Filter Algorithms

There is no universally best nonlinear filter for all applications, and there is much debate about which is the best nonlinear filter for any given application. Even if we knew the best nonlinear filter for a given computer, the answer could be very different for a different computer; in particular, some filters can exploit massively parallel processing architectures, whereas others cannot. Research and development of nonlinear filters should continue rapidly for the foreseeable future. More generally, there is no universal theory of computational complexity for practical algorithms of this type; perhaps the closest approximation to such a theory is “information-based complexity” (IBC); e.g., see Traub and Werschulz (1998) and Dick et al. (2013). The estimation accuracy of x and the computational complexity of the nonlinear filter are intimately connected, as shown below for particle filters. There is no useful way to quantify the computational complexity of nonlinear filters without also quantifying estimation accuracy of x . This contrasts with standard computational complexity theory (e.g., P vs. NP) because we are interested in approximations rather than exact solutions. This is the basic idea of IBC. In practice, engineers compare the estimation accuracy and computational complexity of different nonlinear filters using Monte Carlo simulations for specific applications and specific computers.

The most active area of current research in nonlinear filters is focused on particle filters, which have the promise of optimal accuracy for essentially any nonlinear filter problem, at the cost of very high computational complexity for high-dimensional problems. In the early days (1994–2004), researchers often asserted that particle filters “beat the curse of dimensionality,” but it is well known today that this assertion is wrong (e.g., see Daum 2005). Unfortunately, there is no useful theory of computational complexity for particle filters, but rather the currently available theory gives asymptotic bounds on accuracy with generic “constants.” Such bounds on the variance of estimation error are generally

of the form c/N in which N is the number of particles and c is the generic so-called constant. But we know that the so-called constant actually varies by many orders of magnitude depending on the specifics of the problem, including the following: (1) dimension of the state vector being estimated, (2) uncertainty in the initial state vector, (3) measurement accuracy, (4) stability of the dynamical system that describes the time evolution of the state vector, (5) geometry of the conditional probability densities (e.g., unimodal, log-concave, multimodal, etc.), (6) Lipschitz constants of the log probability densities, (7) curvature of the nonlinear dynamics and measurements, (8) ill-conditioning of the Fisher information matrix for the estimation problem, etc. Moreover, there are no tight bounds on the so-called constant c for practical nonlinear filter problems, but rather the best bounds for simple MCMC problems are known to be 30 orders of magnitude too large; see Dick et al. (2013).

Discrete-Time Measurement Models

Research papers on nonlinear filters are often mathematically abstract, but advanced math is not required for practical engineering applications (e.g., see Ho and Lee 1964). In particular, one can avoid the advanced stochastic mathematics used for continuous-time measurements by using discrete-time measurements, which is the practical case of interest anyway, owing to the use of digital computers to implement such algorithms. The notion that continuous-time measurements results in simpler, better, or more elegant results for nonlinear filters is misleading; for example, we have the elegant innovation theory for continuous-time measurements (Kailath 1970), but this theory is not applicable for discrete-time measurements, likewise with the elegant formula for propagating the conditional mean for continuous-time measurements (the so-called Fujisaki-Kallianpur-Kunita formula). More generally, the simple discrete-time version of Bayes’ rule suffices for practical real-world engineering applications; there is rarely a need to employ the more complex continuous-time version.

The discrete time formula for Bayes' rule is simply

$$p(x(t_k), t_k | Z_k) = p(x(t_k), t_k | Z_{k-1}) p(z_k | x(t_k)) / p(z_k | Z_{k-1})$$

in which

$p(x(t_k), t_k | Z_k)$ = probability density of x at time t_k conditioned on Z_k ; this is also called the “posteriori probability density”

$x(t)$ = state vector of the dynamical system at time t

Z_k = set of all measurements up to and including time t_k

z_k = measurement vector at time t_k

$p(z_k | x(t_k))$ = probability density of z_k conditioned on $x(t_k)$; this is also called the “likelihood”

$p(A|B)$ = probability density of A conditioned on B

This is all one needs to know about Bayes' rule for practical engineering applications of nonlinear filtering; see Ho and Lee (1964). Bayes' rule is a simple formula that multiplies two probability densities and normalizes it by dividing by $p(z_k | Z_{k-1})$. In most applications, there is no need to normalize the density, and hence, Bayes' rule for the unnormalized conditional density is even simpler:

$$p(x(t_k), t_k | Z_k) = p(x(t_k), t_k | Z_{k-1}) p(z_k | x(t_k))$$

We see that Bayes' rule for the unnormalized conditional density is simply a multiplication of two densities (i.e., the likelihood and the prior).

Summary and Future Directions

In practical applications, the most popular nonlinear filter is the extended Kalman filter (EKF), followed by the unscented Kalman filter (UKF). These two filters give good accuracy and robust performance for many practical applications. The computational complexity of both the EKF and UKF grows as the cube of the dimension of the state vector, and hence, they are very practical to run in real

time on laptops or PCs for many real-world applications. But there are also many difficult nonlinear or non-Gaussian problems for which the EKF and UKF give suboptimal accuracy, and in some cases, they give surprisingly bad accuracy. The accuracy of optimal nonlinear filters is limited by the curse of dimensionality. We know how to write the equations for the optimal nonlinear filter, but the solution generally takes an exponentially increasing time to compute as the dimension of the state vector grows. There are many different kinds of nonlinear filters, and this is still an active field of research, as shown in Crisan and Rozovskii (2011). Future research is likely to exploit advances in computational complexity theory for approximation of functions in the style of information-based complexity (IBC) rather than P vs. NP theory. This is because we want good fast approximations rather than exact algorithms. A lucid introduction to IBC is Traub and Werschulz (1998), and recent work is surveyed in Dick et al. (2013). Another fruitful direction of research is to exploit the recent advances in transport theory, as explained in Daum (2013); the best introduction to transport theory is the book by Villani (2003), which is very accessible yet thorough. Research in exact finite-dimensional filters is difficult but could yield substantial improvements in accuracy and computational complexity; for example, see Benes (1981), Marcus (1984), and Daum (2005). Progress in nonlinear filter research could be inspired by many diverse fields, including fluid dynamics, quantum chemistry, quantum field theory, gauge theory, string theory, Lie superalgebras, Lie supergroups, and neuroscience. An important open research topic is the stability of nonlinear filters, which is obviously a fundamental limitation to good theoretical upper bounds on estimation error. We still do not have a practical theory of stability for nonlinear filters. Perhaps the closest approximation to such a theory is the lucid paper by van Handel (2010), which makes an interesting attempt at understanding the stability of nonlinear filters. In particular, van Handel's paper aims to generalize Kalman's theory of

stability for the Kalman filter by connecting stability with the essence of controllability and observability. A good survey of what is known about stability theory for nonlinear filters is given in various articles in Crisan and Rozovskiĭ (2011).

Cross-References

- [Estimation, Survey on](#)
- [Extended Kalman Filters](#)
- [Kalman Filters](#)
- [Particle Filters](#)

Bibliography

- Arasaratnam I, Haykin S, Hurd TR (2010) Cubature Kalman filtering for continuous-discrete systems. *IEEE Trans Signal Process* 58:4977–4993
- Benes V (1981) Exact finite-dimensional filters for certain diffusions with nonlinear drift. *Stochastics* 5:65–92
- Chorin A, Tu X (2009) Implicit sampling for particle filters. *Proc Natl Acad Sci* 106:17249–17254
- Crisan D, Rozovskiĭ B (eds) (2011) *Oxford handbook of nonlinear filtering*. Oxford University Press, Oxford
- Daum F (2005) Nonlinear filters: beyond the Kalman filter. *IEEE AES Magazine* 20:57–69
- Daum F, Huang J (2013a) Particle flow with non-zero diffusion for nonlinear filters. In: *Proceedings of SPIE conference, San Diego*
- Daum F, Huang J (2013b) Particle flow and Monge-Kantorovich transport. In: *Proceedings of IEEE FUSION conference, Singapore*
- Dick J, Kuo F, Peters G, Sloan I (eds) (2013) *Monte Carlo and quasi-Monte Carlo methods 2012*. *Proceedings of conference, Sydney*. Springer, Heidelberg
- Doucet A, Johansen AM (2011) A tutorial on particle filtering and smoothing: fifteen years later. In: Crisan D, Rozovskiĭ B (eds) *The Oxford handbook of nonlinear filtering*. Oxford University Press, Oxford pp 656–704
- Gelb A et al (1974) *Applied optimal estimation*. MIT, Cambridge
- Ho Y-C, Lee RCK (1964) A Bayesian approach to problems in stochastic estimation and control. *IEEE Trans Autom Control* 9:333–339
- Jazwinski A (1998) *Stochastic processes and filtering theory*. Dover, Mineola
- Julier S, Uhlmann J (2004) Unscented filtering and nonlinear estimation. *IEEE Proc* 92:401–422
- Kailath T (1970) The innovations approach to detection and estimation theory. *Proc IEEE* 58: 680–695
- Kushner HJ (1964) On the differential equations satisfied by conditional probability densities of Markov processes. *SIAM J Control* 2:106–119
- Marcus SI (1984) Algebraic and geometric methods in nonlinear filtering. *SIAM J Control Optim* 22: 817–844
- Noushin A, Daum F (2008) Some interesting observations regarding the initialization of unscented and extended Kalman filters. In: *Proceedings of SPIE conference, Orlando*
- Ristic B, Arulampalam S, Gordon N (2004) *Beyond the Kalman filter*. Artech House, Boston
- Sorenson HW (1974) On the development of practical nonlinear filters. *Inf Sci* 7:253–270
- Sorenson HW (1980) *Parameter estimation*. Marcel-Dekker, New York
- Sorenson HW (1988) Recursive estimation for nonlinear dynamic systems. In: Spall J (ed) *Bayesian analysis of time series and dynamic models*. Marcel-Dekker, New York, pp 127–165
- Stratonovich RL (1960) Conditional Markov processes. *Theory Probab Appl* 5:156–178
- Traub J, Werschulz A (1998) *Complexity and information*. Cambridge University Press, Cambridge
- van Handel R (2010) Nonlinear filters and system theory. In: *Proceedings of 19th international symposium on mathematical theory of networks and systems, Budapest*
- Villani C (2003) *Topics in optimal transportation*. American Mathematical Society, Providence
- Zakai M (1969) On the optimal filtering of diffusion processes. *Z fur Wahrscheinlichkeitstheorie und verw Geb* 11:230–243

Nonlinear Sampled-Data Systems

Dragan Nesić¹ and Romain Postoyan^{2,3}

¹Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC, Australia

²Université de Lorraine, CRAN, France

³CNRS, CRAN, France

Abstract

Sampled-data systems are control systems in which the feedback law is digitally implemented via a computer. They are prevalent nowadays due to the numerous advantages they offer compared to analog control. Nonlinear sampled-data systems arise in this context

when either the plant model or the controller is nonlinear. While their linear counterpart is now a mature area, nonlinear sampled-data systems are much harder to deal with and, hence, much less understood. Their inherent complexity leads to a variety of methods for their modeling, analysis, and design. A summary of these methods is presented in this entry.

Keywords

Discrete time · Nonlinear · Sampled data · Sampler · Zero-order hold

Introduction

Definition: A control system in which a continuous-time plant is controlled by a digital computer is referred to as a *sampled-data control system* or simply a *sampled-data system* (Chen and Francis 1994); see Fig. 1. *Nonlinear sampled-data systems* arise when either the model of the plant or the controller is nonlinear; otherwise the system is referred to as a linear sampled-data system.

Motivation: Sampled-data control is preferable to continuous-time (analog) control for a range of reasons including reduced cost, reduced wiring, more robust hardware, easier and more flexible programming, and so on. Nowadays, a large majority of controllers are implemented on digital computers, and, hence, sampled-data systems are prevalent in practice. On the other hand, nonlinear plant models are necessary in numerous applications when a wide range of operating conditions need to be considered or

when truly nonlinear phenomena, such as friction or state/input constraints, are not negligible. Hence, there are many situations where nonlinear plant models are essential, such as vertical takeoff and landing of an aircraft, robots, automotive engines, and biochemical reactors, to name a few. It has to be noted that the nonlinearity may also come from the controller even when we consider linear plants as it is the case in adaptive control or model predictive control with constraints, for example.

Structure of sampled-data systems: Figure 1 presents a typical structure of a sampled-data system which consists of a continuous-time plant, an analog-to-digital (A/D) converter (i.e., a sampler), a digital-to-analog (D/A) converter (i.e., a hold device), and a discrete-time controller.

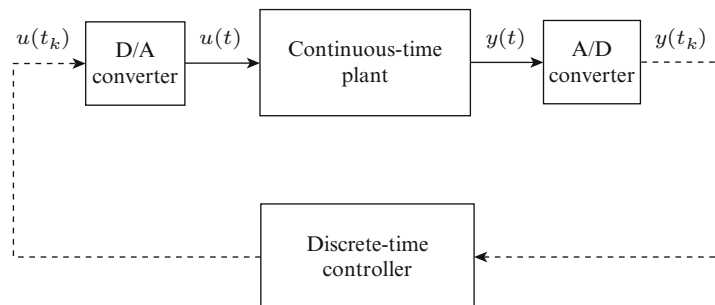
The A/D converter takes measurements $y(t_k)$ of a continuous-time output signal $y(t)$, such as temperature or pressure, at sampling time instants $t_k, k = 0, 1, \dots$ and sends them to the control algorithm. The measurements are obtained with finite precision (i.e., they are quantized); this effect is not considered in this entry. The sampling instants t_k are often equidistant, that is, $t_k = kT, k = 0, 1, \dots$, where the distance T between any two consecutive sampling instants is referred to as the *sampling period*. The sampling period is an important degree of freedom in the design of sampled-data systems and it needs to be carefully selected.

The control algorithm is discrete in nature. It takes the sequence of measurements $y(t_k)$ and processes them to produce a sequence of control values $u(t_k)$. The D/A converter converts the sequence of control values $u(t_k)$

Nonlinear Sampled-Data

Systems,

Fig. 1 Sampled-data (control) system



into a continuous-time signal $u(t)$ that drives the actuators which control the plant. Typically, a zero-order hold is used, i.e., $u(t) = u(t_k), \forall t \in [t_k, t_{k+1})$. However, it is possible to use other types of holds.

Note that the system in Fig. 1 can be generalized in many ways. An important generalization is *multi-rate sampling* where the output of the system is sampled at one sampling rate while the control inputs are updated at a different sampling rate. Another generalization are *networked control systems* which are discussed in the last section.

Modeling

The combination of continuous-time and discrete-time components renders the analysis and the design of sampled-data systems challenging. Still, linear systems allow for computationally efficient analysis and design techniques that benefit from the z and δ transforms, as well as convex optimization (Chen and Francis 1994). Nonlinear sampled-data systems, on the other hand, are much harder to deal with since the aforementioned methods do not apply in this case. This inherent difficulty has led to a variety of models for different analysis and design methods:

1. Continuous-time models
2. Discrete-time models
3. Sampled-data models

We discuss below each of these models, their features, and the analysis or design methods that exploit them.

Continuous-time models basically ignore the sampling process and assume that all signals are continuous time. They are the coarsest approximation of the sampled-data system and they are useful only for very small sampling periods. Nevertheless, they are invaluable and are used as the first step in the controller/observer design in the so-called *emulation* design approach.

Discrete-time models only capture the behavior of the sampled-data system at sampling

instants. Indeed, they ignore the inter-sample behavior of the system and this is their main drawback. There are two ways in which nonlinear discrete-time models arise: (i) from the identification of the plant model using the sampled measurements and (ii) from the discretization of a known continuous-time plant model. For instance, black box identification methods often lead to nonlinear discrete-time models in input-output form, such as NARMA (nonlinear auto-regressive moving average) models (Chen et al. 1989; Juditsky et al. 1995). Depending on the approximating functions used, the nonlinearities can be polynomial, neural network type, fuzzy type, and so on. On the other hand, the discretization of the continuous-time plant model requires an exact analytic solution of a set of nonlinear differential equations. When such an analytic solution exists, we can obtain the exact discrete-time models of the system; this is typically assumed for linear plants. Nonlinear sampled-data systems are different from their linear counterparts in that it is typically impossible to obtain the exact discrete-time model and only approximate discrete-time models are available for analysis and design (Nešić et al. 1999; Nešić and Teel 2004).

Sampled-data models capture the true behavior of the sampled-data system including its inter-sample behavior. There are several ways in which this can be achieved. One way is to model the piecewise constant signals that arise from zero-order hold devices as signals with a time-varying delay; this gives rise to time-delay nonlinear models (Teel et al. 1998). Another recently proposed approach is to model nonlinear sampled-data systems as hybrid dynamical systems (Goebel et al. 2012). An extensive analysis and design toolbox has been developed for hybrid dynamical systems and these results can be used for nonlinear sampled-data systems. Another class of models, based on the so-called lifting, has been applied for linear systems where the system is represented as a discrete-time system with infinite dimensional input and output spaces. While this approach has been very successful in the linear context (Chen and Francis 1994), it appears that it is not as useful for

nonlinear systems due to difficulties arising from harder analysis and prohibitive computational requirements.

The Main Issues and Analysis

Controllability/observability: Issues arising due to sampling in linear systems transfer to the nonlinear context although they are less understood in this case. For instance, it is well known that sampling may “destroy” the controllability and/or observability properties of the system (Chen and Francis 1994). In other words, if the continuous-time plant model is controllable/observable, then the corresponding exact discrete-time model of the plant may not verify these properties for some sampling periods. A simple test is available for linear systems to avoid this phenomenon, but we are not aware of similar results in the nonlinear context.

Finite escape times: A major difference between continuous-time linear and nonlinear systems is that the former have well defined solutions for constant control inputs and arbitrarily long sampling periods. This is not the case, in general, for nonlinear systems as they may exhibit finite escape times. In other words, for a constant input it may happen for some initial conditions of a nonlinear system that solutions blow up within a time that is shorter than the sampling period. As a consequence, for such an initial condition and input, the exact discrete-time system cannot be defined. This is a fundamental obstacle to achieving global stability results for nonlinear systems if the sampling period is fixed and independent of the size of the initial state. Nevertheless, it is possible to ensure semi-global stability properties for very general nonlinear systems which means that any compact domain of convergence can be achieved if the sampling period is sufficiently reduced (Nešić and Teel 2004).

Model structure is changed: An important issue for nonlinear sampled-data systems is that the sampling modifies the structure of the model. When the continuous-time plant model has a

certain structure, such as triangular or affine in the input, the corresponding exact discrete-time model will not inherit it; see Monaco and Normand-Cyrot (2007) and Yuz and Goodwin (2005). This significantly complicates the design of sampled-data systems via the discrete-time approach since many nonlinear design techniques, like backstepping or forwarding, are heavily reliant on the structure of the model.

Zero dynamics: Probably the most significant aspect of the changed structure are the so-called sampling zeros. In linear systems, it is well known that if a continuous-time linear system of relative degree $r \geq 2$ is sampled, then generically for fast sampling the discrete-time models of the plant will have relative degree $r = 1$. In other words, sampling introduces extra zeros in the model which are often unstable and thus render the system non-minimum phase. It is well known that the controller design is much harder for non-minimum phase systems, and, moreover, there are certain fundamental performance limitations in this case. Recently, results that extend the notion of sampling zeros to the nonlinear sampled-data systems have been reported; see the references in Monaco and Normand-Cyrot (2007).

Passivity: Some plant properties like passivity are much more restrictive in discrete time than in continuous time. Indeed, it is necessary for a continuous-time plant to have relative degree 1 or 0 to be passive, whereas only relative degree 0 discrete-time plants may possess this property. In other words, an exact discrete-time model of a passive continuous-time plant of relative degree 1 will not be passive; that is, sampling typically destroys passivity.

Controller Design

Linearization: The simplest way to design sampled-data nonlinear systems is to linearize the plant at a given operating point. In this case, the nonlinear plant dynamics are approximated by a linear model around a chosen equilibrium, and

then any of the linear sampled-data techniques can be applied to the linearized model. The obtained solution is then implemented on the true nonlinear plant. The drawback of this technique is that the solution would typically perform well only in the vicinity of the selected equilibrium point.

Nonlinear methods: An alternative is to perform designs that rely on a nonlinear plant model. These approaches can be divided into feedback linearization, emulation design method, (approximate and exact) discrete-time design method, and sampled-data design method.

Feedback linearization: Some classical problems, like feedback linearization, are harder for sampled-data systems than continuous-time ones. It was shown that a class of discrete-time nonlinear systems for which feedback linearization is possible is smaller than the corresponding class of continuous-time systems (Grizzle 1987). This has led to approximate feedback linearization techniques which consider achieving feedback linearization approximately with an error that can be reduced by reducing the length of the sampling period (Arapostathis et al. 1989).

Continuous-time design method (Emulation design): Emulation is a design technique consisting of two steps. In the first step, a continuous-time controller or observer is designed for the continuous-time plant while ignoring sampling to achieve appropriate stability, performance, and/or robustness guarantees. In the second step, the designed controller/observer is discretized for implementation and the sampling period is reduced sufficiently for the method to work. This method is approximate since the continuous-time plant model approximates well the sampled-data systems only for sufficiently small sampling periods. The discretization can be done using various implicit or explicit Runge-Kutta methods, such as the forward or backward Euler method (Monaco and Normand-Cyrot 2007; Yuz and Goodwin 2005). The emulation method is probably the best understood of all design methods. It was shown that a range of stability properties that can be cast in terms of dissipation inequalities are preserved in an appropriate sense under the emulation approach (Laila et al. 2002). Moreover,

nonconservative estimates of the upper bound for the required sampling period in emulation have been reported recently (Nešić et al. 2009).

Exact discrete-time design method: Exact discrete-time design method assumes that an exact discrete-time model of the plant is available to the designer; see Kötta (1995) and the references cited therein. This approach is reasonable when black box identification techniques are used for modeling. Moreover, in some rare cases it is possible to obtain the exact discrete-time model of the plant by integrating the continuous-time model with fixed inputs (assuming the zero-order hold is used). This is the case when the plant dynamics are linear while the control law is nonlinear (e.g., adaptive control) or the plant is linear with state/input constraints, which is a setup often used in the model predictive control. The literature on exact discrete-time design method is vast and many of the nonlinear continuous-time design techniques, like backstepping, forwarding, and passivity-based designs, are extended to discrete-time nonlinear systems; see Kötta (1995) and Grizzle (1987). A drawback of these methods is that they assume a special structure of the discrete-time nonlinear model, such as upper or lower triangular structure, which is typically much more restrictive in discrete-time than in continuous-time due to the loss of structure due to sampling that was discussed earlier.

Approximate discrete-time design method: Due to the nonlinearity, it is impossible in most cases to obtain an exact discrete-time plant model by integrating its continuous-time model equations; instead, a range of approximate discrete-time plant models, such as Runge-Kutta, can be used for controller/observer design. It was recently shown that this design method may lead to disastrous consequences where the controller stabilizes the approximate discrete-time plant model for all (arbitrarily small) sampling periods while the same controller destabilizes the exact discrete-time plant model for all sampling periods; see Nešić and Teel (2004) and Nešić et al. (1999). This is true even for linear systems and some commonly used

discretization techniques and controller designs. These considerations have led to the development of a framework for controller design based on approximate discrete-time models (Nešić et al. 1999; Nešić and Teel 2004). This framework provides checkable conditions on the continuous-time plant model, the approximate discrete-time model and the controller that guarantee that the controllers designed in this manner would stabilize the exact discrete-time model and, hence, the nonlinear sampled-data system for sufficiently small sampling periods. The design is based on families of approximate discrete-time models parameterized with the sampling period, and the design objectives are more demanding than in the continuous-time nonlinear systems. Ideas from numerical analysis are adapted to this context. This framework was used to design controllers and observers for classes of nonlinear sampled-data systems where typically Euler approximate discretization is employed to generate the approximate discrete-time model.

Sampled-data design method: Both emulation and discrete-time design methods have their drawbacks. Indeed, the former method ignores the sampling at the design stage, whereas the latter method ignores and may produce unacceptable inter-sampling behavior. Thus, methods that use a sampled-data model of the plant for design are much more attractive. There are two possible ways in which this can be achieved for nonlinear sampled-data systems.

The first approach consists of representing nonlinear sampled-data systems as systems with time-varying delays (Teel et al. 1998). However, controller design tools for such systems need to be further developed.

The second approach involves representing the nonlinear sampled-data system as a hybrid dynamical system. Recent advances on modeling and analysis of hybrid dynamical systems (Goebel et al. 2012) offer great opportunities in this context, but the full potential of this approach is still to be exploited. Nonlinear sampled-data systems are just a small subclass of hybrid dynamical systems, and developing specific analysis and design tools tailored to this class of systems seems promising.

It should be emphasized that there are many related techniques, such as discrete-time adaptive control and model predictive control, that deal with classes of nonlinear sampled-data systems but are not a part of the mainstream nonlinear sampled-data literature.

Summary and Future Directions

Summary: Sampled-data control systems are nowadays prevalent and there are many situations where nonlinear models need to be used to deal with wider ranges of operating conditions, more restrictive constraints, and enhanced performance specifications. Despite their increasing importance, the design of nonlinear sampled-data systems remains largely unexplored, and it is much less developed than its continuous-time counterpart. A variety of models, analysis, and design techniques make nonlinear sampled-data literature very diverse and a comprehensive textbook reference or a unifying approach is still missing. Many open questions remain for nonlinear sampled-data systems, such as results on multi-rate sampling, design techniques based on sampled-data models, and other generalizations which are discussed below.

Future Directions: In the 1990s, a new generation of digitally controlled systems has evolved from the more classical sampled-data systems which are generally referred to as networked control systems (NCS); see Heemels et al. (2010) and the references cited therein. These systems exploit digital wired or wireless communication networks within the control loops. Such a setup is introduced to reduce the cost, weight, and volume of the engineered systems, but its special structure imposes new challenges due to the communication constraints, data packet dropouts, quantization of data, varying sampling periods, time delays, etc. At the same time, these systems provide new flexibilities due to the distributed computation within the control system that can be used to improve the performance and mitigate some of the undesirable network effects on the overall system performance. Moreover,

embedded microprocessors allow for event-triggered and self-triggered sampling (Anta and Tabuada 2010) that are still largely unexplored especially for nonlinear systems. Design of NCS was identified as one of the biggest challenges to the control research community in the twenty-first century, and more than a decade of intense research on this topic still has not provided a comprehensive and unifying approach for their analysis and design. Novel results on modeling and Lyapunov stability theory for (nonlinear) hybrid dynamical systems appear to offer the right analysis design tools but they are still to be converted into efficient and easy-to-use design tools in the control engineers' toolbox.

Cross-References

- [Event-Triggered and Self-Triggered Control](#)
- [Hybrid Dynamical Systems, Feedback Control of](#)
- [Optimal Sampled-Data Control](#)
- [Sampled-Data Systems](#)

Bibliography

- Anta A, Tabuada P (2010) To sample or not to sample: self-triggered control for nonlinear systems. *IEEE Trans Autom Control* 55:2030–2042
- Arapostathis A, Jakubczyk B, Lee HG, Marcus SI, Sontag ED (1989) The effect of sampling on linear equivalence and feedback linearization. *Syst Control Lett* 13: 373–381
- Chen T, Francis B (1994) *Optimal sampled-data systems*. Springer, New York
- Chen S, Billings SA, Luo W (1989) Orthogonal least squares methods and their application to non-linear system identification. *Int J Control* 50: 1873–1896
- Goebel R, Sanfelice RG, Teel AR (2012) *Hybrid dynamical systems*. Princeton University Press, Princeton
- Grizzle JW (1987) Feedback linearization of discrete-time systems. *Syst Control Lett* 9:411–416
- Heemels M, Teel AR, van de Wouw N, Nešić D (2010) Networked control systems with communication constraints: tradeoffs between transmission intervals, delays and performance. *IEEE Trans Autom Control* 55:1781–1796
- Juditsky A, Hjalmarsson H, Benveniste A, Delyon B, Ljung L, Sjöberg J, Zhang Q (1995) Nonlinear black-box models in system identification: mathematical foundations (original research article). *Automatica* 31:1725–1750
- Khalil HK (2004) Performance recovery under output feedback sampled-data stabilization of a class of nonlinear systems. *IEEE Trans Autom Control* 49:2173–2184
- Kötta U (1995) Inversion method in the discrete-time nonlinear control systems synthesis problems. *Lecture notes in control and information sciences*, vol 205. Springer, Berlin
- Laila DS, Nešić D, Teel AR (2002) Open and closed loop dissipation inequalities under sampling and controller emulation. *Eur J Control* 18:109–125
- Monaco S, Normand-Cyrot D (2007) Advanced tools for nonlinear sampled-data systems' analysis and control. *Eur J Control* 13:221–241
- Nešić D, Teel AR (2004) A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models. *IEEE Trans Autom Control* 49:1103–1034
- Nešić D, Teel AR, Kokotović PV (1999) Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. *Syst Control Lett* 38:259–270
- Nešić D, Teel AR, Carnevale D (2009) Explicit computation of the sampling period in emulation of controllers for nonlinear sampled-data systems. *IEEE Trans Autom Control* 54: 619–624
- Teel AR, Nešić D, Kokotović PV (1998) A note on input-to-state stability of sampled-data nonlinear systems. In: *Proceedings of the conference on decision and control '98*, Tampa, pp 2473–2478
- Yuz JJ, Goodwin GC (2005) On sampled-data models for nonlinear systems. *IEEE Trans Autom Control* 50:477–1488

Nonlinear System Identification Using Particle Filters

Thomas B. Schön

Department of Information Technology, Uppsala University, Uppsala, Sweden

Abstract

Particle filters are computational methods opening up for systematic inference in nonlinear/non-Gaussian state-space models. The particle filter constitute the most popular sequential Monte Carlo (SMC) method. This

is a relatively recent development, and the aim here is to provide a brief exposition of these SMC methods and how they are key enabling algorithms in solving nonlinear system identification problems. The particle filters are important for both frequentist (maximum likelihood) and Bayesian nonlinear system identification.

Keywords

Bayesian · Backward simulation · Maximum likelihood · Markov chain Monte Carlo (MCMC) · Particle filter · Particle MCMC · Particle smoother · Sequential monte carlo

Introduction

The state-space model (SSM) offers a general tool for modeling and analyzing dynamical phenomena. The SSM consists of two stochastic processes: the states $\{\mathbf{x}_t\}_{t \geq 1}$ and the measurements $\{y_t\}_{t \geq 1}$, which are related according to

$$\mathbf{x}_{t+1} \mid (\mathbf{x}_t = x_t) \sim f_\theta(x_{t+1} \mid x_t, u_t), \quad (1a)$$

$$y_t \mid (\mathbf{x}_t = x_t) \sim h_\theta(y_t \mid x_t, u_t), \quad (1b)$$

and the initial state $\mathbf{x}_1 \sim \mu_\theta(x_1)$. We use boldface for random variables, and $\sim$ means “distributed according to.” The notation $\mathbf{x}_{t+1} \mid (\mathbf{x}_t = x_t)$ stands for the conditional probability of $\mathbf{x}_{t+1}$ given $\mathbf{x}_t = x_t$. The state process $\{\mathbf{x}_t\}_{t \geq 1}$ is a Markov process, implying that we only need to condition on the most recent state $\mathbf{x}_t$, since that contains all information about the past. Furthermore, θ denotes the parameters $f_\theta(\cdot)$ and $h_\theta(\cdot)$ that are probability density functions, encoding the dynamic and the measurement models, respectively. In the interest of a compact notation, we will suppress the input u_t throughout the text.

The SSM introduced in (1) is general in that it allows for nonlinear and non-Gaussian relationships. Furthermore, it includes both black box and gray box models on state-space form. Nonlinear black box and gray box models are

covered by ► [“Nonlinear System Identification: An Overview of Common Approaches.”](#) The off-line nonlinear system identification problem can (slightly simplified) be expressed as recovering information about the parameters θ based on the information in the T measured inputs $u_{1:T} \triangleq \{u_1, \dots, u_T\}$ and outputs $y_{1:T}$. For a thorough exposition of the system identification problem, we refer to ► [“System Identification: An Overview.”](#) Nonlinear system identification has a long history, and a common assumption of the past has been that of linearity and Gaussianity. This assumption is very restrictive, and we have now witnessed well over half a century of research devoted to finding useful approximate algorithms allowing this assumption to be weakened. This development has significantly intensified during the past two decades of research on sequential Monte Carlo (SMC) methods (including particle filters and particle smoothers). However, the use of SMC for nonlinear system identification is more recent than that. The aim here is to introduce the key ideas enabling the use of SMC methods in solving nonlinear system identification problems, and as we will see, it is not a matter of straightforward application. The development of SMC-based identification follows two clear trends that are indeed more general: (1) the problems we are working with are analytically intractable, and hence, the mind-set has to shift from searching for closed-form solutions to the use of *computational methods*, and (2) the new algorithms have basic building blocks that are themselves algorithms. Both these trends call for new developments.

Before the SMC methods are introduced in section [“Sequential Monte Carlo,”](#) their need is clearly explained by formulating both the Bayesian and the maximum likelihood identification problems in sections [“Bayesian Problem Formulation”](#) and [“Maximum Likelihood Problem Formulation,”](#) respectively. Solutions to these problems are then provided in sections [“Bayesian Solutions”](#) and [“Maximum Likelihood Solutions,”](#) respectively. Finally, we give some intuition for online (recursive) solutions in section [“Online Solutions,”](#) and in section [“Summary and Future Directions,”](#) we

conclude with a summary and directions for future research.

Bayesian Problem Formulation

In formulating the Bayesian problem, the parameters θ are modeled as unknown stochastic variables, i.e., the model (1) needs to be augmented with a prior density for the parameters $\theta \sim p(\theta)$. The aim in Bayesian system identification is to compute the posterior density of θ given the measurements $p(\theta | y_{1:T})$. More generally, we typically compute the joint posterior of the parameters θ and the states $\mathbf{x}_{1:T}$,

$$p(\theta, \mathbf{x}_{1:T} | y_{1:T}) = p(\mathbf{x}_{1:T} | \theta, y_{1:T})p(\theta | y_{1:T}). \quad (2)$$

By explicitly including the state variables $\mathbf{x}_{1:T}$ in the problem formulation according to (2), they take on the role of auxiliary variables. The reason for including the state variables $\mathbf{x}_{1:T}$ as auxiliary variables is that the alternative of excluding them would require us to analytically marginalize the states $\mathbf{x}_{1:T}$. This is not possible for the model (1) under study. However, once we have an approximation of $p(\theta, \mathbf{x}_{1:T} | y_{1:T})$ available, the density $p(\theta | y_{1:T})$ is easily obtained by straightforward marginalization.

Maximum Likelihood Problem Formulation

In formulating the maximum likelihood (ML) problem, the parameters θ are modeled as unknown deterministic variables. The ML formulation offers a systematic way of computing point estimates of the unknown parameters θ in a model, by making use of the information available in the obtained measurements $y_{1:T}$. The ML estimate is obtained by finding the θ that maximizes the so-called log-likelihood function, which is defined as

$$\ell_T(\theta) \triangleq \log p_\theta(y_{1:T}) = \sum_{t=1}^T \log p_\theta(y_t | y_{1:t-1}). \quad (3)$$

Note that we use θ as a subindex to denote that the corresponding probability density function

is parameterized by θ , analogously to what was done in (1). The one step ahead predictor $p_\theta(y_t | y_{1:t-1})$ is computed by marginalizing $p(y_t, x_t | y_{1:t-1}) = h_\theta(y_t | x_t)p_\theta(x_t | y_{1:t-1})$ w.r.t. x_t , i.e., integrating out x_t from $p(y_t, x_t | y_{1:t-1})$. To summarize, the ML estimate $\hat{\theta}^{\text{ML}}$ is obtained by solving the following optimization problem:

$$\hat{\theta}^{\text{ML}} \triangleq \arg \max_{\theta} \sum_{t=1}^T \log \int h_\theta(y_t | x_t) p_\theta(x_t | y_{1:t-1}) dx_t. \quad (4)$$

This problem formulation clearly reveals the important fact that the nonlinear state inference problem (here computing $p_\theta(x_t | y_{1:t-1})$) is inherent in any maximum likelihood formulation for identification of SSMs. For linear Gaussian models, the Kalman filter offers closed-form solutions for the state inference problem, but for nonlinear models, there are no closed-form solutions available.

Sequential Monte Carlo

Solving the nonlinear system identification problem implicitly requires us to solve various nonlinear state inference problems. We will, for example, need to approximate the smoothing density $p(\mathbf{x}_{1:T} | y_{1:T})$ and the filtering density $p(x_t | y_{1:t})$. The SMC samplers offer approximate solutions to these and other nonlinear state inference problems, where the accuracy is only limited by the available computational resources. This section only deals with the state inference problem, allowing us to drop the θ in the notation for brevity.

Most SMC samplers hinge upon importance sampling, motivating section “[Importance Sampling](#).” In section “[Particle Filter](#),” we make use of importance sampling in computing an approximation of the filtering density $p(x_t | y_{1:t})$, and in section “[Particle Smoother](#),” a particle smoothing strategy is introduced to approximately compute $p(\mathbf{x}_{1:T} | y_{1:T})$.

Importance Sampling

Let $\mathbf{z}$ be a random variable distributed according to some complicated density $\pi(\mathbf{z})$, and let $\varphi(\cdot)$ be some function of interest. Importance sampling offers a systematic way of evaluating integrals of the form

$$\mathbb{E}[\varphi(\mathbf{z})] = \int \varphi(\mathbf{z})\pi(\mathbf{z})d\mathbf{z}, \quad (5)$$

without requiring samples directly generated from $\pi(\mathbf{z})$. The density $\pi(\mathbf{z})$ is referred to as the *target* density, i.e., the density we are trying to sample from. The importance sampler relies on a *proposal* density $q(\mathbf{z})$, from which it is simple to generate samples, let $\mathbf{z}^i \sim q(\mathbf{z})$, $i = 1, \dots, N$. Since each sample $\mathbf{z}^i$ is drawn from the proposal density rather than from the target density $\pi(\mathbf{z})$, we must somehow account for this discrepancy. The so-called importance weights $\tilde{\mathbf{w}}^i = \pi(\mathbf{z}^i)/q(\mathbf{z}^i)$ encode the difference. By normalizing the weights $\mathbf{w}^i = \tilde{\mathbf{w}}^i / \sum_{j=1}^N \tilde{\mathbf{w}}^j$, we obtain a set of weighted samples $\{\mathbf{z}^i, \mathbf{w}^i\}_{i=1}^N$ that can be used to approximately evaluate the integral (5) resulting in $\mathbb{E}[\varphi(\mathbf{z})] \approx \sum_{i=1}^N \mathbf{w}^i \varphi(\mathbf{z}^i)$. For a general introduction to importance sampling we refer to Robert and Casella (2004).

Particle Filter

The solution to the nonlinear filtering problem is provided by the following two recursive equations:

$$p(x_t | y_{1:t}) = \frac{h(y_t | x_t)p(x_t | y_{1:t-1})}{p(y_t | y_{1:t-1})}, \quad (6a)$$

$$p(x_t | y_{1:t-1}) = \int f(x_t | x_{t-1}) p(x_{t-1} | y_{1:t-1}) dx_{t-1}. \quad (6b)$$

In the general case (1), there are no analytical solutions available for the above equations. The particle filter maintains an empirical approximation of the solution, which at time $t-1$ amounts to

$$\hat{p}^N(x_{t-1} | y_{1:t-1}) = \sum_{i=1}^N \mathbf{w}_{t-1}^i \delta_{\mathbf{x}_{t-1}^i}(x_{t-1}), \quad (7)$$

where $\delta_{\mathbf{x}_{t-1}^i}(x_{t-1})$ denotes the Dirac delta mass located at $\mathbf{x}_{t-1}^i$. Furthermore, $\mathbf{w}_{t-1}^i$ and $\mathbf{x}_{t-1}^i$ are referred to as the weights and the particles, respectively. We will now derive the particle filter by designing an importance sampler allowing us to approximately solve (6). The derivation is performed in an inductive fashion, starting by assuming that $p(x_{t-1} | y_{1:t-1})$ is approximated by (7). Inserting (7) into (6b) results in $\hat{p}^N(x_t | y_{1:t-1}) = \sum_{i=1}^N \mathbf{w}_{t-1}^i f(x_t | \mathbf{x}_{t-1}^i)$, which is used in (6a) to compute an *approximation* of the filtering density $p(x_t | y_{1:t})$ up to proportionality. Hence, this allows us to target $p(x_t | y_{1:t})$ using an importance sampler, where the form of $\hat{p}^N(x_t | y_{1:t-1})$ suggests that new samples can be proposed according to

$$\mathbf{x}_t^i \sim q(x_t | y_{1:t}) = \sum_{i=1}^N \mathbf{w}_{t-1}^i f(x_t | \mathbf{x}_{t-1}^i). \quad (8)$$

It is worth noting that we can obtain a more general algorithm by replacing $f(x_t | \mathbf{x}_{t-1}^i)$ in the above mixture with a density $q(x_t | \mathbf{x}_{t-1}^i, y_t)$. However, in the interest of a simple but still highly useful algorithm, we keep (8). The proposal density (8) is a weighted mixture consisting of N components, which means that we can generate a sample $\tilde{\mathbf{x}}_t^i$ from it via a two-step procedure: first we select which component to sample from, and secondly we generate a sample from that component. More precisely, the first part amounts to selecting one of the N particles $\{\mathbf{x}_{t-1}^i\}_{i=1}^N$ according to

$$\mathbb{P}(\tilde{\mathbf{x}}_{t-1} = \mathbf{x}_{t-1}^i | \{\mathbf{x}_{t-1}^j, \mathbf{w}_{t-1}^j\}_{j=1}^N) = \mathbf{w}_{t-1}^i,$$

where the selected particle is denoted as $\tilde{\mathbf{x}}_{t-1}$. By repeating this N times, we obtain a set of equally weighted particles $\{\tilde{\mathbf{x}}_{t-1}^i\}_{i=1}^N$, constituting an empirical approximation of $p(x_{t-1} | y_{1:t-1})$, analogously to (7). We can then draw $\mathbf{x}_t^i \sim f(x_t | \tilde{\mathbf{x}}_{t-1}^i)$ to generate a realization from the proposal (8). This procedure that turns a weighted set

Algorithm 1 Bootstrap particle filter (for $i = 1, \dots, N$)

1. **Initialization** ($t = 1$):
 - (a) Sample $\mathbf{x}_1^i \sim \mu(x_1)$.
 - (b) Compute the importance weights $\tilde{\mathbf{w}}_1^i = h(y_1 | \mathbf{x}_1^i)$ and normalize $\mathbf{w}_1^i = \tilde{\mathbf{w}}_1^i / \sum_{j=1}^N \tilde{\mathbf{w}}_1^j$.
 2. **For** $t = 2$ **to** T **do**:
 - (a) Resample $\{\mathbf{x}_{t-1}^i, \mathbf{w}_{t-1}^i\}$ resulting in equally weighted particles $\{\tilde{\mathbf{x}}_{t-1}^i, 1/N\}$.
 - (b) Sample $\mathbf{x}_t^i \sim f(x_t | \tilde{\mathbf{x}}_{t-1}^i)$.
 - (c) Compute the importance weights $\tilde{\mathbf{w}}_t^i = h(y_t | \mathbf{x}_t^i)$ and normalize $\mathbf{w}_t^i = \tilde{\mathbf{w}}_t^i / \sum_{j=1}^N \tilde{\mathbf{w}}_t^j$.
-

of samples into an unweighted one is commonly referred to as *resampling*.

Finally, using the approximation $\hat{p}^N(x_t | y_{1:t-1})$ in (6a) and the proposal density according to (8) allows us to compute the weights as $\tilde{\mathbf{w}}_t^i = h(y_t | \mathbf{x}_t^i)$. Once all the N weights are computed and normalized, we obtain a collection of weighted particles $\{\mathbf{x}_t^i, \mathbf{w}_t^i\}_{i=1}^N$ targeting the filtering density at time t . We have now (in a slightly nonstandard fashion) derived the so-called *bootstrap particle filter*, which was the first particle filter introduced by Gordon et al. (1993) two decades ago. Since the introduction of Algorithm 1, the surrounding theory and practice have undergone significant developments; see, e.g., Doucet and Johansen (2011) for an up-to-date survey. The weights $\{\mathbf{w}_{1:T}^i\}_{i=1}^N$ and the particles $\{\mathbf{x}_{1:T}^i\}_{i=1}^N$ are random variables, and in executing the algorithm, we generate one realization from these. This is a useful insight both when it comes to understanding but also when it comes to the analysis of the particle filters. There is by now a fairly good understanding of the convergence properties of the particle filter; see, e.g., Doucet and Johansen (2011) for basic results and further pointers into the literature.

Particle Smoother

A particle smoother is an SMC method targeting the joint smoothing density $p(x_{1:T} | y_{1:T})$ (or one of its marginals). There are several different strategies for deriving particle smoothers. Rather than mentioning them all, we introduce one powerful and increasingly popular strategy based on

backward simulation, giving rise to the family of *forward filtering/backward simulation* (FFBSi) samplers.

In an FFBSi sampler, the joint smoothing density $p(x_{1:T} | y_{1:T})$ is targeted by complementing a forward particle filter with a second recursion evolving in the time-reversed direction. The following factorization of the joint smoothing density

$$p(x_{1:T} | y_{1:T}) = \left(\prod_{t=1}^{T-1} p(x_t | x_{t+1}, y_{1:t}) \right) p(x_T | y_{1:T}),$$

immediately suggests a highly useful time-reversed recursion. Start by generating a sample $\tilde{\mathbf{x}}_T \sim p(x_T | y_{1:T})$. We then continue generating samples backward in time by sampling from the so-called backward kernel $p(x_t | x_{t+1}, y_{1:t})$ according to $\tilde{\mathbf{x}}_t \sim p(x_t | \tilde{x}_{t+1}, y_{1:t})$, for $t = T-1, \dots, 1$. The resulting sample $\tilde{\mathbf{x}}_{1:T} \triangleq (\tilde{x}_1, \dots, \tilde{x}_T)$ is then by construction a sample from the joint smoothing density. Hence, in performing M backward simulations, we obtain the following approximation of the joint smoothing density:

$$\hat{p}^M(x_{1:T} | y_{1:T}) = \sum_{i=1}^M \frac{1}{M} \delta_{\tilde{\mathbf{x}}_{1:T}^i}(x_{1:T}). \quad (9)$$

For details on how to design algorithms implementing the backward simulation strategy, derivations, properties, and references, we refer to the survey on backward simulation methods by Lindsten and Schön (2013).

Bayesian Solutions

Strategies

The posterior density (2) is analytically intractable, but we can make use of Markov chain Monte Carlo (MCMC) samplers to address the inference problem. An MCMC sampler allows us to approximately generate samples from an arbitrary target density $\pi(z)$. This is done by

simulating a Markov chain (i.e., a Markov process) $\{\mathbf{z}[r]\}_{r \geq 1}$, which is constructed in such a way that the stationary distribution of the chain is given by $\pi(\mathbf{z})$. The sample paths $\{\mathbf{z}[r]\}_{r=1}^R$ of the chain can then be used to draw inference about the target distribution. Two *constructive* ways of finding a suitable Markov chain to simulate are provided by the Metropolis Hastings (MH) and the Gibbs samplers, where the latter can be interpreted as a special case of the former. See, e.g., Robert and Casella (2004) for details on MCMC. A Gibbs sampler targeting $p(\theta, x_{1:T} \mid y_{1:T})$ is given by

- (i) Draw $\theta' \sim p(\theta \mid x_{1:T}, y_{1:T})$.
- (ii) Draw $x'_{1:T} \sim p(x_{1:T} \mid \theta', y_{1:T})$.

The second step is hard, since it requires us to generate a sample from the joint smoothing density. Simply replacing step (ii) with a backward simulator does not result in a valid method (Andrieu et al. 2010).

One interesting solution is provided by the family of particle MCMC (PMCMC) sampler, first introduced by Andrieu et al. (2010). PMCMC provides a systematic way of combining SMC and MCMC, where SMC is used to construct the proposal density for the MCMC sampler. The so-called *particle Gibbs* (PG) sampler resolves the problems briefly mentioned above by a nontrivial modification of the SMC algorithm. Introducing the PG sampler lies outside the scope of this work; we refer the reader to the groundbreaking work by Andrieu et al. (2010).

A Nontrivial Example

To place PMCMC in the context of nonlinear system identification, we will now solve a nontrivial identification problem. The PG sampler is used to compute the posterior density for a general Wiener model (linear Gaussian system followed by a static nonlinearity) (Giri and Bai 2010):

$$\mathbf{x}_{t+1} = (\mathbf{A} \ \mathbf{B}) \begin{pmatrix} \mathbf{x}_t \\ \mathbf{u}_t \end{pmatrix} + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \quad (10a)$$

$$\mathbf{z}_t = \mathbf{C}\mathbf{x}_t, \quad (10b)$$

$$\mathbf{y}_t = \mathbf{g}(\mathbf{z}_t) + \mathbf{e}_t, \quad \mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{r}). \quad (10c)$$

Based on observed inputs $u_{1:T}$ and outputs $y_{1:T}$, we wish to identify the model (10). We place a matrix normal inverse Wishart (MNIW) prior on $\{(\mathbf{A}, \mathbf{B}), \mathbf{Q}\}$, an inverse gamma prior on $\mathbf{r}$, and a Gaussian process (Rasmussen and Williams 2006) prior on the function $\mathbf{g}$, resulting in a semiparametric model. We can without loss of generality fix the matrix $\mathbf{C}$ according to $\mathbf{C} = (1, 0, \dots, 0)$. For a complete model specification, we refer to Lindsten et al. (2013).

The posterior distribution $p(\theta, x_{1:T} \mid y_{1:T})$ is computed using a newly developed PG sampler referred to as particle Gibbs with ancestor sampling (PGAS); see Lindsten et al. (2014). In the present experiment, we make use of $T = 1,000$ observations. The dimension of the state-space is 6, the linear dynamics contains complex poles resulting in oscillations as seen in Fig. 1, and the nonlinearity is non-monotonic; see Fig. 2. A subspace method is used to find an initial guess for the linear system, and the static nonlinearity is initialized using a linear function (i.e., a straight line).

It is worth pausing for a moment to reflect upon the posterior distribution $p(\theta, x_{1:T} \mid y_{1:T})$ that we are computing. The unknown “parameters” θ live in the space $\Theta = \mathbb{R}^{64} \times \mathcal{F}$, where $\mathcal{F}$ is an appropriate function space. The states $x_{1:T}$ live in the space $\mathbb{R}^{6 \times 1,000}$. Hence, $p(\theta, x_{1:T} \mid y_{1:T})$ is actually a rather complicated object for this example.

Using the PGAS sampler (with $N = 15$ particles), we construct a Markov chain $\{\theta[r], x_{1:T}[r]\}_{r=1}^R$ with $p(\theta, x_{1:T} \mid y_{1:T})$ as its stationary distribution. We run this Markov chain for $R = 25,000$ iterations, where the first 10,000 are discarded. The result is visualized in Figs. 1 and 2, where we plot the Bode diagram for the linear system and the static nonlinearity, respectively. In both figures we also provide the 99% Bayesian credibility interval. MATLAB code for Bayesian identification of Wiener models is available from user.it.uu.se/thosc112/research/software.html.

The resonance peaks are accurately modeled, but the result is less accurate at low frequencies (likely due to a lack of excitation). The fact that the posterior mean is inaccurate at low frequencies is encoded in our estimate of the posterior distribution as shown by the credibility intervals.

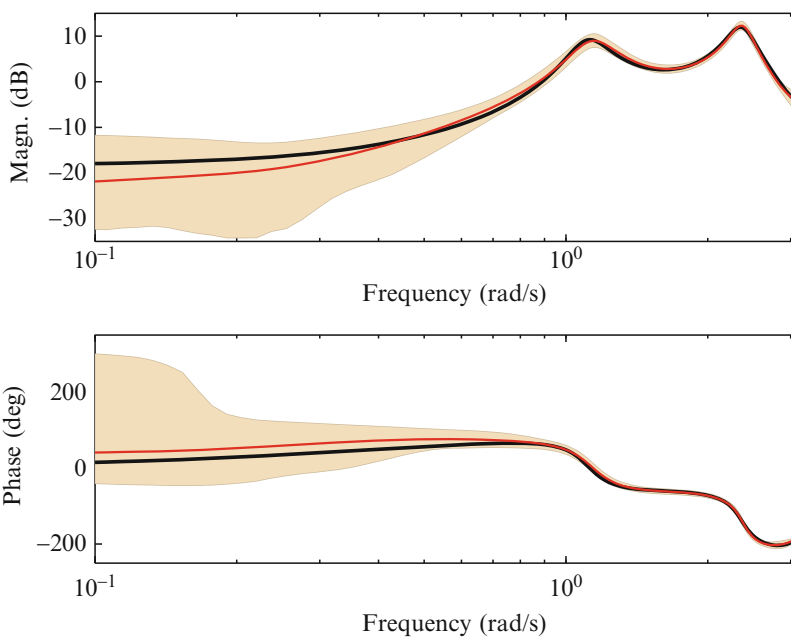
In Figs. 1 and 2, we have visualized not only the posterior mean but also the uncertainty for the entire model. We could do this since the model is a linear dynamical system followed by a static nonlinearity. It would be most interesting if we can come up with ways in which we could visualize the uncertainty inherent in general nonlinear dynamical systems.

Maximum Likelihood Solutions

Identifying the parameters θ in a general nonlinear SSM using maximum likelihood amounts to solving the optimization problem (3). This is a challenging problem for several reasons, for example, it requires the computation of the predictor density $p_\theta(y_t | y_{1:t-1})$. Furthermore, its

gradient (possibly also its Hessian) is very useful in setting up an efficient optimization algorithm. There are no closed-form solutions available for these objects, forcing us to rely on approximations. The SMC methods briefly introduced in section “[Sequential Monte Carlo](#)” provide rather natural tools for this task, since they are capable of producing approximations where the accuracy is only limited by the available computational resources.

To establish a clear interface between the maximum likelihood problem (3) and the SMC methods, it has proven natural to make use of the expectation maximization (EM) algorithm (Dempster et al. 1977). The EM algorithm proceeds in an iterative fashion to compute ML estimates of unknown parameters θ in probabilistic models involving latent variables. The strategy underlying the EM algorithm is to exploit the structure inherent in the probabilistic model to separate the original problem into two closely linked problems. The first problem amounts to computing the so-called *intermediate quantity*

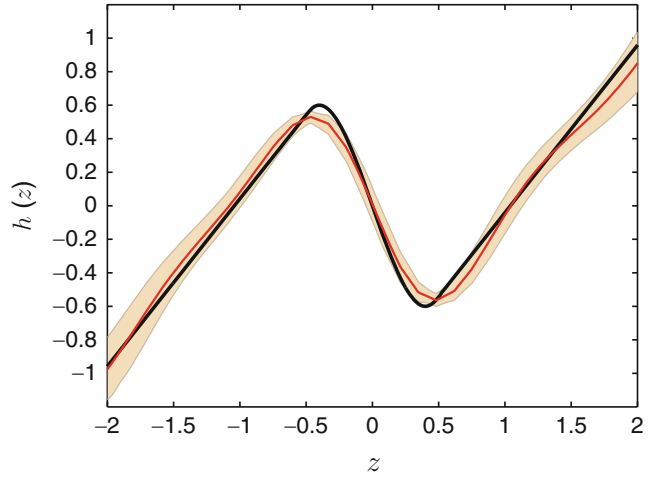


Nonlinear System Identification Using Particle Filters, Fig. 1 Bode diagram of the sixth-order linear system. The *black curve* is the true system. The *red curve* is

the estimated posterior mean of the Bode diagram, and the *shaded area* is the 99% Bayesian credibility interval

Nonlinear System Identification Using Particle Filters, Fig. 2

The *black curve* is the true static nonlinearity (non-monotonic). The *red curve* is the estimated posterior mean of the static nonlinearity, and the *shaded area* is the 99% Bayesian credibility interval



$$\begin{aligned} \mathcal{Q}(\theta, \theta') &\triangleq \int \log p_{\theta}(x_{1:T}, y_{1:T}) \\ &\quad p_{\theta'}(x_{1:T} | y_{1:T}) dx_{1:T} \\ &= E_{\theta'} [\log p_{\theta}(x_{1:T}, y_{1:T}) | y_{1:T}], \end{aligned} \quad (11)$$

where we have already made use of the fact that the latent variables in an SSM are given by the states. Furthermore, θ' denotes a particular value for the parameters θ . We can show that by choosing a new θ such that $\mathcal{Q}(\theta, \theta') \geq \mathcal{Q}(\theta', \theta')$, the likelihood is either increased or left unchanged, i.e., $\ell_T(\theta) \geq \ell_T(\theta')$.

The EM algorithm now suggests itself in that we can generate a sequence of iterates $\{\theta^k\}_{k \geq 1}$ that guarantees that the log-likelihood is not decreased for increasing k by alternating the following two steps: (1) (expectation) compute the intermediate quantity $\mathcal{Q}(\theta, \theta^k)$, and (2) (maximization) compute the subsequent iterate θ^{k+1} by maximizing $\mathcal{Q}(\theta, \theta^k)$ w.r.t. θ . This procedure is then repeated until convergence, guaranteeing convergence to a stationary point on the likelihood surface.

The FFBSi particle smoother offers an approximation of the joint smoothing density $p_{\theta'}(x_{1:T} | y_{1:T})$ according to (9), which inserted into (11) provides an approximative solution $\hat{\mathcal{Q}}^M(\theta, \theta')$ to the expectation step. In solving the maximization step, we typically want gradients of the intermediate quantity $\nabla_{\theta} \hat{\mathcal{Q}}^M(\theta, \theta')$. These can also

be approximated using (9). The above development is summarized in Algorithm 2, providing a solution where the basic building blocks are themselves complex algorithms, an SMC algorithm for the E step and a nonlinear optimization algorithm for the M step. This means that we have the option of replacing the FFBSi particle smoother in step 2a with any other algorithm capable of producing estimates of the joint smoothing density. The family of PMCMC methods introduced in section “Bayesian Solutions” contains several highly interesting alternatives. A detailed account on Algorithm 2 is provided by Schön et al. (2011); see also Cappé et al. (2005).

Finally, we mention Fisher’s identity opening up yet another avenue for designing ML estimators using SMC approximations. Even if

Algorithm 2 EM for nonlinear system identification

1. **Initialization:** Set $k = 0$ and initialize θ^k .
2. **Expectation (E) step:**
 - (a) Compute an approximation $\hat{p}_{\theta^k}^M(x_{1:T} | y_{1:T})$, for example, using an FFBSi sampler.
 - (b) Calculate $\hat{\mathcal{Q}}^M(\theta, \theta^k)$.
3. **Maximization (M) step:** Compute

$$\theta_{k+1} = \arg \max_{\theta} \hat{\mathcal{Q}}^M(\theta, \theta^k).$$

4. Check termination condition. If satisfied, terminate; otherwise, update $k \rightarrow k + 1$ and return to step 2.
-

we are not interested in using EM when solving the nonlinear system identification problem, the intermediate quantity (11) is useful. The reason is provided via *Fisher's identity*,

$$\begin{aligned}\nabla_{\theta} \ell_T(\theta) \Big|_{\theta=\theta'} &= \nabla_{\theta} \mathcal{Q}(\theta, \theta') \Big|_{\theta=\theta'} \\ &= \int \nabla_{\theta} \log p_{\theta}(x_{1:T}, y_{1:T}) \Big|_{\theta=\theta'} \\ &\quad p_{\theta'}(x_{1:T} \mid y_{1:T}) dx_{1:T},\end{aligned}$$

which provides a means to compute approximations of the log-likelihood gradient. Hessian approximations are also available, but these are more involved. Hence, Fisher's identity opens up for direct use of any off-the-shelf gradient-based optimization method in solving (4).

Online Solutions

Online (also referred to as recursive or adaptive) identification refers to the problem where the parameter estimate is updated based on the parameter estimate at the previous time step and the new measurement. This is used when we are dealing with big data sets and in real-time situations. SMC offers interesting opportunities when it comes to deriving online solutions for nonlinear state-space models. The most direct idea is simply to make use of a gradient method

$$\theta_t = \theta_{t-1} + \gamma_t \nabla_{\theta} \log p_{\theta}(y_t \mid y_{1:t-1}),$$

where $\{\gamma_t\}$ is the sequence of step sizes. Fisher's identity (12) opens up for the use of SMC in approximating $\nabla_{\theta} \log p_{\theta}(y_t \mid y_{1:t-1})$. However, this leads to a rapidly increasing variance, something that can be dealt with by the so-called "marginal" Fisher identity; see Poyiadjis et al. (2011) for details.

An interesting alternative is provided by an online EM algorithm; see, e.g., Cappé (2011) for a solid introduction. The online EM approaches rely on the additive properties of the $\mathcal{Q}$ -function. The area of online solutions via SMC is likely to grow in the future as there is a clear need

motivated by the constantly growing data sets and there are also clear theoretical opportunities.

Summary and Future Directions

We have discussed how SMC-based algorithms can be used to solve nonlinear system identification problems, by sketching both Bayesian and ML solutions. A common feature of the resulting algorithms is that they are (nontrivial) combinations of more basic algorithms. We have, for example, seen the combined use of a particle smoother and a nonlinear optimization solver in Algorithm 2 to compute ML estimates. As another example we have the class of PMCMC methods, where the basic building blocks are provided by SMC samplers and MCMC samplers. The use of SMC and MCMC methods for nonlinear system identification has only just started to take off, and it presents very interesting future prospects. Some directions for future research are as follows: (1) The family of PMCMC algorithms is rich and fast growing, with great potential for further developments. For example, its use in solving the state smoothing problem (i.e., computing $p(x_{1:T} \mid y_{1:T})$) is likely to provide better algorithms in the near future. (2) Related to this is the potential to design new particle smoothers capable of generating new particles also in the time-reversed direction. (3) There are open and highly relevant challenges when it comes to designing backward simulators for Bayesian nonparametric methods (Hjort et al. 2010). A key question here is how to represent the backward kernel $p(x_t \mid x_{t+1}, y_{1:t})$ in such nonparametric settings. (4) The use of Bayesian nonparametric models will open up interesting possibilities for hybrid system identification, since they allow us to systematically express and work with uncertainties over segmentations.

Cross-References

- [Nonlinear System Identification: An Overview of Common Approaches](#)
- [System Identification: An Overview](#)

Recommended Reading

An overview of SMC methods for system identification is provided by Schön et al. (2015) and Kantas et al. (2015), and a thorough introduction to SMC is provided by Doucet and Johansen (2011). Good textbook treatments of SMC and related technologies are provided by Särkkä (2013) and Douc et al. (2014). A self-contained introduction to particle smoothers and the backward simulation idea is provided by Lindsten and Schön (2013). The work by Cappé et al. (2005) also contains a lot of very relevant material in this respect.

Bibliography

- Andrieu C, Doucet A, Holenstein R (2010) Particle Markov chain Monte Carlo methods. *J R Stat Soc Ser B* 72(2):1–33
- Cappé O (2011) Online EM algorithm for hidden Markov models. *J Comput Graph Stat* 20(3):728–749
- Cappé O, Moulines E, Rydén T (2005) Inference in hidden Markov models. Springer, New York
- Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the EM algorithm. *J R Stat Soc Ser B* 39(1):1–38
- Douc R, Moulines E, Stoffer D (2014) Nonlinear time series: theory, methods and applications with R examples. CRC Press, Boca Raton/London/New York
- Doucet A, Johansen AM (2011) A tutorial on particle filtering and smoothing: fifteen years later. In: Crisan D, Rozovsky B (eds) *Nonlinear filtering handbook*. Oxford University Press, Oxford
- Giri F, Bai E-W (eds) (2010) Block-oriented nonlinear system identification. Lecture notes in control and information sciences, Vol 404. Springer, Berlin/Heidelberg
- Gordon NJ, Salmond DJ, Smith AFM (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc Radar Signal Process* 140:107–113
- Hjort N, Holmes C, Müller P, Walker S (eds) (2010) Bayesian nonparametrics. Cambridge University Press, Cambridge/New York
- Kantas N, Doucet A, Singh SS, Maciejowski JM, Chopin N (2015) On particle methods for parameter estimation in state-space models. *Stat Sci* 30(3):328–351
- Lindsten F, Schön TB (2013) Backward simulation methods for Monte Carlo statistical inference. *Found Trends Mach Learn* 6(1):1–143
- Lindsten F, Schön TB, Jordan MI (2013) Bayesian semi-parametric Wiener system identification. *Automatica* 49(7):2053–2063
- Lindsten F, Jordan MI, Schön TB (2014) Particle Gibbs with ancestor sampling. *J Mach Learn Res (JMLR)* 15:2145–2184
- Poyiadjis G, Doucet A, Singh SS (2011) Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika* 98(1):65–80
- Rasmussen CE, Williams CKI (2006) *Gaussian processes for machine learning*. MIT, Cambridge
- Robert CP, Casella G (2004) *Monte Carlo statistical methods*, 2nd edn. Springer, New York
- Särkkä S (2013) *Bayesian filtering and smoothing*. Cambridge University Press, Cambridge
- Schön TB, Wills A, Ninness B (2011) System identification of nonlinear state-space models. *Automatica* 47(1):39–49
- Schön TB, Lindsten F, Dahlin J, Wågberg J, Naesseth CA, Svensson A, Dai L (2015) Sequential Monte Carlo methods for system identification. In: *Proceedings of the 17th IFAC symposium on system identification*, Beijing, pp 775–786

Nonlinear System Identification: An Overview of Common Approaches

Qinghua Zhang
Inria, Campus de Beaulieu,
Rennes Cedex, France

Abstract

Nonlinear mathematical models are essential tools in various engineering and scientific domains, where more and more data are recorded by electronic devices. How to build nonlinear mathematical models essentially based on experimental data is the topic of this entry. Due to the large extent of the topic, this entry provides only a rough overview of some well-known results, from gray-box to black-box system identification.

Keywords

Black-box models · Block-oriented models · Gray-box models · Nonlinear system identification

Introduction

The wide success of linear system identification in various applications (Ljung 1999; ► [System Identification: An Overview](#)) does not necessarily mean that the underlying dynamic systems are intrinsically linear. Quite often, linear system identification can be successfully applied to a nonlinear system if its working range is restricted to a neighborhood of some working point. Nevertheless, some advanced engineering systems may exhibit significant nonlinear behaviors under their normal working conditions, so do most biological or social systems. There is therefore an increasing demand on nonlinear dynamic system modeling theory. Nonlinear system identification is studied to partly answer this demand, when experimental data carry the essential information for modeling purpose.

Nonlinear system identification, compared to its linear counterpart, is a much more vast topic, as in principle a nonlinear model can be *any* description of a system which is not linear. For this reason, this entry provides only a rough overview of some well-known results.

An overview of the basic concepts of system identification can be found in ► [System Identification: An Overview](#), notably the five basic elements to be taken into account in each application, among which the (nonlinear) model structures will be mainly focused on by this entry, as they represent the essential particularities of nonlinear system identification problems.

The various model structures used in nonlinear system identification are often classified by the level of available prior knowledge about the considered system: from *white-box* models to *black-box* models, via *gray-box* models. In principle, a white-box model is fully built from prior knowledge. Such a fully white-box approach is rarely feasible for complex systems because of insufficient prior knowledge or of intractable system complexity. Therefore, the system identification methods summarized in this entry concern gray-box and black-box models, for which experimental data play an essential role.

For ease of presentation, the main content of this entry will be restricted to the single-input

single-output (SISO) case. The multiple-input multiple-output (MIMO) case will be discussed in the section “[Multiple-Input Multiple-Output Systems](#).”

Gray-Box Models

This section covers gray-box models, from the most to the least demanding ones in terms of prior knowledge.

Parametrized Physical Models

The dynamic behaviors of some engineering systems are governed by well-known physical laws, typically in the form of differential equations, often with unknown parameters. These parametrized physical equations can be used as gray-box models for system identification. In most situations, such a model can be written in the form of a vectorial first-order ordinary differential equation (ODE), known as *state equation*, and can be generally written as

$$\frac{dx(t)}{dt} = f(x(t), u(t); \theta) \quad (1)$$

where t represents the time, $x(t)$ is the *state* vector, $u(t)$ is the *input*, and $f(\cdot)$ is a (nonlinear) function parametrized by the vector θ .

The observation on the system (typically with electronic sensors), referred to as the *output* and denoted by $y(t)$, is related to $x(t)$ and $u(t)$ through another known parametrized equation

$$y(t) = h(x(t), u(t); \theta) + v(t) \quad (2)$$

where $v(t)$ represents the measurement error.

With digital electronic instruments, the input $u(t)$ and the output $y(t)$ are sampled at some discrete-time instants, say $t = \tau, 2\tau, 3\tau, \dots, N\tau$ with some constant sampling period $\tau > 0$. For the sake of notation simplicity, let the sampling period $\tau = 1$ and assume ideal instantaneous samplers; then the sampled input-output data set is denoted by

$$Z^N = \{u(1), y(1), u(2), y(2), \dots, u(N), y(N)\}. \quad (3)$$

In some applications, data samples are made at irregular time instants. Some studies are particularly focused on system identification in this case (Garnier and Wang 2008).

The main remaining task of gray-box system identification is to estimate the parameter vector θ from the data set Z^N . The identification criterion is typically defined with the aid of an output predictor derived from the system model. A natural output predictor is simply based on the numerical solution of the state equation (1): for some given value of θ , initial state $x(0)$ and some assumed inter-sample behavior of the input $u(t)$ (e.g., with a zero order hold), the trajectory of $x(t)$, denoted by $\hat{x}(t|\theta)$, is computed with a numerical ODE solver, then the output prediction is computed as

$$\hat{y}(t|\theta) = h(\hat{x}(t|\theta), u(t); \theta). \quad (4)$$

The parameter vector θ is typically estimated by minimizing the sum of squared prediction error $\varepsilon(t|\theta) = y(t) - \hat{y}(t|\theta)$. See ► [System Identification: An Overview](#) and (Bohlin 2006) for more details.

The predictor based on the numerical solution of the state equation (1) (known as a *simulator*) may be in trouble if this equation with the given value of θ is unstable. Moreover, the state equation (1) may also be subject to some modeling error that should be taken into account in the output predictor. In such cases, the output predictor can be made with the aid of some nonlinear state observer (Gauthier and Kupka 2001) or some nonlinear filtering algorithm (Doucet and Johansen 2011).

Alternatively, sequential Monte Carlo (SMC) methods can also be applied to the identification of (small size) nonlinear state-space systems, typically assuming a discrete-time counterpart of the model described by Eqs. (1) and (2). See ► [Nonlinear System Identification Using Particle Filters](#).

The gray-box approach is particularly useful in engineering fields where some software libraries of commonly used components are

available. In this case, a system model can be built by connecting available component models. Nevertheless, the “connection” of the component models may introduce algebraic constraints through variables shared by connected components, leading to *differential algebraic equations* (DAE), which are a wider class of dynamic system models than the abovementioned state-space models (► [Modeling of Dynamic Systems from First Principles](#)). For most dynamic systems, it is possible to avoid the DAE formulation by causality analysis, so that the connections between different system components are treated as information flow, instead of algebraic constraints. There exist also some theoretic studies on DAE system identifiability (Ljung and Glad 1994) and some recent developments on the identification of such systems (Gerding et al. 2007).

Combined Physical and Black-Box Models

It may happen that, in a complex system, part of the components is well described by physical laws (possibly with available models from a software library), but some other components are not well studied. In this case, the latter components can be dealt with black-box models (or possibly empirical models). The entire model can be fitted to a collected data set Z^N , like in the case of the previous subsection.

Block-Oriented Models

Complex systems, notably those studied in engineering, are often made of a certain number of components; thus a system model can be built by connecting component models. In this sense, such component-based models could be said “block-oriented.” In the system identification literature, the term *block-oriented model* is often used in a particular context (Giri and Bai 2010), where it is typically assumed that each component is either a *linear dynamic* subsystem or a *nonlinear static* one. Here, the term “static” means that the behavior of the component is memoryless and can be described by an algebraic equation. The study of system identification with such models is motivated by the fact that, when a controlled system is stabilized around a working

point, its dynamic behavior can be well described by a linear model, but its actuators and sensors may exhibit significant nonlinear behaviors like saturation or dead zone. The choice of a particular block-oriented model structure depends on the prior knowledge about the underlying system, with specific identification methods available for different model structures.

The most frequently studied block-oriented models for system identification concern the Hammerstein system and the Wiener system, each composed of two blocks, as illustrated, respectively, in Figs. 1 and 2.

Hammerstein System Identification

A SISO Hammerstein system is typically formulated as

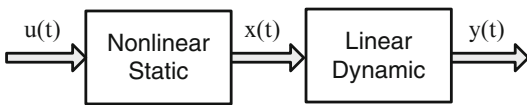
$$x(t) = f(u(t)) \quad (5a)$$

$$\begin{aligned} y(t) + a_1 y(t-1) + \cdots + a_{n_a} y(t-n_a) \\ = b_1 x(t-1) + \cdots + b_{n_b} x(t-n_b) \\ + v(t). \end{aligned} \quad (5b)$$

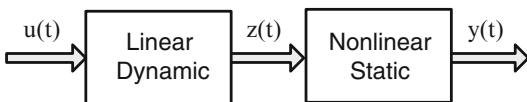
If the nonlinearity $f(\cdot)$ is expressed in the form of

$$f(u) = \sum_{l=1}^m \gamma_l \kappa_l(u) \quad (6)$$

with some chosen basis functions $\kappa_l(\cdot)$, then the identification problem amounts to fitting



Nonlinear System Identification: An Overview of Common Approaches, Fig. 1 Hammerstein system



Nonlinear System Identification: An Overview of Common Approaches, Fig. 2 Wiener system

the model parameters γ_l, a_i, b_j to a collected data set Z^N . A well-known method is based on over-parametrization (Bai 1998): replace in (5b) each $x(t-j)$ with the right-hand side of (6) and treat each parameter product $b_j \gamma_l$ as an individual parameter; then the newly parametrized model is equivalent to a linear ARX model (► [System Identification: An Overview](#)), which can be estimated by a well-established linear system identification method. As the $n_b + m$ parameters b_j and γ_l are replaced by $n_b m$ parameters in the new parametrization, the term “over-parametrization” refers to the fact that typically $n_b + m < n_b m$. The estimated over-parametrized model can be reduced to the original parametrization, usually through the singular value decomposition (SVD) of the matrix filled with the estimated parameter products $b_j \gamma_l$. See Giri and Bai (2010) for other identification methods with variant formulations of Hammerstein system model.

When the linear subsystem is approximated by a finite impulse response (FIR) model, it is possible to first estimate the linear model before estimating a model for the nonlinear block (Greblicki and Pawlak 1989).

Wiener System Identification

A SISO Wiener system is typically formulated as

$$z(t) = \sum_{k=1}^{\infty} h_k u(t-k) \quad (7a)$$

$$y(t) = g(z(t)) + v(t) \quad (7b)$$

where the sequence $h_1, h_2, \dots$ is the impulse response of the linear subsystem, $g(\cdot)$ is some nonlinear function, and $v(t)$ is a noise independent of the input $u(t)$.

Some methods for Wiener system identification assume a finite impulse response (FIR) of the linear subsystem. In this case, the linear subsystem model is characterized by the vector collecting the FIR coefficients $h^T = [h_1, h_2, \dots, h_n]$. There are two typical kinds of efficient solutions, assuming either the Gaussian distribution of the input $u(t)$ (Greblicki 1992) or the monotonicity

of the nonlinear function $g(\cdot)$ (Bai and Reyland 2008). In both cases, it is possible to directly estimate the FIR coefficients h from the input-output data Z^N , without explicitly estimating the unknown nonlinear function $g(\cdot)$. The estimated h can be used to compute the internal variable $z(t)$. It then becomes relatively easy to estimate the nonlinear function $g(\cdot)$ from the computed $z(t)$ and the measured $y(t)$.

Other Block-Oriented Model Structures

Among block-oriented models composed of more blocks, the most well-known ones concern Hammerstein-Wiener system and Wiener-Hammerstein system. They are both composed of three blocks connected in series, the former has a linear dynamic block preceded and followed by two nonlinear static blocks, and the latter has a nonlinear static block in the middle of two linear dynamic blocks. In general, the prediction error method (PEM) (Ljung 1999) is applied to the identification of such systems, with heuristic methods for the initialization of model parameters. Some recent results on Hammerstein-Wiener system identification have been reported in Wills et al. (2013). There exist also some other variants, with parallel blocks (Adil et al. 2018) or feedback loops (Xiao et al. 2018). In most cases, each block is either *linear dynamic* or *nonlinear static*, but there is a notable exception: *hysteresis* blocks. Hysteresis is a phenomenon typically observed in some magnetic or mechanic systems. Its mathematical description is both dynamic and strongly nonlinear and cannot be decomposed into linear dynamic and nonlinear static blocks. Due to the importance of hysteresis components in some control systems, system identification involving such blocks is still an active research topic.

LPV Models

Linear parameter-varying (LPV) models could be classified as black-box models, because typically they rely more on experimental data than on prior knowledge. However, engineers often have good insights into such models; they are thus presented in the gray-box section.

From Gain Scheduling to LPV Models

Gain scheduling is a method originally developed for the control of nonlinear systems. It consists in designing different controllers for different working points of a nonlinear system and in switching among the designed controllers according to the actual working point. It is typically assumed that the working point is determined by some observed variable (vector) referred to as the *scheduling variable* and denoted by ρ . Around each considered working point, the nonlinear system is linearized so that the corresponding controller can be designed from the linear control theory. A by-product of this controller design procedure is a collection of linearized models indexed by the scheduling variable ρ . This collection, seen as a whole model of the globally nonlinear system, is known as an LPV model (Toth 2010). This approach has been particularly successful in the field of flight control.

An LPV model can be formulated either in input-output form or in state-space form. In the input-output form, a SISO model can be written as

$$\begin{aligned} y(t) &+ a_1(\rho) y(t-1) + \dots + a_{n_a}(\rho) y(t-n_a) \\ &= b_1(\rho) u(t-1) + \dots \\ &+ b_{n_b}(\rho) u(t-n_b) + v(t) \end{aligned} \quad (8)$$

and in the state-space form as

$$x(t+1) = A(\rho)x(t) + B(\rho)u(t) + w(t) \quad (9a)$$

$$y(t) = c(\rho)x(t) + D(\rho)u(t) + v(t) \quad (9b)$$

As a global model of the whole nonlinear system, the ρ -dependent parameters (matrices) $a_i(\rho)$, $b_j(\rho)$, $A(\rho)$, etc. are functions defined for all $\rho \in \Omega$, where Ω is the relevant working range of the considered system (a compact subset of a real vector space). If originally the LPV model was built through a collection of linearized models around different working points, then the values of these functions are first defined for the corresponding discrete values of ρ . For other

values of $\rho \in \Omega$, these functions can be defined by interpolation. The interpolation of state-space models is a nontrivial problem (Zhang and Ljung 2017). Alternatively, by choosing some parametric forms of $a_i(\rho)$, $b_j(\rho)$, $A(\rho)$, etc., the whole LPV model can also be estimated by fitting it to a data set Z^N , through nonlinear optimization (Toth 2010).

Local Linear Models

In an LPV model, the model parameters can in principle depend on the scheduling variable ρ in any chosen manner. A particularly useful case is when they are formulated as expansions over local basis functions. For example, in (8), the parameter $a_i(\rho)$ may be expressed as

$$a_i(\rho) = \sum_{l=1}^m a_{i,l} \kappa_l(\rho) \quad (10)$$

where $\kappa_l(\cdot)$ are some chosen bell-shaped (local) basis functions, typically the Gaussian function, centered at different positions $\rho = c_l \in \Omega$, and $a_{i,l}$ are coefficients of the expansion. Similarly

$$b_j(\rho) = \sum_{l=1}^m b_{j,l} \kappa_l(\rho). \quad (11)$$

Assume that the basis functions are normalized such that

$$\sum_{l=1}^m \kappa_l(\rho) = 1 \quad (12)$$

for all $\rho \in \Omega$. Then the LPV model (8) can be viewed as an interpolation of m “local” models

$$\begin{aligned} y(t) &+ a_{1,l} y(t-1) + \dots + a_{n_a,l} y(t-n_a) \\ &= b_{1,l} u(t-1) + \dots + b_{n_b,l} u(t-n_b) + v(t) \end{aligned} \quad (13)$$

indexed by $l = 1, 2, \dots, m$. Each of these linear models is valid for ρ close to c_l , the center of the corresponding basis function $\kappa_l(\cdot)$; hence, they are called *local linear models*.

If the local basis functions $\kappa_l(\rho)$ are viewed as *membership functions* of fuzzy sets, then the local linear model is strongly related to the Takagi-Sugeno fuzzy model (Takagi and Sugeno 1985).

An advantage of this point of view is the possibilities of incorporating prior knowledge in the form of linguistic rules and of interpreting some local linear models resulting from system identification.

There are two approaches to building local linear models. The first one is the local approach: for each chosen value of $c_l \in \Omega$, a local model is estimated from data corresponding to the values of ρ within a neighborhood of c_l . This approach has the advantages of being computationally efficient, easily updatable, and well understood by engineers. The second one is the global approach: all the model parameters are estimated simultaneously by solving a single optimization problem for the whole model. This approach can produce more accurate models in terms of prediction error, but it is numerically much more expensive and may lead to models difficult to be interpreted by engineers.

The practical success of local linear models strongly depends on the possibility of finding a scheduling variable ρ of small dimension relevantly determining the working point of the considered system. If there exists a nonlinear state-space model of the system, then in principle the working point is determined jointly by the state and the input of the system. As quite often physically meaningful state variables are not fully observed, they cannot be used in the definition of ρ . It is possible to define ρ as delayed output and input variables, e.g.,

$$\rho^T = [y(t), \dots, y(t-n_a), u(t-1), \dots, u(t-n_b)]$$

but it typically leads to a vector of quite large dimension (Wang et al. 2017). It is thus important to use practical insights about a given system to find a relevant vector ρ of reduced dimension.

For a single-dimensional ρ , the choice of the local basis function centers c_l can be made following some practical insight or equally spaced within Ω . For a large-dimensional ρ , this task is more difficult. The equally spaced approach would lead to too many local models, as their number would exponentially increase with the dimension of ρ . In this case, an empirical approach, called *local linear model*

tree (LOLIMOT) (Nelles 2001), can be applied. It iteratively partitions Ω in order to place the local basis functions where the system is more likely nonlinear or where the available data are more concentrated.

Black-Box Models

Ideally speaking, a black-box model should be solely built from experimental data, without any prior knowledge. In practice, some prior knowledge is always necessary, though experimental data play a much more important role. For instance, the choice of the input and output variables, implying a causality relationship, involves some important prior knowledge.

With the fast development of electronic devices, more and more sensor signals are available in various fields, notably for engineering, environmental, and biomedical systems. Meanwhile, the processing power of modern computers increases every year. Black-box modeling has thus more and more potential applications. Nevertheless, the importance of prior knowledge in a modeling procedure should not be forgotten. In general prior knowledge leads to more reliable models in terms of validity range, as the validity of physical equations is often well understood. In contrast, for a black-box model essentially based on experimental data, it may be hard to ensure its validity for interpolation and even harder for extrapolation.

Input-Output Black-Box Models

As the primary role of a mathematical model is to predict the output of the system for given input values, it is natural to design black-box models directly in the form of a predictor. As the output $y(t)$ of a dynamic system depends on the past inputs, a predicted output $\hat{y}(t)$ may be formulated in the form of

$$\hat{y}(t) = f(u(t-1), u(t-2), \dots, u(t-n_b)) \quad (14)$$

where $f(\cdot)$ is some nonlinear function (to be estimated from experimental data) and n_b is a chosen integer. In principle, n_b can be infinitely

large (as $y(t)$ depends on *all* the past inputs in general), but in practice, a model of finite complexity has to be chosen. If the considered system is stable in the sense that sufficiently old past inputs are (gradually) forgotten, then it is reasonable to truncate the dependence on the past inputs.

The model structure (14) is similar to the linear finite impulse response (FIR) model (Ljung 1999). It is known that, for linear system identification, the use of ARX models, predicting $y(t)$ from both past inputs and past outputs, is often more efficient than FIR models, in the sense of requiring fewer model parameters. By analogy, the nonlinear ARX model takes the form

$$\hat{y}(t) = f(y(t-1), \dots, y(t-n_a), u(t-1), \dots, u(t-n_b)). \quad (15)$$

This nonlinear ARX model and some variants are likely the most frequently used black-box model structures for nonlinear dynamic system identification (Sjöberg et al. 1995; Juditsky et al. 1995; Billings 2013).

Nonlinear Function Estimators

For a nonlinear ARX model in the form of (15), the nonlinear function $f(\cdot)$ has to be estimated from an available input-output data set Z^N . Typically, an estimator of $f(\cdot)$ with some chosen parametric structure is used. Let

$$\phi^T(t) = [y(t-1), \dots, y(t-n_a), u(t-1), \dots, u(t-n_b)], \quad (16)$$

then system identification in this case amounts to solving a nonlinear regression problem

$$y(t) = g(\phi(t); \theta) + v(t) \quad (17)$$

where $g(\cdot)$ is a chosen nonlinear function parametrized by θ , capable of approximating a large class of nonlinear functions by appropriately adjusting θ , and $v(t)$ is the modeling error to be minimized in some sense.

The most well-known nonlinear function estimators implementing $g(\cdot)$ in practice are polynomials, splines, multiple-layer neural networks, radial basis networks, wavelets, and fuzzy-neural estimators. Most of these estimators can be written in the form

$$g(\phi(t); \theta) = \sum_{l=1}^m \gamma_l \kappa(\alpha_l(\phi - \beta_l)) \quad (18)$$

or in some close variant of this form, where $\kappa(\cdot)$ is some “mother” basis function dilated and translated by α_l and β_l before being weighted by γ_l in the sum forming the estimator (Sjöberg et al. 1995). For example, $\kappa(\cdot)$ is typically chosen as a (Gaussian) bell-shaped function in radial basis networks or a sigmoid (S-shaped) function in some multiple-layer neural networks.

Another approach to nonlinear function estimation is called *nonparametric estimation*. Its main idea is to estimate $f(\varphi^*)$ for any given value of φ^* by the (weighted) average of the values of $y(t)$ in the available data set corresponding to values of $\varphi(t)$ close to φ^* . This category includes *kernel* estimators (Nadaraya 1964) and *memory-based* estimators (Specht 1991).

The nonlinear function estimation problem as formulated in (17) can also be addressed with the Gaussian process model. Assume that g in (17) is a Gaussian process whose covariance matrix for any regressor pair $\varphi(t)$ and $\varphi(\tau)$ is a known function of the regressor pair, then the posterior distribution of g given observations on $y(t)$ can be computed by applying the Bayes’ theorem under certain assumptions (Rasmussen and Williams 2006). This method is strongly related to the least squares support vector machines (Suykens et al. 2002) and to some extent is similar to kernel estimators.

The difficulty for estimating a nonlinear function $f(\varphi)$ strongly depends on the dimension of φ . In the single-dimensional case, most existing methods can produce satisfactory results. When the dimension of φ , say n , increases, in order to keep the data “density” unchanged, the number of data points must increase exponentially with n . This fact implies that, in the high-dimensional

case (say $n > 10$), for most practically available data sets, the data points are sparse in the space of φ . It is thus practically impossible to estimate $f(\varphi)$ with a good accuracy everywhere in the space of φ . In order to remedy this problem, prior knowledge can be used to form a more elaborated vector φ of reduced dimension, instead of the simple form of past input and output variables. The resulting model will be more of gray-box nature. Recently, there have been attempts using *tensor networks* to reduce the complexity of nonlinearity estimators (Batselier et al. 2017; Gunes et al. 2018). A more direct approach is to expect that estimation algorithms automatically discover some low-dimension nature of the nonlinear relationship being estimated. The success would depend on the suitability of the chosen particular nonlinear function estimator for the considered system. This problem is strongly related to studies in machine learning (Pillonetto et al. 2014). The recent progress in deep learning has naturally inspired new studies in nonlinear system identification (De la Rosa and Yu 2016; Yu and Pacheco 2018; Woo et al. 2018).

State-Space Black-Box Models

For a gray-box model in the form of (1) and (2), it is assumed that the parametric forms of the nonlinear functions $f(\cdot)$ and $h(\cdot)$ are known from prior knowledge. If no such knowledge is available, it is possible to estimate these nonlinear functions with some function estimator, like those introduced in the previous subsection. Such an approach leads to state-space black-box models. In practice, it is easier to use the discrete-time counterpart of the state equation (1). Because typically the state vector $x(t)$ is not directly observed, the estimation of $f(\cdot)$ and $h(\cdot)$ cannot be formulated as nonlinear regression problems, in contrast to the case of input-output black-box models. Another difficulty is related to the nonuniqueness of the state-space representation of a given system: any (linear or nonlinear) state transformation would lead to a different state-space representation of the same system. In some existing methods, a linear state-space model is first estimated; then nonlinear function estimators

are used to compensate the residuals of $f(\cdot)$ and $h(\cdot)$ after their linear approximations (Paduart et al. 2010).

Multiple-Input Multiple-Output Systems

For multiple-input multiple-output (MIMO) systems, state-space models like (1)–(2) remain in the same form, by considering vector values of the notations $u(t)$ and $y(t)$ at each time instant, up to some similar adaptation of the other involved notations. For input-output models like (15), the involved notations can also be vector valued, but the fact that different inputs and/or outputs can have different delays makes the notations more complicated. For block-oriented models, though a MIMO linear block is usually described by a general linear model in state-space form or in input-output form, there seems no consensus for the structural choice of MIMO nonlinear blocks.

Some Practical Issues

The general practical aspects discussed in ► [System Identification: An Overview](#) are of course also valid for nonlinear system identification, but some particularities in the nonlinear case should be highlighted here.

It is important to apply appropriate input signals so that the collected data convey sufficient information for system identification. The design of input signals for this purpose is known as *experiment design*. In the framework of linear system identification, experiment design is usually formulated through the optimization of the covariance matrix of model parameter estimates (► [Experiment Design and Identification for Control](#)), which often leads to non-convex optimization problems. Experiment design in the nonlinear case has not been systematically studied. If possible, the chosen input signal should be similar to what will be actually applied to the considered system and cover various working conditions. Another simple rule is that the input should excite a nonlinear system at different amplitudes,

whereas binary input signals are often used for linear systems.

Model validation is a particularly delicate task for nonlinear black-box models. As already mentioned when such models were introduced, the available data points are usually sparse when a nonlinear function is estimated in a high-dimensional space; it is thus practically impossible to uniformly ensure the estimation accuracy of the nonlinear function. It is important to extensively perform *cross-validation*, by testing the validity of the model on large data sets that have not been used for model estimation.

Regularization is also an important issue for nonlinear black-box models. Because of lack of prior knowledge, each nonlinear black-box model has a flexible structure in order to cover a large class of nonlinear systems, typically with many model parameters, implying large variances of parameter estimates (► [System Identification: An Overview](#)). Appropriately applying a regularized criterion for model parameter estimation can reduce the variances (► [System Identification Techniques: Convexification, Regularization, Relaxation](#)). For gray-box models, prior knowledge can be used for regularization through a Bayesian approach, but this approach is not applicable to black-box models.

Summary and Future Directions

Compared to linear system identification, the nonlinear case is a much more vast topic, of which this entry provides only a rough overview. The main lines that should be retained are that both prior knowledge and experimental data are required for system identification, and that the more prior knowledge is incorporated in a model, the better the extent of its validity is understood. The lack of prior knowledge should be compensated by the processing of large amounts of data. The data that can be processed within an acceptable time depend on the power of computers that progresses every year. Meanwhile, the research and development

of efficient algorithms for large data processing with multiple or massively parallel processors are an exciting topic for system identification.

Cross-References

- [Experiment Design and Identification for Control](#)
- [Modeling of Dynamic Systems from First Principles](#)
- [Nonlinear System Identification Using Particle Filters](#)
- [System Identification: An Overview](#)
- [System Identification Techniques: Convexification, Regularization, Relaxation](#)

Recommended Reading

Nonlinear system identification is covered by a vast literature. After the readings about general topics on system identification (see ► [System Identification: An Overview](#) and references therein), the reader may further read (Nelles 2001; Billings 2013) for black-box system identification, (Bohlin 2006) for gray-box system identification, (Giri and Bai 2010) for block-oriented system identification, and (Toth 2010) for LPV system identification.

Bibliography

- Adil B, Mohamed B, Laila K (2018) Recursive parallel nonlinear systems identification. *Robot Autom Eng J* 2(4):1–3
- Bai E-W (1998) An optimal two-stage identification algorithm for Hammerstein-Wiener nonlinear systems. *Automatica* 34(3):333–338
- Bai E-W, Reyland J Jr (2008) Towards identification of Wiener systems with the least amount of a priori information on the nonlinearity. *Automatica* 44(4): 910–919
- Batselier A, Chen Z, Wong N (2017) A Tensor network alternating linear scheme for MIMO Volterra system identification. *Automatica* 84:26–35
- Billings SA (2013) Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains. Wiley, London
- Bohlin T (2006) Practical grey-box process identification – theory and applications. Springer, London
- Doucet A, Johansen AM (2011) A tutorial on particle filtering and smoothing: fifteen years later. In: Crisan D, Rozovsky B (eds) *Nonlinear filtering handbook*. Oxford University Press, Oxford
- Garnier H, Wang L (eds) (2008) Identification of continuous-time models from sampled data. Springer, London
- Gauthier J-P, Kupka I (2001) Deterministic observation theory and applications. Cambridge University Press, Cambridge/New York
- De la Rosa E, Yu W (2016) Randomized algorithms for nonlinear system identification with deep learning modification. *Inf Sci* 2016:197–212
- Gerdin M, Schön T, Glad T, Gustafsson F, Ljung L (2007) On parameter and state estimation for linear differential-algebraic equations. *Automatica* 43: 416–425
- Giri F, Bai E-W (eds) (2010) Block-oriented nonlinear system identification. Springer, Berlin/Heidelberg
- Giri F, Rochdi Y, Chaoui FZ, Brouri A (2008) Identification of Hammerstein systems in presence of hysteresis-backlash and hysteresis-relay nonlinearities. *Automatica* 44(3):767–775
- Greblicki W (1992) Nonparametric identification of Wiener systems. *IEEE Trans Inf Theory* 38(5): 1487–1493
- Greblicki W, Pawlak M (1989) Nonparametric identification of Hammerstein systems. *IEEE Trans Inf Theory* 35(2):409–418
- Gunes B, van Wingerden J-W, Verhaegen M (2018) Tensor networks for MIMO LPV system identification. *Int J Control* 2018:1–15
- Juditsky A, Hjalmarsson H, Benveniste A, Delyon B, Ljung L, Sjöberg J, Zhang Q (1995) Nonlinear black-box models in system identification: mathematical foundations. *Automatica* 31(11):1725–1750
- Ljung L (1999) *System identification – theory for the user*, 2nd edn. Prentice-Hall, Upper Saddle River
- Ljung L, Glad T (1994) On global identifiability for arbitrary model parametrizations. *Automatica* 30(2): 265–276
- Nadaraya EA (1964) On estimating regression. *Theory Probab Appl* 9:141–142
- Nelles O (2001) *Nonlinear system identification*. Springer, Berlin/New York
- Paduart J, Lauwers L, Swevers J, Smolders K, Schoukens J, Pintelon R (2010) Identification of nonlinear systems using polynomial nonlinear state space models. *Automatica* 46(4):647–656
- Rasmussen CE, Williams CKI (2006) *Gaussian processes for machine learning*. MIT, Cambridge
- Pillonetto G, Dinuzzo F, Chen T, De Nicolao G, Ljung L (2014) Kernel methods in system identification, machine learning and function estimation: a survey. *Automatica* 50(3):567–682
- Sjöberg J, Zhang Q, Ljung L, Benveniste A, Delyon B, Glorennec P-Y, Hjalmarsson H, Juditsky A (1995)

- Non-linear black-box modeling in system identifications unified overview. *Automatica* 31(11):1691–1724
- Specht DF (1991) A general regression neural network. *IEEE Trans Neural Netw* 2(5):568–576
- Suykens JAK, Van Gestel T, De Brabanter J, De Moor B, Vandewalle J (2002) Least squares support vector machines. World Scientific, Singapore
- Takagi T, Sugeno M (1985) Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans Syst Man Cybern* 15(1):116–132
- Toth R (2010) Modeling and identification of linear parameter-varying Systems. Springer, Berlin
- Wang L, Cheng Y, Hu J, Liang J, Abdullah MD (2017) Nonlinear system identification using quasi-ARX RBFN models with a parameter-classified scheme. *Complexity* 2017:1–12
- Wills A, Schön T, Ljung L, Ninness B (2013) Identification of Hammerstein-Wiener models. *Automatica* 49(1):70–81
- Woo J, Park J, Yu C, Kim N (2018) Dynamic model identification of unmanned surface vehicles using deep learning network. *Appl Ocean Res* 2018:123–133
- Xiao Z, Shan S, Chen L (2018) Identification of cascade dynamic nonlinear systems: a bargaining-game-theory-based approach. *IEEE Trans Signal Process* 66(17):4657–4669
- Yu W, Pacheco M (2018) Impact of random weights on nonlinear system identification using convolutional neural networks. *Inf Sci* 2019:1–14
- Zhang Q, Ljung L (2017) From structurally independent local LTI models to LPV model. *Automatica* 84: 232–235

Nonlinear Zero Dynamics

Alberto Isidori
Department of Computer and System Sciences
“A. Ruberti”, University of Rome “La
Sapienza”, Rome, Italy

Abstract

The notion of zero dynamics plays a role in nonlinear systems that is analogous to the role played, in a linear system, by the notion of zeros of the transfer function. In this article, we review the basic concepts underlying the definition of zero dynamics and discuss its relevance in the context of nonlinear feedback design.

Keywords

High-gain feedback · Inverse systems · Minimum-phase nonlinear systems · Normal forms · Output regulation · Stabilization

Introduction

The concept of zero dynamics of a nonlinear system was introduced in the early 1980s as the nonlinear analogue of the concept of transmission zero of a linear system. This concept played a fundamental role in the development of systematic methods for asymptotic stabilization of relevant classes of nonlinear systems. As a matter of fact, a nonlinear system in which the zero dynamics possess a globally asymptotically stable equilibrium can be robustly stabilized, globally or at least with guaranteed region of attraction, by means of output feedback. This is a nonlinear analogue of a well-know property of linear systems, namely, the property that an n -dimensional linear systems having $n - 1$ zeros with negative real part can be stabilized by means of proportional output feedback, if the feedback gain is sufficiently large. The concept of zero dynamics also plays a relevant role in variety of other problems of feedback design, such as input-output linearization with internal stability, non-interacting control with internal stability, output regulation, and feedback equivalence to passive systems.

The Zero Dynamics

One of the cornerstones of the geometric theory of control systems (for linear as well as for nonlinear systems) is the analysis of how the observability property can be influenced by feedback. This study, originally conceived in the context of the problem of disturbance decoupling, had far reaching consequences in a number of other domains. One of these consequences is the possibility of characterizing in “geometric terms” the notion of *zero* of the transfer function of a system. In a (single-input single-output and

minimal) linear system, a complex number z is a zero of the transfer function if and only if the input $u(t) = \exp(zt)$ yields – for a suitable choice of the initial state – a forced response in which the output is identically zero. This “open-loop” and “time-domain” characterization has a “closed-loop and “geometric” counterpart: all such z ’s coincide with the eigenvalues of the unobservable part of the system, once the latter has been rendered maximally unobservable by means of feedback. One of the earlier successes of the geometric approach to the analysis and design of nonlinear systems was the possibility of extending these equivalent characterizations to the domain of nonlinear systems.

To see how this is possible, consider for simplicity the case of a system modeled by equations of the form

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}$$

with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}$, output $y \in \mathbb{R}$ and in which $f(x), g(x), h(x)$ are smooth functions. Systems of this forms are called *input-affine systems*. The analysis of such systems is rendered particularly simple if appropriate notations are used. Given any real-valued smooth function $\lambda(x)$ and any n -vector valued smooth function $X(x)$, let $L_X \lambda(x)$ denote the (directional) derivative of $\lambda(x)$ along $X(x)$, that is the real-valued smooth function

$$L_X \lambda(x) = \sum_{i=1}^n \frac{\partial \lambda}{\partial x_i} X_i(x),$$

and, recursively, set $L_X^d \lambda = L_X L_X^{d-1} \lambda(x)$ for any $d \geq 1$.

Suppose there exists an integer $r \geq 1$ with the following properties

$$\begin{aligned}L_g h(x) &= L_g L_f h(x) = \cdots = L_g L_f^{r-2} h(x) \\ &= 0 \quad \forall x \in \mathbb{R}^n \\ L_g L_f^{r-1} h(x) &\neq 0 \quad \forall x \in \mathbb{R}^n.\end{aligned}$$

If this is the case, it is possible to show that the set

$$\begin{aligned}Z^* &= \{x \in \mathbb{R}^n : h(x) = L_h(x) = \cdots \\ &= L_f^{r-1} h(x) = 0\}\end{aligned}$$

is a smooth sub-manifold of $\mathbb{R}^n$, of codimension r . It is also easy to show that the state-feedback law

$$u^*(x) = -\frac{L_f^r h(x)}{L_g L_f^{r-1} h(x)}$$

renders the vector

$$f^*(x) = f(x) + g(x)u^*(x)$$

tangent to Z^* , at each point x of Z^* . In other words, Z^* is an *invariant* manifold of the feedback-modified system

$$\dot{x} = f^*(x).$$

It is seen from this construction that the output $y(t) = h(x(t))$ of the system is identically zero if and only if $x(0) \in Z^*$ and $u(t) = u^*(x(t))$, where $x(t)$ is the solution of $\dot{x} = f^*(x)$ passing through $x(0)$ at time $t = 0$. As a consequence, the *restriction* of $\dot{x} = f^*(x)$ to its invariant manifold Z^* characterizes all *internal* dynamics that occur in the system once initial condition and input are chosen in such a way that the output is constrained to be identically zero. The dynamics in question are called the *zero-dynamics* of the system. Note that this construction demonstrates, as anticipated, the equivalence between an “open-loop” and a “closed-loop” characterization of all the (internal) dynamics of a given system that are compatible with the constraint that the output is identically zero. This construction can be extended to multi-input multi-output systems, with the aid of an appropriate recursive algorithm, known as the *zero dynamics algorithm* (Isidori 1995).

Normal Forms

The coordinate-free construction presented above becomes even more transparent if special coordi-

nates are chosen. To this end, set

$$g^*(x) = \frac{1}{L_g L_f^{r-1} h(x)} g(x)$$

and define, recursively,

$X_0(x) = g^*(x)$, $X_k(x) = [f^*(x), X_{k-1}(x)]$, for $1 \leq k \leq r-1$, in which $[Y(x), X(x)]$ denotes the Lie bracket of $Y(x)$ and $X(x)$. It is possible to show that if the vector fields $X_0(x), \dots, X_{n-1}(x)$ are *complete*, there exists a smooth nonlinear, *globally defined*, change of variables by means of which the system can be transformed into a system of the form

$$\begin{aligned} \dot{z} &= f_0(z, \xi) \\ \dot{\xi} &= A_r \xi + B_r [q_0(z, \xi) + b(z, \xi)u] \\ y &= C_r \xi \end{aligned}$$

in which $z \in \mathbb{R}^{n-r}$, $\xi \in \mathbb{R}^n$, the matrices A_r, B_r, C_r have the form

$$\begin{aligned} A_r &= \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}, \quad B_r = \begin{pmatrix} 0 \\ 0 \\ \cdot \\ 0 \\ 1 \end{pmatrix}, \\ C_r &= (1 \ 0 \ 0 \ \dots \ 0), \end{aligned}$$

and $b(z, \xi) \neq 0$ for all (z, ξ) . These equations are said to be in *normal form* (Isidori 1995).

It is easy to check that, in these coordinates, the manifold Z^* is the set of all pairs (z, ξ) having $\xi = 0$, the state feedback law $u^*(x)$ is the function

$$u^*(z, \xi) = -\frac{q_0(z, \xi)}{b(z, \xi)}$$

and the restriction of $\dot{x} = f^*(x)$ to the manifold Z^* is nothing else than

$$\dot{z} = f_0(z, 0).$$

The latter provide a simple characterization of the zero dynamics of the system, once that the latter has been brought to its normal form.

It is worth observing that, in the case of a linear system, functions $f_0(z, \xi)$ and $q_0(z, \xi)$ are linear functions, and $b(z, \xi)$ is a constant. Consequently, the normal form can be written as

$$\begin{aligned} \dot{z} &= Fz + G\xi \\ \dot{\xi} &= A_r \xi + B_r [Hz + K\xi + bu] \\ y &= C_r \xi. \end{aligned}$$

It is also easy to check that the transfer function of the system can be expressed as

$$T(s) = b \frac{\det(sI - F)}{\det(sI - A)},$$

in which

$$A = \begin{pmatrix} F & G \\ B_r H & A_r + B_r K \end{pmatrix}.$$

From this it is concluded that in a (controllable and observable) linear system, the zeros of the transfer function $T(s)$ coincide with the eigenvalues of F . In other words, in a linear system the zero dynamics are linear dynamics whose eigenvalues coincide with the zeros of the transfer function of the system.

The Inverse System

Another property associated with the notion of zero of the transfer function, in a (single-input single-output) linear system, is the fact that the zeros characterize the dynamics of the inverse system (the latter being – loosely speaking – a system able to reproduce the input $u(t)$ from output $y(t)$ that this input has generated). This property has an immediate analogue for nonlinear systems. Considering system in normal form and setting

$$\mathbf{y}^{r-1}(t) = \text{col}(y(t), y^{(1)}(t), \dots, y^{(r-1)}(t)),$$

it is easily seen that the input $u(t)$ can be determined as the output of a dynamical system, driven by $\mathbf{y}^{r-1}(t)$ and $\mathbf{y}^{(r)}(t)$, modeled by

$$\begin{aligned}\dot{z} &= f_0(z, \mathbf{y}^{r-1}) \\ u &= \frac{\mathbf{y}^{(r)} - q_0(z, \mathbf{y}^{r-1})}{b(z, \mathbf{y}^{r-1})}.\end{aligned}\quad (1)$$

Thus, it is concluded that the unforced internal dynamics of the inverse system coincide with the zero dynamics as defined above.

It should be stressed, though, that the coincidence is limited to the case of single-input single-output systems. For a multi-input multi-output nonlinear systems, the link between zero dynamics and the dynamics of the inverse system is more subtle. This is essentially due to the fact that while the concept of zero dynamics only seeks to determine the dynamics compatible with the constraint that the output is identically zero, the inverse system must describe all dynamics resulting in *any* admissible output function. As a consequence, computation of the zero dynamics and computation of the inverse system (whenever this is possible) are not equivalent and the latter is possible only under substantially stronger assumptions. The computation of the zero dynamics is based on an extension (Isidori 1995) of the classical algorithm of Wonham (1979) for the computation of the largest controlled invariant subspace in the kernel of the output map, while the computation of the inverse system is based on extensions, due to Hirschorn (1979) and Singh (1981) of the so-called structure algorithm introduced by Silverman (1969) for the computation of inverses and zero structure at the infinity. For a comparison of such assumptions and of their influence on the outcome of the associated algorithms, see Isidori and Moog (1988).

Input-Output Linearization

An appealing feature of the normal form described above is the straightforward observation that a (state) feedback law of the form

$$u = \frac{1}{b(z, \xi)} [-q_0(z, \xi) + K_r \xi + v]$$

changes the system into a system

$$\begin{aligned}\dot{z} &= f_0(z, \xi) \\ \dot{\xi} &= (A_r + B_r K_r) \xi + B_r v \\ y &= C_r \xi\end{aligned}$$

whose input-output behavior (between input v and output y) is *fully linear* (and stable if K_r is chosen so that the matrix $A_r + B_r K_r$ in Hurwitz). In fact, the law in question renders the system partially unobservable, with all nonlinearities confined to its unobservable part (Isidori et al. 1981). This control law is clearly non-robust, as it relies upon exact cancelation of possibly uncertain terms, but it can be rendered robust by means of appropriate dynamic compensation (Freidovich and Khalil 2008).

The system obtained in this way has the structure of a cascade of two sub-systems, one of which, modeled as

$$\dot{z} = f_0(z, \xi),$$

is seen as “driven” by the input ξ . This motivates the interest in classifying the asymptotic properties of such subsystem, as discussed below.

Asymptotic Properties of the Zero Dynamics

Linear systems with no zeroes in the right-half complex plane are traditionally called *minimum-phase* systems, in view of certain properties of the Bode gain and phase plots of its transfer function. Thus, in view of the interpretation given above, linear systems whose zero dynamics are asymptotically stable are minimum-phase systems. This terminology has been (somewhat abusively, but with the clear intent of providing a concise and expressive characterization) borrowed to classify nonlinear systems whose zero dynamics have desirable (from the stability viewpoint) properties. Assuming that $z = 0$ is an equilibrium of

$\dot{z} = f_0(z, 0)$, the following cases are considered:

- A nonlinear system is *locally minimum-phase* (respectively, *locally exponentially minimum-phase*) if the equilibrium $z = 0$ of $\dot{z} = f_0(z, 0)$ is locally asymptotically (respectively locally exponentially) stable (Byrnes and Isidori 1984).
- A nonlinear system is *globally minimum-phase* if the equilibrium $z = 0$ of $\dot{z} = f_0(z, 0)$ is globally asymptotically stable (Byrnes and Isidori 1991).
- A nonlinear system is *strongly minimum-phase* if the system $\dot{z} = f_0(z, \xi)$, viewed as a system with input ξ and state z , is input-to-state stable (Liberzon 2002).

According to the well-known criterion of Sontag (1995) for input-to-state stability, a system is strongly minimum phase if and only if there exists a positive definite and proper smooth real-valued function $V(z)$, class $\mathcal{K}_\infty$ functions $\underline{\alpha}(\cdot), \bar{\alpha}(\cdot), \alpha(\cdot)$ and a class $\mathcal{K}$ function $\chi(\cdot)$ satisfying

$$\begin{aligned} \underline{\alpha}(|z|) &\leq V(z) \leq \bar{\alpha}(|z|) \quad \forall z \\ \frac{\partial V}{\partial z} f_0(z, \xi) &\leq -\alpha(|z|) \quad \forall (z, \xi) \\ \text{such that } |z| &\geq \chi(|\xi|). \end{aligned}$$

As a special case, it is seen that a system is globally minimum phase if and only if there exists a function $V(z)$, bounded as above, such that

$$\frac{\partial V}{\partial z} f_0(z, 0) \leq -\alpha(|z|) \quad \forall z.$$

If, instead, the weaker inequality

$$\frac{\partial V}{\partial z} f_0(z, 0) \leq 0 \quad \forall z$$

holds, the system is said to be *globally weakly minimum-phase*.

The criterion summarized above is of paramount importance in the design of feedback laws to the purpose of stabilizing nonlinear

systems that are globally (or strongly) minimum phase, as it will be seen below.

Zero Dynamics and Stabilization

The first and foremost immediate implication of the properties described above is the fact that the feedback law

$$u = \frac{1}{b(z, \xi)} [-q_0(z, \xi) + K_r \xi],$$

if K_r is chosen so that the matrix $A_r + B_r K_r$ in Hurwitz, globally asymptotically stabilizes the equilibrium $(z, \xi) = (0, 0)$ of a strongly minimum-phase system. In fact, as observed, the corresponding closed-loop system can be seen as an asymptotically stable (linear) system driving an input-to-state stable (nonlinear) system. As already observed, this control mode is non-robust (as it relies upon exact cancelations) and requires the availability of the full state (z, ξ) of the controlled system. However, both these deficiencies can be to some extent fixed, by means of appropriate techniques, that will be briefly reviewed below.

If the requirement of global stability is replaced by the (weaker) requirement of *stability with a guaranteed region of attraction*, then the desired control goal can be achieved by means of a much simpler law, depending only on the partial state ξ and not requiring cancelations. Stability with a guaranteed region of attraction essentially means that a given equilibrium is rendered asymptotically stable, with a region of attraction that contains an a priori *fixed* compact set. In this context, the most relevant results can be summarized as follows.

Assume the system possesses a globally defined normal form and, without loss of generality, let $b(z, \xi) > 0$. Let the system be controlled by a “partial state” feedback of the form

$$u = -k K_r \xi,$$

in which $k \in \mathbb{R}$. Under this control mode, the following results are obtained:

- Suppose the system is strongly minimum phase. Then, there is a matrix K_r and, for every choice of a compact set $\mathcal{C}$ and of a number $\varepsilon > 0$, there are a number k^* and a time T^* such that, if $k \geq k^*$, all trajectories of the closed-loop system with initial condition in $\mathcal{C}$ are bounded and satisfy $|x(t)| \leq \varepsilon$ for all $t \geq T^*$.
- Suppose the system is strongly minimum phase and also locally exponentially minimum phase. Suppose $q_0(0, 0) = 0$. Then, there is a matrix K_r and, for every choice of a compact set $\mathcal{C}$ there is a number k^* such that, if $k \geq k^*$, the equilibrium $x = 0$ of the system is locally asymptotically stable, with a domain of attraction that contains the set $\mathcal{C}$.

In these results, the system is stabilized by means of a *static* control law that depends only on the partial state ξ and not on the (possibly unknown) quantities $q_0(z, \xi)$, $b(z, \xi)$. Bearing in mind the fact that the r components of ξ coincide with the output y and its derivatives $y^{(1)}, \dots, y^{(r-1)}$, it is possible to replace the control in question by means of a *dynamic* control law that only depends on the output y , following a design paradigm originally proposed by H. Khalil. In fact, if the system is strongly minimum phase and also locally exponentially minimum phase and if $q_0(0, 0) = 0$, asymptotic stability with a guaranteed region of attraction can be achieved by means of dynamical feedback law of the form Khalil and Esfandiari (1993)

$$\begin{aligned}\dot{\hat{\xi}}_1 &= \hat{\xi}_2 + \kappa c_{r-1}(y - \hat{\xi}_1) \\ \dot{\hat{\xi}}_2 &= \hat{\xi}_3 + \kappa^2 c_{r-2}(y - \hat{\xi}_1) \\ &\dots \\ \dot{\hat{\xi}}_{r-1} &= \hat{\xi}_r + \kappa^{r-1} c_1(y - \hat{\xi}_1) \\ \dot{\hat{\xi}}_r &= \kappa^r c_0(y - \hat{\xi}_1) \\ u &= -\sigma_L(k K_r \hat{\xi}),\end{aligned}$$

in which κ and the c_i are design parameters and $\sigma_L(s)$ is a smooth saturation function, characterized as follows: $\sigma_L(s) = s$ if $|s| \leq L$, $\sigma_L(s)$

is odd and monotonically increasing, with $0 < \sigma'_L(s) \leq 1$, and $\lim_{s \rightarrow \infty} \sigma_L(s) = L(1 + c)$ with $0 < c \ll 1$. The number L is a design parameter also.

It is also possible to show that a suitable “extension” of this dynamic feedback law can be used to asymptotically recover the effects of the input-output linearizing law considered earlier. In this way, the lack of robustness intrinsically present in such control law is overcome (Freidovich and Khalil (2008)).

Output Regulation

The concept of zero dynamics plays a fundamental role in the problem of output regulation. The problem in question considers a controlled plant modeled by

$$\begin{aligned}\dot{x} &= f(w, x, u) \\ e &= h(w, x),\end{aligned}$$

in which u is the control input, w is a set of exogenous variables (command and disturbances), and e is a set of regulated variables. The exogenous variables are thought of as generated by an autonomous system

$$\dot{w} = s(w)$$

known as the *exosystem*. The problem is to design a (possibly dynamic) controller

$$\begin{aligned}\dot{x}_c &= f_c(x_c, e) \\ u &= h_c(x_c, e)\end{aligned}$$

driven by the regulated variable e , such that in the resulting closed-loop system all trajectories are ultimately bounded and $\lim_{t \rightarrow \infty} e(t) = 0$. The problem in question has been the object of intensive research in the past years. In what follows we limit ourselves to highlight the role of the concept of zero dynamics in this problem.

Assume that the set W where the exosystem evolves is compact and invariant and suppose a controller exists that solves the problem of output

regulation. Then, the associated closed-loop has a steady-state locus (see Isidori and Byrnes 2008), the graph of a possibly set-valued map defined on W . Suppose the map in question is single-valued, which means that for each given exogenous input function $w(t)$, there exists a *unique* steady-state response, expressed as $x(t) = \pi(w(t))$ and $x_c(t) = \pi_c(w(t))$. If, in addition, $\pi(w)$ and $\pi_c(w)$ are continuously differentiable, it is readily seen that

$$\begin{aligned} L_s \pi(w) &= f(w, \pi(w), \psi(w)) \\ 0 &= h(w, \pi(w)) \\ L_s \pi_c(w) &= f_c(\pi_c(w), 0) \\ \psi(w) &= h_c(\pi_c(w), 0) \end{aligned} \quad \forall w \in W.$$

The first two equations, introduced in Isidori and Byrnes (1990), are known as the *nonlinear regulator equations*. They clearly show that the graph of the map $\pi(w)$ is a manifold contained in the zero set of the output map e , rendered invariant by the control $u = \psi(w)$. In particular, the steady-state trajectories of the closed-loop system are trajectories of the *zero dynamics* of the controlled plant. The second two equations, on the other hand, interpret the ability, of the controller, to generate the feedforward input necessary to keep $e(t) = 0$ in steady-state. This is a nonlinear version of the well-known *internal model principle* of Francis and Wonham (1975).

Passivity

Consider a nonlinear input-affine system having the same number m of inputs and outputs and recall that this system is said to be *passive* if there exists a continuous nonnegative function real-valued function $W(x)$, with $W(0) = 0$, that satisfies

$$W(x(t)) - W(x(0)) \leq \int_0^t y^T(s)u(s)ds$$

along trajectories. The function $W(x)$ is the so-called *storage function* of the system.

It is well known that the notion of passivity plays an important role in system analysis and

that the theory of passive systems leads to powerful methodologies for the design of feedback laws for nonlinear systems. In this context, the question of whether a given, non-passive, nonlinear system could be rendered passive by means of state feedback is indeed relevant. It turns out that this possibility can be simply expressed as a property of the zero dynamics of the system.

Suppose that $L_g h(x)$ is nonsingular and set $g^*(x) = g(x)[L_g h(x)]^{-1}$. If the m columns of $g^*(x)$ are complete and commuting vector fields, there exists a globally defined change of coordinates that brings the system in normal form

$$\begin{aligned} \dot{z} &= f_0(z, y) \\ \dot{y} &= q_0(z, y) + b(z, y)u \end{aligned}$$

Then, there exists a feedback law $u = \alpha(z, y)$ that renders the resulting closed-loop system passive, with a C^2 and positive definite storage function $W(x)$, if and only if the system is globally weakly minimum phase (Byrnes et al. 1991).

Limits of Performance

It is well-known that linear systems having zeros in the left-half plane are difficult to control, and obstruction exists to the fulfillment of certain control specifications. One of these is found in the analysis of the so-called *cheap control problem*, namely, the problem of finding a stabilizing feedback control that minimizes the functional

$$J_\varepsilon = \frac{1}{2} \int_0^\infty [y^T(t)y(t) + \varepsilon u^T(t)u(t)]dt$$

when $\varepsilon > 0$ is small. As $\varepsilon \rightarrow 0$, the optimal value J_ε^* tends to J_0^* , the *ideal performance*. It is well-known that, in a linear system, $J_0^* = 0$ if and only if the system is minimum phase and right invertible and, in case the system has zeros with positive real part, it is possible to express explicitly J_0^* in terms of the zeros in question. If the (linear) system is expressed in normal form as

$$\begin{aligned}\dot{z} &= Fz + G\xi \\ \dot{\xi} &= Hz + K\xi + bu \\ y &= \xi\end{aligned}$$

with $b \neq 0$, and the zero dynamics are antistable (that is *all* the eigenvalues of F have positive real part), it can be shown that J_0^* coincides with the minimal value of the energy

$$J = \frac{1}{2} \int_0^\infty \xi^T(t) \xi(t) dt$$

required to stabilize the (antistable) system $\dot{z} = Fz + G\xi$. In other words, the limit as $\varepsilon \rightarrow 0$ of the optimal value of J_ε is equal to the least amount of energy required to stabilize the dynamics of the inverse system.

This result has an appealing nonlinear counterpart (Seron 1999). In fact, for a nonlinear input-affine system having the same number m of inputs and outputs in normal form, with $f_0(z, \xi)$ of the form $f_0(z, \xi) = f_0(z) + g_0(z)\xi$ and $\dot{z} = f_0(z)$ antistable, under appropriate technical assumptions (mostly related to the existence of the solution of the associated optimal control problems), the same result holds: the lowest attainable value of the L_2 norm of the output coincides with the least amount of energy required to stabilize the dynamics of z .

Summary and Future Directions

The concept of zero dynamics plays an important role in a large number of problems arising in analysis and design of nonlinear control systems, among which the most relevant ones are the problems of asymptotic stabilization and those of asymptotic tracking/rejection of exogenous command/disturbance inputs. Essentially, all such applications deal with single-input single-output systems, require the system to be preliminarily reduced to a special form by means of appropriate change of coordinates, and assume the dynamics in question to be globally asymptotically stable. The analysis of systems having many inputs and many outputs, of systems in which normal forms cannot be defined, and of

systems in which the zero dynamics are unstable is still a challenging and unexplored area of research.

Cross-References

- [Differential Geometric Methods in Nonlinear Control](#)
- [Input-to-State Stability](#)
- [Regulation and Tracking of Nonlinear Systems](#)

Bibliography

- Byrnes CI, Isidori A (1984) A frequency domain philosophy for nonlinear systems. *IEEE Conf Dec Control* 23:1569–1573
- Byrnes CI, Isidori A (1991) Asymptotic stabilization of minimum-phase nonlinear systems. *IEEE Trans Autom Control* AC-36:1122–1137
- Byrnes CI, Isidori A, Willems JC (1991) Passivity, feedback equivalence, and the global stabilization of minimum phase nonlinear aystems. *IEEE Trans Autom Control* AC-36:1228–1240
- Francis BA, Wonham WM (1975) The internal model principle for linear multivariable regulators. *J Appl Math Optim* 2:170–194
- Freidovich LB, Khalil HK (2008) Performance recovery of feedback-linearization-based designs. *IEEE Trans Autom Control* 53:2324–2334
- Hirschorn RM (1979) Invertibility for multivariable nonlinear control systems. *IEEE Trans Autom Control* AC-24:855–865
- Isidori A (1995) *Nonlinear control systems*, 3rd edn. Springer, Berlin/New York
- Isidori A, Byrnes CI (1990) Output regulation of nonlinear systems. *IEEE Trans Autom Control* AC-35:131–140
- Isidori A, Byrnes CI (2008) Steady-state behaviors in nonlinear systems, with an application to robust disturbance rejection. *Ann Rev Control* 32:1–16
- Isidori A, Moog C (1988) On the nonlinear equivalent of the notion of transmission zeros. In: CI Byrnes, A Kurzhanski (eds) *Modelling and adaptive control. Lecture notes in control and information sciences*, vol 105. Springer, Berlin/New York pp 445–471
- Isidori A, Krener AJ, Gori-Giorgi C, Monaco S (1981) Nonlinear decoupling via feedback: a differential geometric approach. *IEEE Trans Autom Control* AC-26:331–345
- Khalil HK, Esfandiari F (1993) Semiglobal stabilization of a class of nonlinear systems using output feedback. *IEEE Trans Autom Control* AC-38:1412–1415
- Liberzon D, Morse AS, Sontag ED (2002) Output-input stability and minimum-phase nonlinear systems. *IEEE Trans Autom Control* AC-43:422–436

- Seron MM, Braslavsky JH, Kokotovic PV, Mayne DQ (1999) Feedback limitations in nonlinear systems: from Bode integrals to cheap control. *IEEE Trans Autom Control* AC-44:829–833
- Singh SN (1981) A modified algorithm for invertibility in nonlinear systems. *IEEE Trans Autom Control* AC-26:595–598
- Silverman LM (1969) Inversion of multivariable linear systems. *IEEE Trans Autom Control* AC-14:270–276
- Sontag ED (1995) On the input–to–state stability property. *Eur J Control* 1:24–36
- Wonham WM (1979) *Linear multivariable control: a geometric approach*. Springer, New York

Nonparametric Techniques in System Identification

Rik Pintelon and Johan Schoukens
Department ELEC, Vrije Universiteit Brussel,
Brussels, Belgium

Abstract

This entry gives an overview of classical and state-of-the-art nonparametric time and frequency-domain techniques. In opposition to parametric methods, these techniques require no detailed structural information to get insight into the dynamic behavior of complex systems. Therefore, nonparametric methods are used in system identification to get an initial idea of the model complexity and for model validation purposes (e.g., detection of unmodeled dynamics). Their drawback is the increased variability compared with the parametric estimates. Although the main focus of this entry is on the classical identification framework (estimation of dynamical systems operating in open loop from known input, noisy output observations), the reader will also learn more about (i) the connection between transient and leakage errors, (ii) the estimation of dynamical systems operating in closed loop, (iii) the estimation in the presence of input noise, and (iv) the influence of nonlinear distortions on the linear framework. All results are valid for discrete- and continuous-time

systems. The entry concludes with some user choices and practical guidelines for setting up a system identification experiment and choosing an appropriate estimation method.

Keywords

Frequency response function · Noise (co) variances · Noise power spectrum · Correlation method · Gaussian process regression · Empirical transfer function estimate · Spectral analysis · Local polynomial method · Local rational method · Impulse transient response modeling method · Feedback · Errors-in-variables · Best linear approximation

Introduction

Nonparametric representations such as frequency response functions (FRFs) and noise power spectra are very useful in system identification: they are used (i) to verify the quality of the identification experiment (high or poor signal-to-noise ratio?), (ii) to get quickly insight into the dynamic behavior of the plant (complex or easy identification problem?), and (iii) to validate the parametric plant and noise models (detection of unmodeled dynamics); see also ► [“System Identification: An Overview”](#), by Lennart Ljung. In addition, via specially designed periodic excitation signals, it is possible to detect and quantify the nonlinear distortions in the FRF estimate. As such, without estimating a parametric model, the users can easily decide whether or not the linear framework is accurate enough for their particular application.

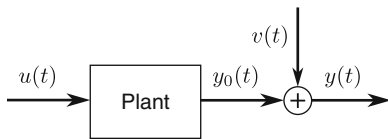
The estimation of the nonparametric models typically starts from sampled input-output signals $u(nT_s)$ and $y(nT_s)$, $n = 0, 1, \dots, N - 1$, that are transformed to the frequency domain via the discrete Fourier transform (DFT)

$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(nT_s) e^{-j2\pi kn/N} \quad (1)$$

with T_s the sampling period, $x = u$ or y , and $X = U$ or Y . One of the main difficulties

in estimating an FRF and noise power spectrum is the leakage error in the DFT spectrum $X(k) = \text{DFT}(x(t))$ (1). It is due to the finite duration NT_s of the experiment, and it increases the mean square error of the nonparametric estimates. Therefore, all methods try to suppress the leakage error as much as possible.

This entry starts by a detailed analysis of the leakage problem (section “[The Leakage Problem](#)”), followed by an overview of standard and advanced nonparametric time (section “[Nonparametric Time-Domain Techniques](#)”) and frequency (section “[Nonparametric Frequency-Domain Techniques](#)”) domain techniques. First, it is assumed that the system operates in open loop (see Fig. 1) and that known input, noisy output observations are available (sections “[Nonparametric Time-Domain Techniques](#)” and “[Nonparametric Frequency-Domain Techniques](#)”). Next, section “[Extensions](#)” extends the results to systems operating in closed loop (section “[Systems Operating in Feedback](#)”); to noisy input, noisy output observations (section “[Noisy Input, Noisy Output Observations](#)”); and to nonlinear systems (section “[Nonlinear Systems](#)”). Finally, some user choices are discussed (section “[User Choices](#)”) and some practical guidelines are given (section “[Guidelines](#)”). Unless otherwise stated, the input $u(t)$ and the disturbing noise $v(t)$ are assumed to be statistically uncorrelated.



Nonparametric Techniques in System Identification, Fig. 1 Classical identification framework: discrete- or continuous-time plant operating in open loop; known input $u(t)$, noisy output $y(t)$ observations; and $v(t)$ filtered discrete-time or band-limited continuous-time white noise $e(t)$ that is independent of $u(t)$. $y_0(t)$ denotes the true output of the plant. In the continuous-time case, it is assumed that the unobserved driving noise source $e(t)$ has finite variance and constant (*white*) power spectrum within the acquisition bandwidth

The Leakage Problem

For arbitrary excitations $u(t)$, the relationship between the true input $U(k)$ and true output $Y_0(k)$ DFT spectra (1) of a linear dynamic system is given by

$$Y_0(k) = G(\Omega_k)U(k) + T_G(\Omega_k) \quad (2)$$

where $\Omega_k = j\omega_k$ or $\exp(-j\omega_k T_s)$ for, respectively, continuous- and discrete-time systems; $\omega_k = 2\pi k/(NT_s)$; $G(\Omega_k)$ the plant frequency response function; and $T_G(\Omega_k)$ the leakage error due to the plant dynamics (Pintelon and Schoukens 2012, Section 6.3.2). The leakage error $T_G(\Omega)$ is a smooth function of the frequency that decreases to zero as $O(N^{-1/2})$ for N increasing to infinity. It depends on the difference between the initial and final conditions of the experiment and has exactly the same poles as the plant transfer function. Therefore, the time-domain response of $T_G(\Omega)$ is decaying exponentially to zero as a transient error.

From this short discussion, it can be concluded that the leakage error in the frequency domain is equivalent to the transient error in the time domain. The only difference being that the former depends on the difference between the initial and final conditions, while the latter solely depends on the initial conditions.

Standard spectral analysis methods (see section “[Spectral Analysis Method](#)”) suppress the leakage term $T_G(\Omega_k)$ in (2) by multiplying the time-domain signals with a window $w(t)$ before taking the DFT (1)

$$(X(k))_W = \frac{1}{\sqrt{N}w_{\text{rms}}} \sum_{n=0}^{N-1} w(nT_s) x(nT_s) e^{-j2\pi \frac{kn}{N}} \quad (3)$$

with $w_{\text{rms}} = \left(\sum_{n=0}^{N-1} |w(nT_s)|^2 / N \right)^{1/2}$ the root mean square (rms) value of the window $w(t)$. The scaling in (3) is such that the transformation preserves the rms value of the signal. The relationship between the DFT spectra $(U(k))_W$ and

$(Y_0(k))_W$ of the windowed input-output signals $w(t)u(t)$ and $w(t)y_0(t)$ is given by

$$(Y_0(k))_W = G(\Omega_k) (U(k))_W + E_{\text{int}}(k) + E_{\text{leak}}(k) \quad (4)$$

where $E_{\text{int}}(k)$ and $E_{\text{leak}}(k)$ are, respectively, the interpolation error and the remaining leakage error

$$\begin{aligned} E_{\text{int}}(k) &= (G(\Omega_k)U(k))_W - G(\Omega_k) (U(k))_W \\ E_{\text{leak}}(k) &= (T_G(\Omega_k))_W \end{aligned} \quad (5)$$

Note that $E_{\text{int}}(k) = 0$ if $G(\Omega_k)$ is constant within the bandwidth of $W(k)$, while the interpolation error is large if the FRF varies significantly within the window bandwidth. To keep $E_{\text{int}}(k)$ small, the frequency resolution $1/(NT_s)$ should be sufficiently large and the window bandwidth should be small enough. On the other hand, a larger window bandwidth is beneficial for reducing the leakage error $E_{\text{leak}}(k)$. Hence, choosing an appropriate window for nonparametric FRF and noise power spectrum estimation is making a trade-off between the reduction of the leakage error $E_{\text{leak}}(k)$ and the increase of the interpolation error $E_{\text{int}}(k)$ (Schoukens et al. 2006).

Note that exactly the same analysis can be made for the continuous- or discrete-time dynamics of the disturbing output noise $v(t)$ in Fig. 1

$$V(k) = H(\Omega_k)E(k) + T_H(\Omega_k) \quad (6)$$

with $H(\Omega_k)$ the noise frequency response function, $E(k)$ the DFT of the unobserved driving discrete-time or band-limited continuous-time white noise source $e(t)$ (Pintelon and Schoukens 2012, Section 6.7.3), and $T_H(\Omega_k)$ the noise leakage (transient) term. The noise leakage term is often neglected but can be important for lightly damped systems (e.g., in modal analysis). Most nonparametric techniques suppress the sum of the plant and noise leakage errors $T_G(\Omega_k) + T_H(\Omega_k)$.

If an integer number of periods of the steady-state response to a periodic excitation is

measured, then the plant leakage error $T_G(\Omega_k)$ in (2) is zero, which simplifies significantly the estimation problem. Therefore, for the frequency-domain techniques, a distinction is made between periodic and nonperiodic excitations. Note, however, that the noise leakage (transient) term $T_H(\Omega_k)$ in (6) remains different from zero.

Nonparametric Time-Domain Techniques

The time-domain methods estimate the impulse response of the plant via the time-domain relationship that the true output $y_0(t)$ equals the convolution product between the impulse response $g(t)$ and the true input $u(t)$. For discrete-time systems, it takes the form

$$y_0(t) = \sum_{n=0}^{\infty} g(n)u(t-n) \quad (7)$$

In practice only a finite number of impulse response coefficients $g(t)$ can be estimated from N input-output samples, and, therefore, (7) is approximated by a finite sum

$$y_0(t) \approx \sum_{n=0}^L g(n)u(t-n) \quad (8)$$

where $L \leq N-1$ should also be determined from the data. From (8), it can be seen that the response depends on the past input values $u(-1), u(-2), \dots, u(-L)$. Since these values are unknown, an exponentially decaying transient error is present in the first L samples of the predicted output (8). This transient error is the time-domain equivalent of the leakage error $T_G(\Omega_k)$ in (2). To remove the transient error, the first L output samples can be discarded in the predicted output (8). It reduces the amount of data from N to $N-L$ and, hence, increases the mean square error of the estimates. If it is known that the transfer function has no direct term, then $g(0) = 0$, and the sum (8) starts from $n = 1$.

Correlation Methods

Correlation methods have been studied intensively since the end of the 1950s (see Eykhoff 1974) and are nowadays still used in telecommunication channel estimation and equalization. The impulse response coefficients are found by minimizing the sum of the squared differences between the observed output samples and the output samples predicted by (8)

$$\sum_{t=L}^{N-1} \left(y(t) - \sum_{n=0}^L g(n) u(t-n) \right)^2 \quad (9)$$

w.r.t. $g(m)$, $m = 0, 1, \dots, L$. The solution of this linear least squares problem is given by the famous Wiener-Hopf equation

$$\hat{R}_{yu}(m) = \sum_{n=0}^L g(n) \hat{R}_{uu}(m-n) \quad (10)$$

for $m = 0, 1, \dots, L$, where $\hat{R}_{yu}$ and $\hat{R}_{uu}$ are estimates of, respectively, the cross- and autocorrelation functions $R_{yu}(\tau) = \mathbb{E}\{y(t)u(t-\tau)\}$ and $R_{uu}(\tau) = \mathbb{E}\{u(t)u(t-\tau)\}$

$$\hat{R}_{yu}(m) = \frac{1}{N-L} \sum_{t=L}^{N-1} y(t)u(t-m) \quad (11)$$

$$\hat{R}_{uu}(m-n) = \frac{1}{N-L} \sum_{t=L}^{N-1} u(t-n)u(t-m) \quad (12)$$

(Godfrey 1993, Chapter 1; Ljung 1999; Chapter 6). Since the number of estimated impulse response coefficients L can grow with the amount of data N , the correlation method (10) is classified as being nonparametric. If the input is white noise, then the expected value of $\hat{R}_{uu}(m)$ is proportional to the Kronecker delta $\delta(m)$, and the cross-correlation $\hat{R}_{yu}(m)$ (10) is – within a scaling factor – a good approximation of the impulse response. This property is used in blind channel estimation.

Gaussian Process Regression

The linear least squares (9) solution can be (very) sensitive to disturbing output noise if L is not much smaller than N . This problem is circumvented by the Gaussian process regression approach. The key idea consists in modeling the impulse response coefficients $g(n)$ as a zero-mean Gaussian process with a certain covariance structure P_L that depends on a few hyper-parameters (Pillonetto et al. 2011). In Chen et al. (2012), it has been shown that the Gaussian process regression is equivalent to the following regularized linear least squares problem (see also ▶ “System Identification Techniques: Convexification, Regularization, Relaxation”), by Alessandro Chiuso

$$\sum_{t=L}^{N-1} \left(y(t) - \sum_{n=0}^L g(n)u(t-n) \right)^2 + \sigma^2 g^T P_L^{-1} g \quad (13)$$

where $g = (g(0), g(1), \dots, g(L))^T$ and with σ^2 the variance of the output disturbance. The hyper-parameters defining P_L and the noise variance σ^2 are estimated via an empirical Bayes method.

Nonparametric Frequency-Domain Techniques

The frequency-domain techniques estimate the frequency response function (FRF) using relationship (2) or (4) between the input-output DFT spectra. We start with the simplest approach and gradually increase the complexity of the estimation methods. Note that nonparametric FRF estimation is still a quickly evolving research area, such that the pros and cons of the advanced methods are yet not well established.

Empirical Transfer Function Estimation

If an integer number of periods P of the steady-state response to a *periodic excitation* is observed, then the leakage term in $T_G(\Omega_k)$ in (2) is zero, and the FRF is estimated by dividing the output by the input DFT spectra at the *excited frequencies* (Pintelon and Schoukens

2012, Section 2.4)

$$\hat{G}(\Omega_k) = \frac{Y(k)}{U(k)} \quad (14)$$

The output noise variance $\sigma_V^2(k)$ is estimated via the sample variance $\hat{\sigma}_V^2(k)$ of the output DFT spectra over the P consecutive signal periods. The variance of the FRF estimate (14) is then given by

$$\text{var}(\hat{G}(\Omega_k)) = \frac{\sigma_V^2(k)}{P |U(k)|^2} \quad (15)$$

where $|U(k)|$ is the magnitude of $U(k)$.

Applying (14) to *random excitations* gives the empirical transfer function estimate (Ljung 1999, Section 6.3). Due to the presence of the plant leakage error $T_G(\Omega_k)/U(k)$, the statistical properties of (14) for random inputs are quite different from those for periodic inputs. While the empirical transfer function estimate (ETF) is unbiased and has finite variance (15) for periodic inputs, it is biased and has infinite variance for random inputs (Broersen 2004). To improve the statistical properties of the ETF for random inputs, one can either approximate locally the ETF by a polynomial (Stenman et al. 2000) or perform a weighted average of ETFs over subrecords of the total response (Ljung 1999, Section 6.4). In Heath (2007), it is shown that the optimally (in mean square sense) weighted ETF equals the spectral analysis method.

Spectral Analysis Method

The spectral analysis method is available in any digital spectrum analyzer. It is based on the relationship between the FRF and the cross- and autopower spectra of the input-output signals

$$G(\Omega) = \frac{S_{yu}(\Omega)}{S_{uu}(\Omega)} = \frac{F\{R_{yu}(\tau)\}}{F\{R_{uu}(\tau)\}} \quad (16)$$

with $F\{\}$ the Fourier transform (Bendat and Piersol 1980, Chapter 4; Brillinger 1981, Chapter 8). Comparing (10) and (16), it can be seen that the spectral analysis method is the frequency-domain equivalent of the correlation method (take the

Fourier transform of the expected value of (10)). There are basically two methods for estimating the cross- and autopower spectra in (16) from sampled data: the Blackman and Tukey (1958) and the Welch (1967) procedures.

The Blackman-Tukey procedure (Blackman and Tukey 1958; Ljung 1999, Section 6.4) consists in taking the DFT (3) of the windowed cross- and autocorrelation functions, viz.,

$$\hat{R}_{yu}(\tau) = \frac{1}{N} \sum_{t=\tau}^{N-1} y(t) u(t-\tau) \quad (17)$$

$$\hat{S}_{Ryu}(k) = \frac{1}{\sqrt{N}} \sum_{\tau=0}^{N-1} w(\tau) \hat{R}_{yu}(\tau) e^{-j2\pi \frac{k\tau}{N}} \quad (18)$$

resulting in an FRF estimate (16) at the full frequency resolution $1/(NT_s)$ of the measurement. It can be shown that (18) is a smoothed version of the periodogram $Y(k)\overline{U(k)}$, where is $\bar{U}$ the complex conjugate of U (Brillinger 1981, Chapter 5).

In the Welch approach (Welch 1967; Pintelon and Schoukens 2012, Section 2.6), the N input-output samples are split into M subrecords of N/M samples each, and the DFT spectra $(U^{[m]}(k))_w$ and $(Y^{[m]}(k))_w$ of the windowed input and output samples are calculated via (3) where N is replaced by N/M , giving

$$\hat{S}_{Y_W U_W}(k) = \frac{1}{M} \sum_{m=1}^M (Y^{[m]}(k))_w \overline{(U^{[m]}(k))_w} \quad (19)$$

$$\hat{S}_{U_W U_W}(k) = \frac{1}{M} \sum_{m=1}^M \left| (U^{[m]}(k))_w \right|^2 \quad (20)$$

The spectral analysis estimate of the FRF and its variance are then given by

$$\hat{G}(\Omega_k) = \frac{\hat{S}_{Y_W U_W}(k)}{\hat{S}_{U_W U_W}(k)} \quad (21)$$

$$\text{var}(\hat{G}(\Omega_k)) \approx \frac{\sigma_V^2(k)}{M} \mathbb{E} \left\{ \hat{S}_{U_W U_W}^{-1}(k) \right\} \quad (22)$$

(Brillinger 1981, Chapter 8; Heath 2007). Finally, the output noise variance $\sigma_V^2(k)$ in (22) is estimated as

$$\hat{\sigma}_V^2(k) = \frac{M}{M-1} \left(\hat{S}_{Y_W Y_W}(k) - \frac{|\hat{S}_{Y_W U_W}(k)|^2}{\hat{S}_{U_W U_W}(k)} \right) \quad (23)$$

(Brillinger 1981, Chapter 8; Pintelon and Schoukens 2012, Section 2.5.4). Due to the spectral width of the window used, the estimates (21) and (23) are correlated over the frequency (the correlation length is about twice the spectral width). Note that (20) is used for estimating noise power spectra (Brillinger 1981, Chapter 5). Note also that for *periodic excitations* combined with a rectangular window $w(nT_s) = 1$, the spectral analysis estimate (21), where each subrecord is equal to a signal period, simplifies to the ETFE (14).

Compared with the Blackman-Tukey procedure (18), the FRF estimate (21) based on the Welch approach (19) and (20) has a frequency resolution and a variance (22) that are M times smaller. In measurement devices, the FRFs are estimated using the Welch approach (19)–(21) where each subrecord is an independent measurement with a fixed number of samples. The reason for this is that the cross- and autopower spectra estimates (19) and (20) can easily be updated as more experiments (input-output data records) are available. If the number of measured records M increases to infinity, then (21) converges to the true value, provided a perfect suppression of the leakage error.

In measurement devices, the quality of the spectral analysis estimate (21) is often quantified via the coherence $\gamma^2(\omega)$

$$\gamma^2(\omega) = \frac{|S_{yu}(\Omega)|^2}{S_{yy}(\Omega)S_{uu}(\Omega)} \quad (24)$$

which is comprised between 0 and 1. It is related to the variance of the spectral analysis estimate as

$$\text{var}(\hat{G}(\Omega_k)) = \frac{1 - \gamma^2(\omega_k)^2}{\gamma} (\omega_k) |G(\Omega_k)|^2$$

A coherence smaller than 1 indicates the presence of disturbing noise, residual leakage errors, non-linear distortions, or a nonobserved input.

Following the same lines of Welch (1967), the statistical properties of the spectral analysis estimate (21) can be improved via overlapping subrecords in the cross- and autopower spectra estimates (19) and (20). This has been studied in detail for noise power spectra in Carter and Nuttall (1980) and for FRFs in Antoni and Schoukens (2007).

Advanced Methods

The goal of the advanced methods is to estimate the FRF at the full frequency resolution $1/(NT_s)$ of the experiment duration NT_s while suppressing the influence of the leakage and the noise errors. Without some extra information, it is impossible to achieve this goal via (2). The additional piece of information that allows one to solve the problem is that the FRF and the leakage error are locally smooth functions of the frequency.

The *local polynomial method* (Pintelon and Schoukens 2012, Chapter 7) approximates the FRF and the leakage error in (2) locally in the frequency band $[k - n, k + n]$ by a polynomial. From the residuals of the local linear least squares solution, one also gets an estimate of the output noise variance σ_V^2 and, hence, also of the variance of the FRF. The whole procedure is repeated for all DFT frequencies k in the frequency band of interest. The correlation length of the estimates equals $\pm 2n$, which is twice the local bandwidth of the polynomial approximation.

The *local rational method* (McKelvey and Guérin 2012) follows the same lines as the local polynomial method, except that the FRF and the leakage error in (2) are locally approximated by rational forms with the same poles ($G = B/A$ and $T_G = I/A$). Due to the common poles, the local rational approximation problem can be transformed into a local linear least squares problem. The method is biased but suppresses better the plant leakage error of lowly damped systems.

The *transient impulse response modeling method* (Hägg and Hjalmarsson 2012) approximates the FRF and the leakage error by,

respectively, finite impulse and transient response models, giving a large sparse global linear least squares problem. From the residuals of the global linear least squares solution, one gets an estimate of the output noise variance σ_V^2 and, hence, also of the variance of the FRF. This approach has the best smoothing properties and is recommended in case the noise error is dominant.

Extensions

In sections “Nonparametric Time-Domain Techniques” and “Nonparametric Frequency-Domain Techniques,” it is assumed that the linear plant operates in open loop and that the input is known exactly. If the plant operates in feedback and/or the input observations are noisy, then the presented time and frequency-domain techniques are biased. In sections “Systems Operating in Feedback” and “Noisy Input, Noisy Output Observations,” it is shown that the estimation bias can be avoided if a known external reference signal is available (typically the signal stored in the arbitrary waveform generator).

Since most real-life systems behave to some extent nonlinearly, it is important to detect and quantify the nonlinear effects in FRF estimates. This issue is handled in section “Nonlinear Systems.”

Systems Operating in Feedback

The key difficulty of estimating the FRF of a plant operating in feedback (see Fig. 2) using *nonperiodic excitations* is that the true input $u(t)$ is correlated with the process noise $v(t)$. The direct

approaches of sections “Nonparametric Time-Domain Techniques” and “Nonparametric Frequency-Domain Techniques” lead to biased estimates (Wellstead 1981). This can easily be seen from the ETFE (14) applied to the feedback setup in Fig. 2

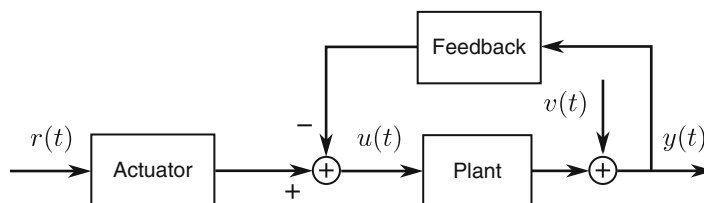
$$\hat{G}(\Omega_k) = \frac{G(\Omega_k)G_{\text{act}}(\Omega_k)R(k) + V(k)}{G_{\text{act}}(\Omega_k)R(k) - G_{\text{fb}}(\Omega_k)V(k)} \quad (25)$$

where $G_{\text{act}}(\Omega_k)$ and $G_{\text{fb}}(\Omega_k)$ are, respectively, the actuator and feedback dynamics. From (25), it follows that in those frequency bands where the process noise $V(k)$ dominates, one rather estimates minus the inverse of the feedback dynamics instead of the plant FRF. On the other hand, at those frequencies where the reference signal injects most power, the ETFE (25) will be close to the plant FRF.

If a known external reference signal is available, then the bias is avoided via the indirect method proposed in Wellstead (1981)

$$G(\Omega) = \frac{S_{yr}(\Omega)/S_{rr}(\Omega)}{S_{ur}(\Omega)/S_{rr}(\Omega)} = \frac{S_{yr}(\Omega)}{S_{ur}(\Omega)}. \quad (26)$$

The basic idea consists in modeling the feedback setup (see Fig. 2) from the known reference to the input and output simultaneously. This reduces the single-input, single-output closed loop problem to a single-input, two-output open loop problem. Since the process noise $v(t)$ is independent of the reference signal $r(t)$, the direct estimate of the single-input, two-output FRF is unbiased. Calculating the ratio of the two FRFs finally gives the indirect estimate (26). This procedure



Nonparametric Techniques in System Identification, Fig. 2 Plant operating in closed loop: $r(t)$ is the external reference signal, the known input $u(t)$ depends on the process noise $v(t)$, and $y(t)$ is the noisy output observation

can be applied to any of the direct methods of sections “Nonparametric Time-Domain Techniques” and “Nonparametric Frequency-Domain Techniques.” Proceeding in this way, unstable plants operating in a stabilizing feedback loop can also be handled.

If the excitation is *periodic*, then the process noise $v(t)$ is independent of the periodic part of the input $u(t)$, and the ETFE (14) converges to the true value as the number of periods P tends to infinity (Pintelon and Schoukens 2012, Section 2.5). Hence, in the periodic case, no external reference is needed.

Noisy Input, Noisy Output Observations

The key difficulty of estimating the FRF of a plant excited by a *nonperiodic signal* from noisy input, noisy output observations (see Fig. 3) is that the input autopower spectrum in (16) is biased. Indeed, due to the noise on the input, $S_{uu}(\Omega)$ is too large, resulting in too small direct FRF estimates. This is true for all direct FRF approaches in sections “Nonparametric Time-Domain Techniques” and “Nonparametric Frequency-Domain Techniques.” Applying the indirect method of section “Systems Operating in Feedback” removes the bias because the noise on the input is independent of the reference signal (e.g., see (26)). Proceeding in this way, the closed loop case (see Fig. 2) with noisy input, noisy output observations is also solved by the indirect method.

If the excitation is *periodic*, then the mean value of the input-output DFT spectra over the P consecutive periods converges to the their

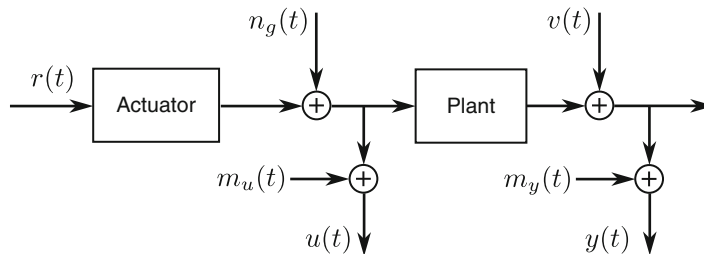
true values as P tends to infinity (Pintelon and Schoukens 2012, Section 2.5). Hence, the ETFE (14) is still consistent, and no external reference is needed. The same conclusion is valid for systems operating in feedback.

Nonlinear Systems

The classes of nonlinear systems considered are those systems whose steady-state response to a periodic input is periodic with the same period as the input. It excludes phenomena such as chaos and subharmonics but allows for hard nonlinearities such as saturation, dead zones, and clipping.

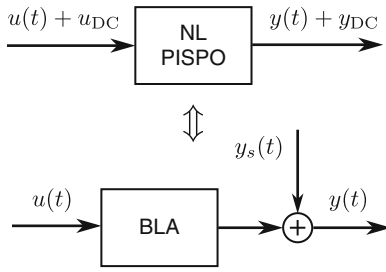
The classes of excitations considered are stationary random signals with a specified power spectrum and probability density function. An important special case is the class of Gaussian excitation signals with a specified power spectrum. This class includes random phase multisines (a sum of harmonically related sinewaves with user-specified amplitudes and random phases) with the same Riemann equivalent power spectrum (Pintelon and Schoukens 2012, Section 4.2).

Consider a nonlinear (NL) period in, same period out (PISPO) system excited by a random excitation belonging to a particular class (see Fig. 4). The FRF (16), where the expected value is taken w.r.t. the random realization of the excitation, is the best (in mean square sense) linear approximation (BLA) of the nonlinear PISPO system, because the difference $y_s(t)$ between the actual output of the nonlinear system (DC value excluded) and the output predicted by the linear approximation is uncorrelated with the



Nonparametric Techniques in System Identification, Fig. 3 Errors-in-variables framework: $r(t)$ is the external reference signal; $n_g(t)$ is the generator noise; $m_u(t)$, $m_y(t)$

are the input and output measurement errors; $v(t)$ is the process noise; and $u(t)$, $y(t)$ are the noisy input, noisy output observations



Nonparametric Techniques in System Identification,

Fig. 4 Best linear approximation (BLA) of a nonlinear (NL) period in, same period out (PISPO) system, excited by a zero-mean random signal $u(t)$ with a given power spectrum and probability density function. $y(t)$ is the zero-mean part of the actual output of the nonlinear system. u_{DC} and y_{DC} are the DC levels of the actual input and output of the nonlinear system. The zero-mean output residual $y_s(t)$ is uncorrelated with – but not independent of – the input $u(t)$

input $u(t)$ (Enqvist and Ljung 2005). Although uncorrelated with the input, the output residual $y_s(t)$ still depends on $u(t)$. If the NL PISPO system operates in feedback (see Fig. 2), then the indirect method (26) is used for calculating the BLA, and the output residual $y_s(t)$ is uncorrelated with – but not independent of – the reference signal $r(t)$ (Fig. 2).

For the class of Gaussian excitation signals, it can be shown that the DFT spectrum $Y_S(k)$ of $y_s(t)$ has the following properties (Pintelon and Schoukens 2012, Section 3.4.4):

1. $Y_S(k)$ has zero-mean value: $\mathbb{E}\{Y_S(k)\} = 0$.
2. $Y_S(k)$ it is uncorrelated with – but not independent of – $U(k)$: $\mathbb{E}\{Y(k)\overline{U(k)}\} = 0$.
3. $Y_S(k)$ is asymptotically ($N \rightarrow \infty$) normally distributed.
4. $Y_S(k)$ is asymptotically ($N \rightarrow \infty$) uncorrelated over the frequency.

These second-order properties are exactly the same as those of a filtered white noise disturbance, except that the noise is independent of the input. It shows that it is impossible to distinguish the nonlinear distortions $y_s(t)$ from the disturbing noise $v(t)$ in FRF measurements using stationary random excitations (only second-order statistics are involved in (21)–(23)).

Using random phase multisines, it is possible to detect and quantify the nonlinear distortions

because $y_s(t)$ is then periodically related to the input $u(t)$ (property of the NL PISPO system). Indeed, analyzing the FRF over consecutive signal periods quantifies the noise variance $v(t)$ ($y_s(t)$ does not change over the periods), while analyzing the FRF over different random phase realizations of the input quantifies the sum of the noise variance and the variance of the nonlinear distortions ($y_s(t)$ depends on the random phase realization of the input). Subtracting both variances gives an estimate of the variance of the nonlinear distortions. While this variance quantifies exactly the variability of the nonparametric FRF estimate due to the nonlinear distortions, it can (significantly) underestimate the variability of a parametric plant model. The basic reason for this is that the true variance of the parametric plant model also depends on the nonzero higher (>2) order moments between the input $u(t)$ and the nonlinear distortions $y_s(t)$.

User Choices

There is no clear answer to the question which of the presented techniques is the best. It strongly depends on the intended use of the nonparametric estimates and the particular application handled. For example, the intended use can be:

1. A smooth representation of the FRF
2. Use of the nonparametric estimates as an intermediate step for parametric modeling of the plant

In the first case, one should opt for the minimum mean square error solution, while in the second case, it is crucial that the nonparametric estimates are unbiased, possibly at the price of an increased variance. Indeed, the parametric plant modeling step cannot eliminate the bias error in the nonparametric estimates while it suppresses the variance error.

The application-dependent answers to the following questions strongly influence the choice and the settings of the method used:

1. Is a large frequency resolution needed and/or is leakage the dominant error?

2. Is the noise or the leakage error dominant?
3. Is it necessary to detect and quantify the nonlinear behavior?

If the answer to the first question is yes, then one should opt for one of the advanced methods (section “Advanced Methods”) or use the spectral analysis estimates (section “Spectral Analysis Method”) with a small number M of subrecords. On the other hand, if the noise error is dominant, then M in (21)–(23) should be chosen as large as possible. To detect and quantify the nonlinear effects, one should use periodic signals (random phase multisines) combined with the ETFE (section “Empirical Transfer Function Estimation”).

Finally, comparing the different nonparametric techniques is also not straightforward because of their different

1. Frequency resolution
2. Quality of the estimated noise model
3. Correlation length over the frequency

The latter is set by the spectral width of the window used in the spectral analysis method and the local bandwidth in the advanced methods.

Guidelines

While the previous sections give well-established facts about the different nonparametric techniques, in this section, we provide some advices/guidelines based on our personal interpretation of these facts:

- Always store the reference signal together with the observed input-output signals. The knowledge of the reference signal allows one to solve nonparametrically the closed loop and errors-in-variables problems.
- Whenever possible use periodic excitation signals (random phase multisines): they allow one to estimate from one experiment the FRF, the noise level, and the level of the nonlinear distortions. As such the deviation of the true dynamic behavior from the ideal linear time-invariant framework is quantified.
- Select one of the advanced methods if frequency resolution is of prime interest.
- If the goal of the identification experiment is to minimize the prediction error, then the Gaussian process regression method is a very promising approach.
- For lowly damped systems and a limited frequency resolution, the local rational method is a good candidate solution.
- Use a minimum mean square solution for a smooth representation of the FRF.
- Choose unbiased nonparametric estimates for use in parametric plant modeling (estimation, validation, and model selection).
- When comparing nonparametric techniques, always take into account all aspects of the estimates: the bias and variance of the FRF and noise model, the frequency resolution, and the correlation length over the frequency.

Summary and Future Directions

Nonparametric techniques are very useful because they simplify the parametric plant modeling in the initial selection of the model complexity and in the detection of unmodeled dynamics. The classical correlation and spectral analysis methods developed in the 1950s and refined till the 1980s are still widely used. Recently, advanced time- and frequency-domain methods have been developed which all try to minimize the sensitivity (bias and variance) of the nonparametric estimates to disturbing noise, nonlinear distortion, and transient (leakage) errors.

The renewed research interest in nonparametric techniques should be continued to handle the following challenging problems: short data sets, missing data, detection and quantification of time-variant behavior, modeling of time-variant dynamics, and modeling of nonlinear dynamics.

Cross-References

- [Frequency Domain System Identification](#)
- [Frequency-Response and Frequency-Domain Models](#)

- [Nonparametric Techniques in System Identification: the Time-Varying and Missing Data Cases](#)
- [System Identification: An Overview](#)
- [System Identification Techniques: Convexification, Regularization, Relaxation](#)

Recommended Reading

The classical correlation (see section “Correlation Methods”) and spectral analysis (see section “Spectral Analysis Method”) methods are well covered by the text books listed below. The recommended reading list includes the basic papers on the spectral analysis methods (Blackman and Tukey 1958; Welch 1967; Wellstead 1981) and the most recent developments described in sections “Gaussian Process Regression” and “Advanced Methods.”

Acknowledgments This work is sponsored by the Research Foundation Flanders (FWO-Vlaanderen), the Flemish Government (Methusalem Fund, METH1), the Belgian Federal Government (Interuniversity Attraction Poles programme IAP VII, DYSCO), and the European Research Council (ERC Advanced Grant SNLSID).

Bibliography

- Antoni J, Schoukens J (2007) A comprehensive study of the bias and variance of frequency-response-function measurements: optimal window selection and overlapping strategies. *Automatica* 43(10):1723–1736
- Bendat JS, Piersol AG (1980) Engineering applications of correlations and spectral analysis. Wiley, New York
- Blackman RB, Tukey JW (1958) The measurement of power spectra from the point of view of communications engineering – Part II. *Bell Syst Tech J* 37(2): 485–569
- Brillinger DR (1981) Time series: data analysis and theory. McGraw-Hill, New York
- Broersen PMT (2004) Mean square error of the empirical transfer function estimator for stochastic input signals. *Automatica* 40(1):95–100
- Carter GC, Nuttall AH (1980) On the weighted overlapped segment averaging method for power spectral estimation. *Proc IEEE* 68(10):1352–1353
- Chen T, Ohlsson H, Ljung L (2012) On the estimation of transfer functions, regularizations and Gaussian processes – revisited. *Automatica* 48(8):1525–1535
- Enqvist M, Ljung L (2005) Linear approximations of nonlinear FIR systems for separable input processes. *Automatica* 41(3):459–473
- Eykhoff P (1974) System identification. Wiley, New York
- Godfrey K (1993) Perturbation signals for system identification. Prentice-Hall, Englewood Cliffs
- Hägg P, Hjalmarsson H (2012) Non-parametric frequency response function estimation using transient impulse response modelling. Paper presented at the 16th IFAC symposium on system identification, Brussels, 11–13 July, pp 43–48
- Heath WP (2007) Choice of weighting for averaged nonparametric transfer function estimates. *IEEE Trans Autom Control* 52(10):1914–1920
- Ljung L (1999) System identification: theory for the user, 2nd edn. Prentice-Hall, Upper Saddle River
- McKelvey T, Guérin G (2012) Non-parametric frequency response estimation using a local rational model. Paper presented at the 16th IFAC symposium on system identification, Brussels, 11–13 July, pp 49–54
- Pillonetto G, Chiuso A, De Nicolao G (2011) Prediction error identification of linear systems: a nonparametric Gaussian regression approach. *Automatica* 47(2):291–305
- Pintelon R, Schoukens J (2012) System identification: a frequency domain approach, 2nd edn. IEEE, Piscataway/Wiley, Hoboken
- Schoukens J, Rolain Y, Pintelon R (2006) Analysis of windowing/leakage effects in frequency response function measurements. *Automatica* 42(1):27–38
- Stenman A, Gustafsson F, Rivera DE, Ljung L, McKelvey T (2000) On adaptive smoothing of empirical transfer function estimates. *Control Eng Pract* 8(11):1309–1315
- Welch PD (1967) The use of the fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Trans Audio Electroacoust* 15(2):70–73
- Wellstead PE (1981) Non-parametric methods of system identification. *Automatica* 17(1):55–69

Nonparametric Techniques in System Identification: The Time-Varying and Missing Data Cases

Rik Pintelon and J. Lataire
Department ELEC, Vrije Universiteit Brussel,
Brussels, Belgium

Abstract

This entry is an extension of the nonparametric techniques presented in “► [Nonparametric Techniques in System Identification](#)”, by Rik Pintelon and Johan Schoukens, to time- and

parameter-varying systems and missing data problems. To increase its readability, we first briefly recall in the introduction some definitions given in “► [Nonparametric Techniques in System Identification](#)”, by Rik Pintelon and Johan Schoukens, and clarify the leakage contribution to nonparametric frequency response function (FRF) estimation. Given its importance in the nonparametric estimation of time- and parameter-varying dynamics, the entry starts by describing the newest developments in advanced techniques for estimating FRFs. Next, the time-varying, parameter-varying, and missing data cases are handled.

As a main result the reader will learn about (i) the detection and quantification of time-varying effects and nonlinear distortions in FRF estimates, (ii) the estimation of time- and parameter-varying FRFs, and (iii) the estimation of (time-varying) FRFs in the presence of missing data. All results are valid for discrete- and continuous-time systems operating in open or closed loop.

Keywords

Best linear time-invariant approximation · Frequency response function · Local rational methods · Parameter-varying systems · Time-varying frequency response function · Missing data

Introduction

Nonparametric estimation typically starts from sampled input-output signals $u(nT_s)$ and $y(nT_s)$, $n=0, 1, \dots, N-1$. In a frequency domain approach, these samples are transformed into the frequency domain via the discrete Fourier transform (DFT)

$$X(k) = \text{DFT}\{x(t)\} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(nT_s) z_k^{-n} \quad (1)$$

with $z_k = e^{j2\pi k/N}$, T_s the sampling period, $x=u$ or y , and $X = U$ or Y . A major issue in FRF and noise power spectra estimation is the leakage

error in the DFT spectrum (1) (see Brillinger 1981). This error is due to the finite duration NT_s of the experiment and increases the mean square error of the estimates. However, it is zero for time-limited signals and periodic signals that are sampled coherently (Pintelon and Schoukens 2012, Sections 2.2.3 and 2.2.4).

For arbitrary excitations $u(t)$, the relationship between the true input $U(k) = \text{DFT}\{u(nT_s)\}$ and true output $Y_0(k) = \text{DFT}\{y_0(nT_s)\}$ discrete Fourier transform spectra (1) of a linear dynamic system is given by

$$Y_0(k) = G(\Omega_k) U(k) + T_G(\Omega_k) \quad (2)$$

where $\Omega_k = j\omega_k$ or $\exp(-j\omega_k T_s)$ for, respectively, continuous- and discrete-time systems; $\omega_k = 2\pi k/(NT_s)$; $G(\Omega_k)$ the plant frequency response function; and $T_G(\Omega)$ a rational function of Ω that decreases to zero as $O(N^{-1/2})$ for N increases to infinity (Pintelon and Schoukens 2012, Section 6.3.2). The numerator coefficients of $T_G(\Omega)$ depend on the difference between the initial and final conditions of the experiment, and $T_G(\Omega)$ has the same denominator as the plant transfer function $G(\Omega)$. Therefore, the time domain response of $T_G(\Omega)$ is decaying exponentially to zero as a transient error. Since $T_G(\Omega_k)$ is zero when the initial and final conditions are equal (e.g., for time-limited input-output signals or coherently sampled steady-state responses to periodic excitations), it is called the leakage error here.

First, the newest developments in advanced nonparametric frequency domain techniques are described (section “[Advanced Nonparametric Frequency Domain Techniques](#)”). The reason for this is that the nonparametric estimation of time-varying dynamics can be reformulated as a multivariate FRF estimation problem (section “[Detection and Quantification of Time-Varying Effects](#)”). Moreover, some of the nonparametric methods for estimating parameter-varying dynamics also rely on advanced frequency domain methods (section “[Parameter-Varying Systems](#)”). Next, section “[Missing Data](#)” handles the estimation of (time-varying) frequency response functions in the presence of

missing data. Further, some additional practical guidelines are given in section “[Guidelines](#)”. Finally, section “[Summary and Future Directions](#)” summarizes the main results and indicates some future research directions.

Advanced Nonparametric Frequency Domain Techniques

Leakage is a major issue in FRF estimation of lowly damped systems. Using the smoothness of the FRF and the leakage error, advanced methods have been developed that suppress better the impact of leakage on the estimates than the standard spectral analysis methods (see “[Nonparametric Techniques in System Identification](#)”, by Rik Pintelon and Johan Schoukens). In this section we briefly discuss the newest developments.

A *first class* of methods approximates the FRF and the leakage error in (2) in a local frequency band $[k - n, k + n]$ by a polynomial (see “[Nonparametric Techniques in System Identification](#)”, by Rik Pintelon and Johan Schoukens). This procedure is then repeated for all frequencies k of interest. The local polynomial approximation works well so long as the frequency resolution of the measurement is large enough (more than seven frequencies within the 3 dB bandwidth of the resonances).

To handle lowly damped systems measured with a low frequency resolution (less than seven frequencies within the 3 dB bandwidth of the resonances), a *second class* of methods has been developed that approximates the FRF and the leakage error locally by a rational function. In McKelvey and Guérin (2012), the local rational approximation problem is transformed into a linear least squares problem via a common denominator parametrization of the FRF and the leakage error ($G = B/A$ and $T_G = I/A$). Although the resulting estimator is biased, it works well for lowly damped systems and sufficiently high signal-to-noise ratios (typically ≥ 40 dB). In Voorhoeve et al. (2018), this method is extended to multivariate frequency response functions that are locally approximated by a

left matrix fraction description. To avoid the bias of the linear least squares approach, the local parameters are obtained by minimizing $\|Y - GU - T_G\|_2^2$ over each local frequency band. This requires a nonlinear optimization, which is circumvented in Peumans et al. (2018) via a weighted total least squares estimator.

Extensions to the Time-Varying and Missing Data Cases

In “[Nonparametric Techniques in System Identification](#)”, by Rik Pintelon and Johan Schoukens, it is assumed that the plant dynamics are time-invariant. However, in quite some (bio-) engineering applications, the dynamics are time- or parameter-varying, for example, thermal drift in power electronics, pit corrosion of metals, muscle fatigue in biomedical experiments, myocardial electrical impedance spectroscopy. The time variation can also be the consequence of a linearization of nonlinear dynamics around a varying set point or a periodic steady-state solution. Section “[Detection and Quantification of Time-Varying Effects](#)” discusses the detection, quantification, and nonparametric modeling of the time-varying effects in FRF estimates, while parameter-varying systems are handled in section “[Parameter-Varying Systems](#)”.

Due to sensor malfunction, for example, saturation, clipping, and outliers, and/or data loss during wireless transmission, FRF estimates are subject to missing data. How to handle missing data in FRF estimation is explained in section “[Missing Data](#)”.

Detection and Quantification of Time-Varying Effects

Under some smoothness assumption on the time variation, the time-varying dynamics of a linear time-varying system are uniquely characterized by the time-varying frequency response function (TV-FRF) $G(j\omega, t)$ introduced in Zadeh (1950a). It is related to the time-varying impulse response $g(t, \tau)$ [$y(t) = \int_0^t g(t, \tau)u(\tau)d\tau$] via the Fourier transform

$$G(j\omega, t) = \mathcal{F}_\tau \{g(t, t - \tau)\} \\ = \int_0^\infty g(t, t - \tau) e^{-j\omega\tau} d\tau \quad (3)$$

where t is the response time and τ the time at which the Dirac impulse is applied to the system. The time-varying FRF (3) has the following properties:

1. The steady-state response $y(t)$ to a sinewave input $u(t) = A \sin(\omega t)$ is given by

$$y(t) = A |G(j\omega, t)| \sin(\omega t + \angle G(j\omega, t)) \quad (4)$$

2. The transient response $y(t)$ to an arbitrary input $u(t)$ is calculated as

$$y(t) = \mathcal{L}^{-1} \{G(s, t)U(s)\} \quad (5)$$

where $U(s)$ is the Laplace transform of $u(t)$ and with $\mathcal{L}^{-1} \{ \}$ the inverse Laplace transform

(see Zadeh 1950a). Note that these properties are natural extensions of those of an FRF.

Assuming that $G(j\omega, t)$ is a smooth function of t , it can be expanded in series as

$$G(j\omega, t) = \sum_{m=0}^{\infty} G_m(j\omega) f_m(t) \\ \approx \sum_{m=0}^{N_b} G_m(j\omega) f_m(t) \quad \text{for } t \in [0, T] \quad (6a)$$

with $f_m(t)$, $m = 0, 1, \dots$ a complete set of basis functions over $[0, T]$ satisfying

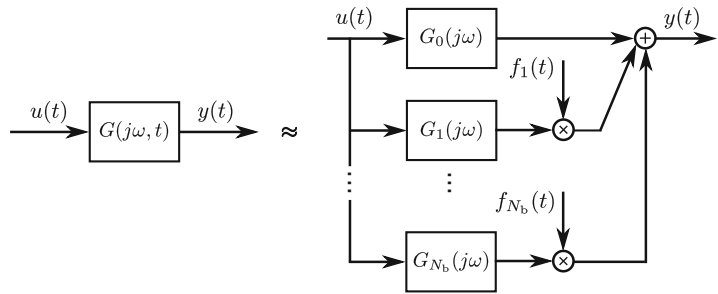
$$f_0(t) = 1 \quad \text{and} \quad \frac{1}{T} \int_0^T f_m(t) dt = 0 \quad \text{for } m \neq 0 \quad (6b)$$

for example, a polynomial basis (arbitrary time variation) or sines and cosines (periodic time variation).

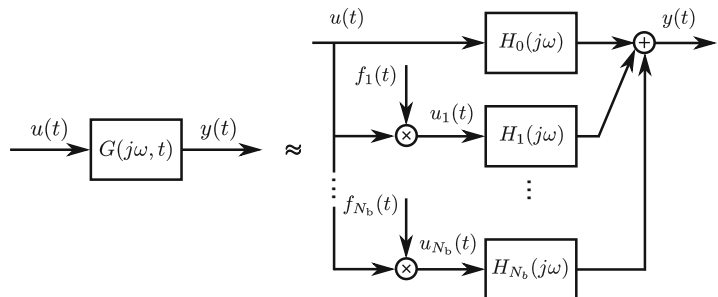
Combining (5) with (6) results in the block diagram shown in Fig. 1. The output gains of this block diagram can be shifted to the input resulting in the indirect model in Fig. 2. There exists an analytic relationship between the FRFs $G_m(j\omega)$ in the direct model and the FRFs $H_m(j\omega)$ in the indirect model (see Pintelon et al. 2015, for polynomial basis functions). For example, for the choice $f_{\pm m}(t) = e^{\pm j m \omega_{\text{sys}} t}$, with ω_{sys} the angular frequency of the periodic time variation, the FRFs

N

Nonparametric Techniques in System Identification: The Time-Varying and Missing Data Cases,
Fig. 1 Direct model (right block diagram) of the time-varying frequency response function $G(j\omega, t)$



Nonparametric Techniques in System Identification: The Time-Varying and Missing Data Cases,
Fig. 2 Indirect model (right block diagram) of the time-varying frequency response function $G(j\omega, t)$



are related as

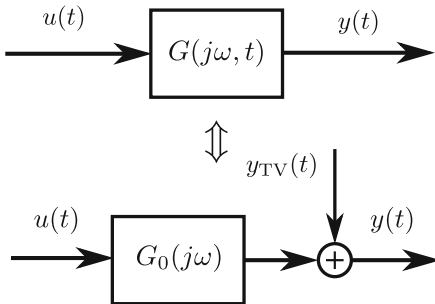
$$G_{\pm m}(j\omega) = H_{\pm m}(j(\omega \pm m\omega_{\text{sys}})) \quad (7)$$

for $m = 0, 1, \dots, N_b$.

Since the inputs $u_m(t)$ of the linear time-invariant blocks $H_m(j\omega)$ in Fig. 2 are linearly independent (the gains $f_m(t)$ form a basis), the indirect model can be viewed as a multiple-input, single-output linear time-invariant system. Hence, the FRFs $H_m(j\omega)$ can be estimated from input-output data using the techniques of section “[Advanced Nonparametric Frequency Domain Techniques](#)” and, next, transformed into $G_m(j\omega)$ (see (7) for complex exponential basis functions and Pintelon et al. (2015) for polynomial basis functions).

The direct model in Fig. 1 can be redrawn as in Fig. 3, where $y_{\text{TV}}(t)$ quantifies the impact of the time-varying branches in the direct model and where $G_0(j\omega)$ can be interpreted as the best linear time-invariant approximation (Lataire et al. 2012). Since $Y_{\text{TV}}(k) = \text{DFT}\{y_{\text{TV}}(t)\}$ has the following properties

1. $Y_{\text{TV}}(k)$ has zero mean value: $\mathbb{E}\{Y_{\text{TV}}(k)\} = 0$.
2. $Y_{\text{TV}}(k)$ is uncorrelated with – but not independent of – $U(k)$: $\mathbb{E}\{Y_{\text{TV}}(k)U(k)\} = 0$.
3. $Y_{\text{TV}}(k)$ is not asymptotically ($N \rightarrow \infty$) normally distributed.



Nonparametric Techniques in System Identification: The Time-Varying and Missing Data Cases, Fig. 3 Best linear time-invariant approximation $G_0(j\omega)$ of a linear time-varying system. The output residual $y_{\text{TV}}(t)$ represents the time-varying effects, and $Y_{\text{TV}}(k) = \text{DFT}\{y_{\text{TV}}(t)\}$ is uncorrelated with – but not independent of – $U(k) = \text{DFT}\{u(t)\}$

4. $Y_{\text{TV}}(k)$ is not asymptotically ($N \rightarrow \infty$) uncorrelated over the frequency.

(see Lataire et al. 2012), it acts as non-Gaussian frequency correlated noise on FRF estimates.

The concept of the best linear time-invariant approximation can be generalized to nonlinear time-varying systems (Ljung 2001). Assuming that the time-invariant branch in the direct model of Fig. 1 is a nonlinear period-in, same period-out system (see “[Nonparametric Techniques in System Identification](#)”, by Rik Pintelon and Johan Schoukens), the $G_0(j\omega)$ block in Fig. 3 is replaced by the best linear approximation of the nonlinear time-invariant block plus an output residual $y_s(t)$ (see “[Nonparametric Techniques in System Identification](#)”, by Rik Pintelon and Johan Schoukens; and Enqvist and Ljung 2005). This results in a best linear time-invariant approximation (BLTIA) of the nonlinear time-varying system with an output residual that consists of two contributions: a nonlinear distortion $y_s(t)$ and a time-varying effect $y_{\text{TV}}(t)$ (see Fig. 4). The DFT spectra of these residuals are uncorrelated

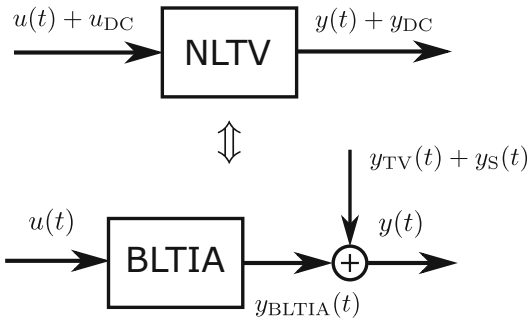
$$\mathbb{E}\{Y_S(k)\overline{Y_{\text{TV}}(k)}\} = 0$$

and using specially designed periodic excitation signals, it is possible to detect and quantify their presence in noisy FRF estimates (Pintelon et al. 2013).

If a known reference signal is available then, following the same lines of “[Nonparametric Techniques in System Identification](#)”, by Rik Pintelon and Johan Schoukens, nonlinear time-varying systems operating in closed loop can be handled (Pintelon et al. 2013).

Parameter-Varying Systems

In quite some applications, the time variation is induced by one or more known physical quantities called the scheduling parameters $p(t)$, for example, the speed and height in airplane dynamics or the varying set point in the local linearization of a nonlinear dynamical system. The connection between parameter- and time-varying systems is that each particular trajectory of the scheduling parameter $p(t)$



Nonparametric Techniques in System Identification: The Time-Varying and Missing Data Cases, Fig. 4

Best linear time-invariant approximation (BLTIA) of a nonlinear time-varying (NLTV) system excited by a zero mean random excitation $u(t)$ with a given power spectral density and probability density function. $y(t)$ is the zero mean part of the actual output of the nonlinear time-varying system. u_{DC} and y_{DC} are the DC levels of the actual input and output of the system. Assuming that the output can be split into a nonlinear time-invariant and a linear time-varying contribution, the difference between the actual output $y(t)$ and the output $y_{BLTIA}(t)$ of the BLTIA can be split into a stochastic nonlinear distortion $y_S(t)$ and a time-varying effect $y_{TV}(t)$. $Y_S(k) = \text{DFT}\{y_S(t)\}$ and $Y_{TV}(k) = \text{DFT}\{y_{TV}(t)\}$ are mutually uncorrelated and uncorrelated with $-$ but not independent of $-U(k) = \text{DFT}\{u(t)\}$

defines a time-varying system. Therefore, the time-varying frequency response function (3) can be generalized to a parameter-varying frequency response function $G_{p(t)}(j\omega, t)$. Linear parameter-varying models are more powerful than linear time-varying models because they are also valid for other trajectories of $p(t)$. As such, linear parameter-varying models can be used for prediction and control purposes (Tóth 2010). They are also a first step toward the modeling and control of nonlinear dynamics (Bamieh and Giarré 2002).

The parameter-varying dynamics can be measured via a local or a global experiment:

- Local experiment: Several experiments are performed for different fixed values of $p(t)$.
- Global experiment: One experiment is performed with a varying $p(t)$.

While the global experiments can handle dynamic dependencies on $p(t)$, the local

experiments are suitable for modeling static dependencies on $p(t)$ or slow (w.r.t. the system bandwidth) parameter variations only.

Local experiments are used in van der Maas et al. (2017) to estimate nonparametrically $G_{p(t)}(j\omega, t)$ via a local polynomial approximation over the frequency and the scheduling parameter(s).

Using (Gaussian) kernel functions (Pillonetto et al. 2011) and one global experiment, parameter-varying state space and Box-Jenkins models are estimated nonparametrically in Rizvi et al. (2018) and Darwish et al. (2018).

Missing Data

Assuming that M_y consecutive output samples are missing starting at position K_y , relationship (2) is modified as

$$Y_0^m(k) = G(\Omega_k)U(k) + T_G(\Omega_k) - \frac{1}{\sqrt{N}} z_k^{-K_y} \sum_{n=K_y}^{K_y+M_y-1} y_0(nT_s) z_k^{-n} \quad (8)$$

with $Y_0^m(k)$ the output DFT spectrum (1) where the missing samples are set to zero and with $y_0(nT_s)$, $n = K_y, \dots, K_y + M_y - 1$, the missing output samples. Approximating the FRF $G(\Omega_k)$ for each value of k in a local frequency band around k by a polynomial (see “► Nonparametric Techniques in System Identification”, by Rik Pintelon and Johan Schoukens) transforms (8) into a sparse overdetermined set of equations that can be solved time and memory efficiently using QR-decompositions (Ugryumova et al. 2015).

Note that (8) can easily be extended to the case where (blocks of) samples are missing at non-consecutive time instants. As such, any missing data pattern (e.g., bursts, random, clipping) can be handled so long as missing samples do not occur at the borders of the record. The reason for the latter is that data missing at the borders cannot be distinguished from $T_G(\Omega_k)$ in (8).

If a known reference signal is available then, following the same lines of “► Nonparametric Techniques in System Identification”, by Rik Pintelon and Johan Schoukens, missing input data

and systems operating in closed loop can be handled (Ugryumova et al. 2015).

Since the estimation of the time-varying FRF can be reformulated as the estimation of a multiple-input, single-output FRF, the missing data algorithm developed in Ugryumova et al. (2015) for time-invariant systems is also applicable to the time-varying case (Pintelon et al. 2017).

Guidelines

In addition to the guidelines of “► [Nonparametric Techniques in System Identification](#)”, by Rik Pintelon and Johan Schoukens:

- If lowly damped systems are measured with a limited frequency resolution (less than seven frequencies within the 3 db bandwidth of the resonance), then the local rational methods of section “[Advanced Nonparametric Frequency Domain Techniques](#)” for estimating the FRF are preferred over the local polynomial methods.
- Use periodic excitation signals (random phase multisines), whenever possible: they allow one to estimate from one experiment the FRF, the noise level, and the level of the nonlinear distortions and time-varying effects. Proceeding in this way, the deviation of the true dynamic behavior from the linear time-invariant framework is detected and quantified.
- If data is missing, then the best option is to fix the hardware problem and to redo the experiment(s). If this is not possible because of time and/or cost or because it concerns historical data sets, then the missing data techniques of section “[Missing Data](#)” are very helpful.

Summary and Future Directions

Leakage suppression in frequency response function estimates of lowly damped systems remains a challenging problem. Therefore, there

is a continuing interest to improve the existing advanced methods. Currently, the problem of handling a large number of inputs and outputs is addressed (Peumans et al. 2019).

Using a special class of periodic excitation signals, the recently developed nonparametric techniques allow one to detect and quantify the disturbing noise, the nonlinear distortion, and the time-varying effects in frequency response function estimates. As such, via a simple experiment, the (in)validity of the linear time-invariant framework can easily be established before the parametric modeling step. It answers the following important questions: “Is a linear time-invariant model accurate enough for my application?”, “Can I neglect or do I need to model the nonlinear behavior?”, and “Is time variation of the dynamics an issue?”. Note that all of this is possible in the presence of missing data.

Although started in the early 1950s by the work of Zadeh (1950a, b), a renewed interest in modeling time- and parameter-varying dynamics is observed in the literature. One of the reasons for this is that time-varying-models are a first step toward parameter-varying modeling, while the latter is on its turn a first step toward nonlinear modeling.

The ongoing research interest in nonparametric techniques should be continued to handle the following challenging problems: short data sets and fast time and parameter variations. Although some interesting initial results are available, more effort is needed on nonparametric modeling of parameter-varying (see section “[Parameter-Varying Systems](#)”) and nonlinear (see Birpoutsoukis et al. 2017) dynamical systems.

Cross-References

- [Frequency Domain System Identification](#)
- [Frequency-Response and Frequency-Domain Models](#)
- [Nonparametric Techniques in System Identification](#)
- [System Identification: An Overview](#)
- [System Identification Techniques: Convexification, Regularization, Relaxation](#)

Recommended Reading

The theory of time-varying systems is well covered by the early work of Zadeh (1950a,b) and the book of Rugh (1996). Some basics about parameter-varying systems can be found in the book of Tóth (2010). In addition, the recommended reading list includes the most recent developments described in sections “Advanced Nonparametric Frequency Domain Techniques”, “Extensions to the Time-Varying and Missing Data Cases”, and “Summary and Future Directions”.

Acknowledgments This work is sponsored in part by the Research Foundation Flanders (FWO-Vlaanderen) and in part by the Flemisch Government (Methusalem Fund, METH1).

Bibliography

- Bamieh B, Giarré L (2002) Identification of linear parameter varying models. *Int J Robust Nonlinear Control* 12(9):841–853
- Birpoutsoukis G, Marconato A, Lataire J, Schoukens J (2017) Regularized nonparametric Volterra kernel estimation. *Automatica* 82:324–327
- Brillinger DR (1981) *Time series: data analysis and theory*. McGraw-Hill, New York
- Darwish MAH, Cox PB, Proimadis I, Pillonetto G, Toth R (2018) Prediction-error identification of LPV systems: a nonparametric Gaussian regression approach. *Automatica* 97:92–103
- Enqvist M, Ljung L (2005) Linear approximations of nonlinear FIR systems for separable input processes. *Automatica* 41(3):459–473
- Lataire J, Louarroudi E, Pintelon R (2012) Detecting a time-varying behavior in frequency response function measurements. *IEEE Trans Instrum Meas* 61(8):2132–2143
- Ljung L (2001) Estimating linear time-invariant models of nonlinear time-varying systems. *Eur J Control* 7(2–3):203–219
- McKelvey T, Guérin G (2012) Non-parametric frequency response estimation using a local rational model. *IFAC Proc Vol* 45(16):49–54. Brussels
- Pillonetto G, Chiuso A, De Nicolao G (2011) Prediction error identification of linear systems: a nonparametric Gaussian regression approach. *Automatica* 47(2):291–305
- Peumans D, Busschots C, Vandersteen G, Pintelon R (2018) Improved FRF measurements of lightly damped systems using local rational models. *IEEE Trans Instrum Meas* 67(7):1749–1759
- Peumans D, De Vestel A, Busschots C, Rolain Y, Pintelon R, Vandersteen G (2019) Accurate estimation of the non-parametric FRF of lightly-damped mechanical systems using arbitrary excitations. *Mech Syst Signal Process* 130(1):545–564
- Pintelon R, Schoukens J (2012) *System identification: a frequency domain approach*, 2nd edn. Wiley/IEEE Press, Hoboken
- Pintelon R, Louarroudi E, Lataire J (2013) Detecting and quantifying the nonlinear and time-variant effects in FRF measurements using periodic excitations. *IEEE Trans Instrum Meas* 62(12):3361–3373
- Pintelon R, Louarroudi E, Lataire J (2015) Nonparametric time-variant frequency response function estimates using arbitrary excitations. *Automatica* 51(1):308–317
- Pintelon R, Ugrumova D, Vandersteen G, Louarroudi E, Lataire J (2017) Time-variant frequency response function measurement in the presence of missing data. *IEEE Trans Instrum Meas* 66(11):3091–3099
- Rizvi SZ, Velni JM, Abbasi F, Toth R, Meskin N. (2018) State-space LPV model identification using kernelized machine learning. *Automatica* 88:38–47
- Rugh W.J. (1996) *Linear system theory*, 2nd edn. Prentice Hall, Upper Saddle River.
- Tóth R (2010) *Modeling and identification of linear parameter-varying systems*. Springer, Berlin
- Ugrumova D, Pintelon R, Vandersteen G (2015) Frequency response matrix estimation from missing input-output data. *IEEE Trans Instrum Meas* 64(11):3124–3136
- van der Maas R, van der Maas A, Oomen T (2017) Accurate FRF identification of LPV systems: nD-LPM with application to a medical X-ray system. *IEEE Trans Contr Syst Techn* 25(5):1724–1735
- Voorhoeve R, van der Maas A, Oomen T (2018) Non-parametric identification of multivariable systems: a local rational modeling approach with application to a vibration isolation benchmark. *Mech Syst Signal Process* 105:120–152
- Zadeh LA (1950a) Frequency analysis of variable networks. *Proc I.R.E.* 38:291–299
- Zadeh LA (1950b) The determination of the impulsive response of variable networks. *J Appl Phys* 21:642–645

Non-raster Methods in Scanning Probe Microscopy

Sean B. Andersson
Mechanical Engineering and Division of
Systems Engineering, Boston University,
Boston, MA, USA

Abstract

Scanning probe microscopy (SPM) refers to a family of technologies for probing systems with nanometer-scale features in which a

probe interacts with a sample. Traditionally, images of a signal of interest are built pixel-by-pixel by rastering the probe across the sample. While simple, this method is at least partially responsible for the slow imaging times which are inherent to SPM imaging. Non-raster methods seek to improve image acquisition time by modifying this scanning process to one that is more efficient. This efficiency can be with respect to the probe trajectories, moving to patterns that are easier for scanners to follow, or it can be with respect to scanning area, increasing speed by reducing the amount of scanning to be done.

Keywords

High-speed SPM · Non-raster scanning · Feature-based imaging

Introduction

Scanning probe microscopy (SPM) refers to a broad class of technologies aimed at studying samples with nanometer-scale features. The essential idea is to measure some physical property of the sample, such as topography, material stiffness, conductance, temperature, and many others, through the local interaction between the sample and a probe. An image of this property is formed by rastering the probe across the sample, building the result pixel-by-pixel. There are a variety of names under the SPM umbrella depending on the type of interaction and physical property being measured, including scanning tunneling microscopy (STM), which measures the tunneling current between a sharp tip and the sample surface; atomic force microscopy (AFM) (also known as scanning force microscopy), which measures the interaction force between a sharp tip on the end of a cantilever and the sample; near-field scanning optical microscopy (NSOM), which uses the near-field of a light source to probe the sample surface; and many others.

The probe is typically actuated relative to the sample using a piezoelectric scanning stage and

the signal of interest (such as the tunneling current or the cantilever deflection) is held constant by using a feedback loop to actuate the vertical position of the probe relative to the surface. Due to a combination of the mechanical limitations of the scanning process (such as the bandwidths of the actuators and induced vibrations of the stages) and the fact that images are built pixel by pixel, the imaging process can be very slow. For example, for many AFMs the imaging time is on the order of seconds to minutes. While this is not a major limitation for measuring static signals, it is a severe drawback when studying dynamics in systems with nanometer-scale features or in applications where throughput is a significant concern.

Improving the imaging time of SPM can have a major impact on a broad array of applications, from understanding dynamics in biomolecular systems to high-throughput testing of novel engineered materials. As a result, there is a significant body of literature on overcoming this challenge. Approaches can be broadly categorized into three bins: faster hardware, faster controllers, and alternative (non-raster) scanning. As SPM devices are mechanical microscopes, faster hardware means reducing the physical time constants of different elements in the systems, from the piezoelectric scanners (Yong and Leang 2016) to the probe used to interact with the sample (Walters et al. 1998). The second category, faster controllers, seeks to do more with the existing hardware, applying approaches like adaptive proportional-integral-derivative (Ando 2012), robust control (Helfrich et al. 2009), and feedforward control (Clayton et al. 2009). Combining techniques from these two categories has led to instruments that are orders of magnitude faster than older generations, opening up new avenues of research (Ando et al. 2013; Heron et al. 2014; Yoshioka et al. 2018).

The third category of methods for high-speed AFM, namely non-raster scanning, is the focus of this encyclopedia entry. Non-raster scanning methods, which build an image using alternatives to the simple raster scan, can be loosely organized into three groups: full scans using patterns with reduced frequency content when compared to

the raster pattern, sub-sampling combined with image reconstruction, and local, feature-based methods. The first of these seeks to improve the imaging rate by moving the probe faster than in the raster-scan pattern, while the second two both aim to reduce the imaging time by reducing the number of pixels that are measured. Non-raster methods are complementary to the hardware and control schemes and thus can be combined with either or both for the most impact.

The goal of this entry is to introduce the interested reader to the general ideas in non-raster scanning and provide at least one view on the direction the field is moving. While this is not a survey paper, a few select references are provided throughout to point the reader in hopefully fruitful directions into the literature.

Scanning Patterns with Reduced Frequency Content

In the basic raster scan pattern, shown in Fig. 1a, the probe is scanned using a triangle wave in one direction (the *fast scan* direction) and a ramp in the other (the *slow scan* direction). Note that, as shown in the figure, the pattern makes two passes over a single row of pixels, one in each direction. Due to details of the probe-sample interaction, the measurements differ somewhat in the two directions and thus each scan yields *two* images: a trace scan and a retrace scan. Given a specific frame rate, then, the probe actually moves twice as fast as might be expected.

The primary challenge of the raster-scan pattern is the fact that accurate tracking of the triangle wave requires high bandwidth nanopositioners. For example, to scan an area of $5\ \mu\text{m} \times 5\ \mu\text{m}$ with a resolution of 5 nm in a total time of 1 s, the fast axis must track a 1 kHz triangle wave. To ensure that at least the first five harmonics are covered, the actuator bandwidth must be at least 10 kHz. At lower bandwidths, the output trajectory lags the input and rounds off the corners. This is illustrated in Fig. 1d which shows the reference triangle wave (blue) and the output using an actuator with a (closed-loop) bandwidth sufficient to capture the first 10 harmonics (black)

and with a bandwidth only sufficient for the first five (red). This finite bandwidth eventually limits the frequency of triangle wave which can be reasonably tracked and thus the frame rate that can be achieved.

Clearly one approach to overcoming this limitation is to increase the bandwidth of the scanning stage; this is the goal of the first two categories noted in section “[Introduction](#)” through either hardware or control design. An alternative is to replace the raster scan pattern with one that has lower frequency content than the triangle wave and which thereby allows faster scan patterns without changes in the hardware or controller.

While there are a variety of curves that can be used, there are two primary ones that have been applied: spiral scans and Lissajous patterns. Under a spiral scan, the probe is driven along an Archimedean spiral, shown in Fig. 1b and defined by

$$x(t) = r(t) \cos(\omega t),$$

$$y(t) = r(t) \sin(\omega t),$$

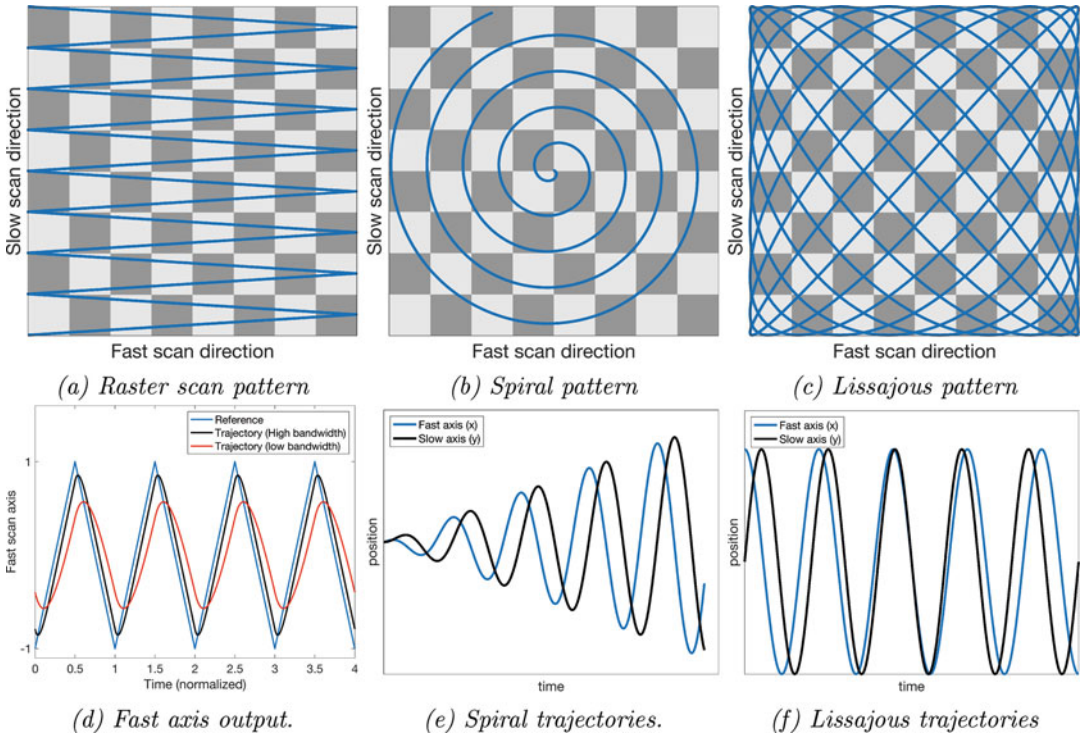
$$r = \frac{P}{2\pi} \omega t,$$

where P is the pitch of the spiral (defining the resolution of the scan) and ω is the frequency of rotation. In particular, there is a single frequency of growing amplitude in each axis, as illustrated in Fig. 1e and as a result the scanning stage can be driven at a high frequency without exciting mechanical resonances of the scanner.

By construction, spiral scan patterns cover a circular imaging area. An alternative family of scan patterns are those defined by Lissajous curves where the two axes are driven according to

$$x(t) = \frac{A}{2} \sin(2\pi f_x t), \quad y(t) = \frac{B}{2} \sin(2\pi f_y t),$$

where A determines the range of the scan while the frequencies f_x and f_y define both the shape and total imaging time of the scan. While these trajectories are quite easy to implement, they do yield a variable resolution across the image.



Non-raster Methods in Scanning Probe Microscopy, Fig. 1 Different scan trajectories for full-frame scanning. (a) Standard raster scan and (d) fast axis tracking. Slower actuators (or faster scans) clip more of the high-frequency content of the triangle wave, leading to poor tracking. (b)

Spiral scans steer the probe along an Archimedean spiral using (e) a single frequency with time-varying amplitude. (c) Lissajous scans scan the image area by (f) driving the two actuators using two different frequencies

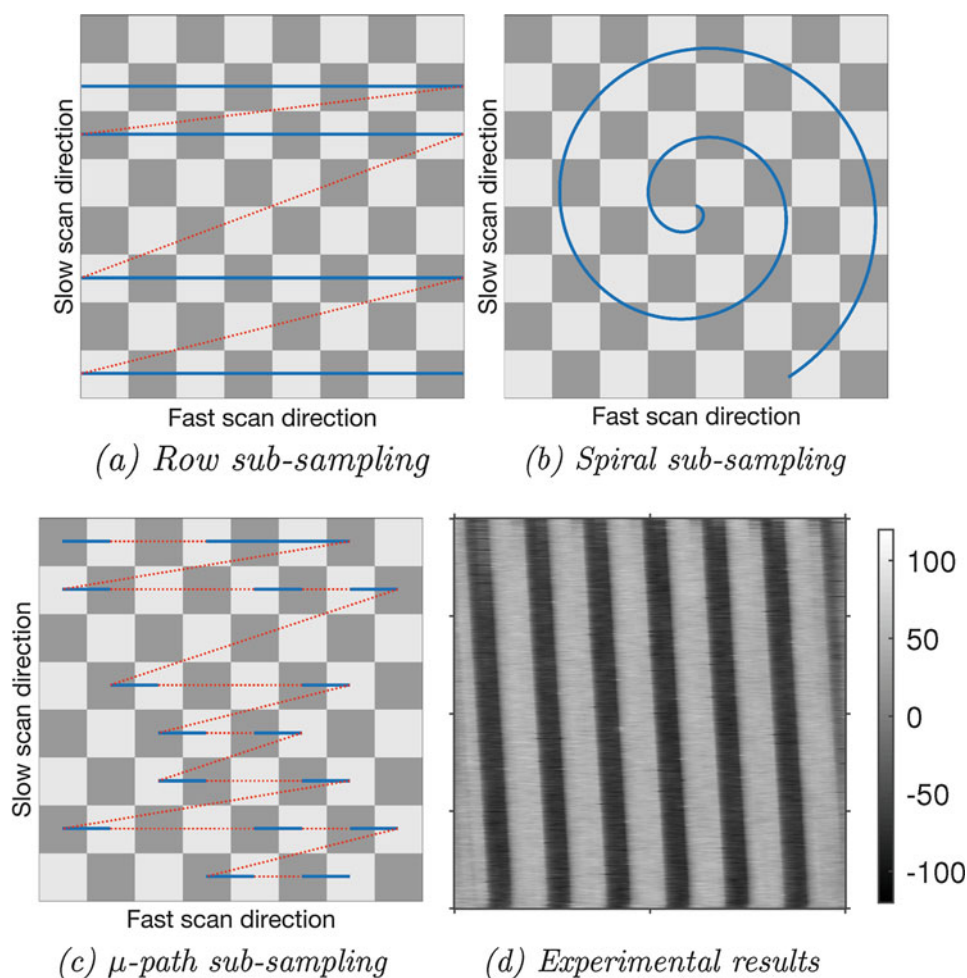
Both of these patterns have been implemented by multiple groups and combined with both novel actuators and advanced controllers to yield frame rates significantly above 1 Hz (Tuma et al. 2012; Rana et al. 2014; Bazaei et al. 2017). Because the patterns sample all the pixels, the images are completely equivalent to those created by a raster scan (with the exception that a spiral scan image area is circular rather than rectangular).

Sub-sampling the Image

The scanning patterns discussed in section “[Scanning Patterns with Reduced Frequency Content](#)” only address limitations due to the bandwidth of the scanning actuators. As the speed of the probe increases, features are encountered more rapidly and thus the bandwidth of the vertical

axis must often also be increased to keep pace with the faster scanning (Teo et al. 2016). The second group of non-raster methods seeks to improve imaging rate by sampling less rather than scanning faster. That is, the entire image is sub-sampled and the final result generated from this sub-sampled data using image reconstruction algorithms.

A variety of sampling patterns can be applied, including simple row-subsampling (Fig. 2a), wide spirals (Fig. 2b), or randomly selected, short horizontal paths known as μ -path patterns (Fig. 2c) (Braker et al. 2018). In some patterns, the probe is repeatedly engaged to the sample, scanned, and then disengaged to move to a new measurement location. This can help with increasing the randomness in the sampling pattern but also slows the overall imaging rate as the engagement process can take a significant



Non-raster Methods in Scanning Probe Microscopy, Fig. 2 Example sub-sampling patterns. Blue, solid lines indicate where measurements are acquired while red, dotted lines indicate where the probe is simply moved to the next sampling position. (a) Row sub-sampling scans randomly selected rows. (b) Spiral sub-sampling steers

the probe along a wide spiral. (c) μ -path sub-sampling moves the probe along short, randomly selected horizontal paths. (d) Experimental result using a μ -path scan of a linear grating based on 15% sampling. Image reproduced with permission from (Braker et al. 2018). Tick marks are 10 μm and the colorbar units are nm

amount of time. As noted below, however, the randomness has a positive impact on the quality of the final image and thus can be worth the time penalty.

Once the data has been collected, a complete image must be generated from the sub-sampled data. Once again there are a variety of methods for achieving this with the two main groups of algorithms being inpainting and sparse reconstruction (Chen et al. 2012; Luo and Andersson 2015). Inpainting schemes seek to fill in missing data using information from nearby pixels. It is

important to note that the methods do not simply average neighboring data (as would be done in interpolation) but rather attempt to continue image features into the unknown regions. The methods work well for capturing low frequency content but can miss high frequency details.

Sparse reconstruction leverages the fact that most natural images are approximately sparse when expressed in an appropriate basis (such as a discrete cosine or wavelet representation), meaning that most of the image information is captured by a small number of coefficients, often

as low as 1%. These algorithms produce the final image through an optimization problem of the form

$$\min \|\eta\|_1 \text{ subject to } \|y - \Phi \Psi \eta\|_2 \leq \epsilon$$

where Φ is the measurement matrix describing which pixels of the image are sampled, Ψ is the selected basis, $\eta = \Psi^{-1}x$ is the representation of x in that basis, y is the vector of measurements, $\|\cdot\|_{1,2}$ is the l_1 or l_2 norm, and ϵ is a real number allowing for some noise in the measurements. Randomness in the measurements leads to a good value of a property known as mutual coherence and from there a better bound on the error of the reconstruction. Sparse reconstruction is a very rich field with many algorithms and a deep literature (see, e.g. (Zhang et al. 2015b)). These algorithms can produce high-quality reconstructions provided the assumption of (approximate) sparsity is met.

Feature-Tracking

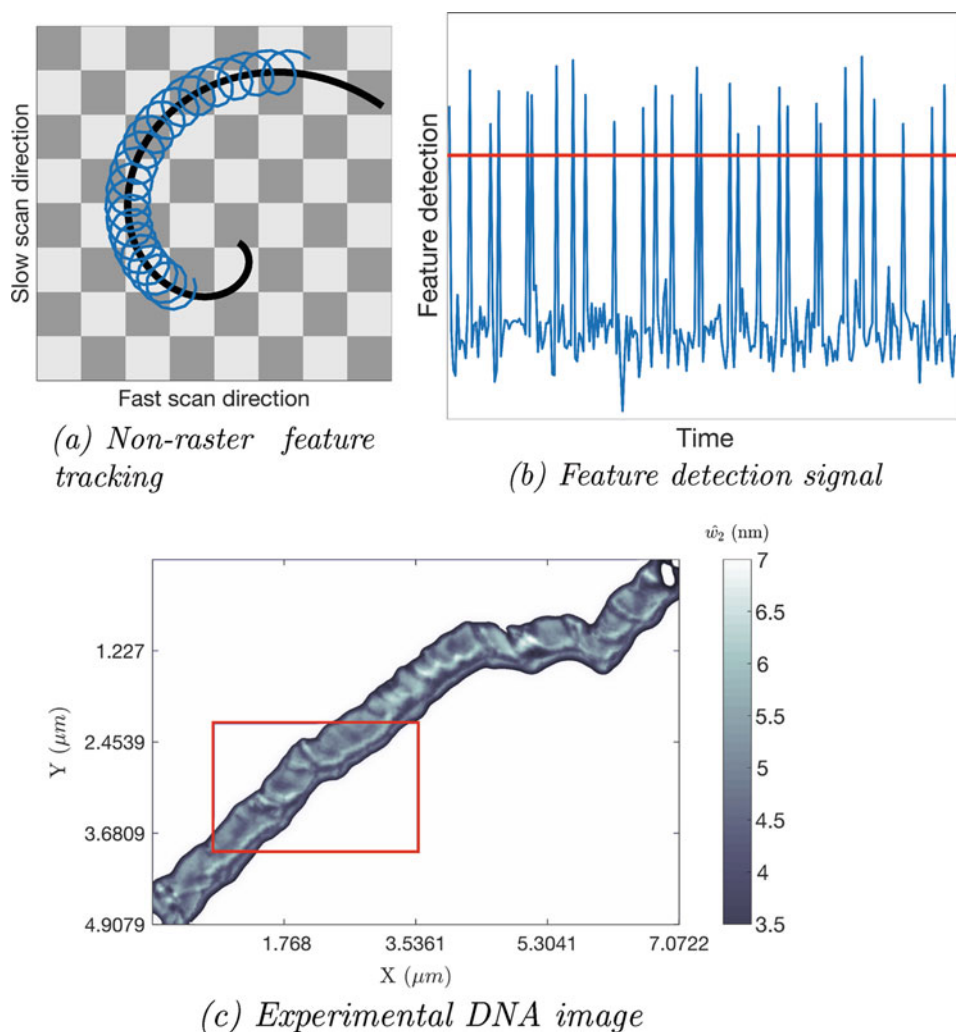
Both of the previous two groups of sub-sampling methods aim to produce entire images from sub-sampled data. In many cases, however, such as when interested in dynamics occurring on a biopolymer or the growth of a crystal boundary, the area of interest is a small fraction of the overall image. This final group of approaches is similar to sub-sampling in that the improvement in imaging rate is due to a reduction of sampling rather than an increase in the speed of the probe relative to the sample. Unlike the sub-sampling approaches, however, this group aims to focus the measurements in the region of interest, using feedback to track a sample feature.

The basic idea is to steer the probe along a local dither pattern, such as a sinusoid or circle, detect in real time the presence of the feature in that local scan, and update the pattern to track the spatial evolution of the feature. As an example, consider imaging a thin but long sample such as a strand of DNA or a length of actin. As illustrated in Fig. 3a, the location of the feature of interest can be determined from the data and the future

evolution of that feature can be predicted. The probe is then moved along that predicted path and feedback used to correct any prediction error. The method has been used to image DNA (Hartman and Andersson 2018) and to quickly build contour maps of cells (Zhang et al. 2015a). Because it naturally eliminates drift in the microscope over time, the method will likely be particularly beneficial to examine samples over long periods of time. These algorithms can yield over an order-of-magnitude improvement in imaging rate for samples to which they apply, without any need to increase probe speed or improve the system hardware.

Summary and Future Directions

Improvements in the hardware and control algorithms have been making their way into industrial instruments, leading to the current generation of high-speed instruments, particularly in atomic force microscopy, with frame rates from 1 to 10 frames/second. Non-raster methods, on the other hand, remain primarily in the labs of individual researchers and have not yet made it into common usage. Despite this, they offer significant potential for drastic improvement in imaging rate and can be applied even to the latest instruments to further improve their imaging speed. Realizing this potential, however, relies on overcoming a variety of challenges. For both reduced frequency content and feature-based tracking patterns, vertical features arrive more rapidly than in standard raster scanning and thus the bottleneck shifts to the vertical bandwidth. These techniques will therefore have the most impact when combined with next-generation vertical positioners and control algorithms. Sub-sampling schemes by design do not sample the entire imaging area, inferring information from the acquired data. Open questions remain about how to ensure the information at a given resolution level has been completely acquired. Adaptive schemes aimed at either refining measurements over time or adjusting measurements in response to dynamic changes



Non-raster Methods in Scanning Probe Microscopy,

Fig. 3 Illustrating feature-tracking based imaging. (a) The path of the probe (blue, circular curve) is determined in real time using feedback from the measurements to follow the spatial evolution of the sample (black curve). (b) The location of the feature is determined from a signal available in real time such as the amplitude of the cantilever motion in tapping mode AFM. As the probe

moves from surface to sample, there is a momentary spike in the amplitude signal before the controller can regulate its value back to the setpoint. Looking for values that exceed a threshold (red line) indicate the location of the sample edge (simulated data). (c) Experimental image of a strand of λ -DNA acquired using a feature-tracking method known as local circular scan. Image reproduced with permission from (Hartman and Andersson 2018)

in the sample are likely to prove particularly beneficial.

Cross-References

- [Scanning Probe Microscope Imaging Control](#)
- [Sensor Drift Rejection in X-Ray Microscopy: A Robust Optimal Control Approach](#)

Bibliography

- Anderson CM, Georgiou GN, Morrison IG, Stevenson GVW, Cherry RJ (1992) Tracking of cell surface receptors by fluorescence digital imaging microscopy using a charge-coupled device camera. *J Cell Sci* 101(2):415–425
- Ando T (2012) High-speed atomic force microscopy coming of age. *Nanotechnology* 23(6):062001

- Ando T, Uchihashi T, Kodera N (2013) High-speed AFM and applications to biomolecular systems. *Ann Rev Biophys* 42(1):393–414
- Bazaei A, Yong YK, Moheimani SOR (2017) Combining spiral scanning and internal model control for sequential AFM imaging at video rate. *IEEE/ASME Trans Mechatron* 22(1):371–380
- Braker RA, Luo Y, Pao LY, Andersson SB (2018) Hardware demonstration of atomic force microscopy imaging via compressive sensing and ℓ_1/ℓ_2 -Path scans. In: American Control Conference (ACC), IEEE, pp 6037–6042
- Chen A, Bertozzi AL, Ashby PD, Getreuer P, Lou Y (2012) Enhancement and recovery in atomic force microscopy images. In: *Excursions in Harmonic Analysis*, vol 2, Birkhäuser, Boston, pp 311–332
- Clayton GM, Tien S, Leang KK, Zou Q, Devasia S (2009) A review of feedforward control approaches in nanopositioning for high-speed SPM. *J Dyn Syst Measur Control* 131(6):061101
- Hartman B, Andersson SB (2018) Feature tracking for high speed AFM imaging of biopolymers. *Int J Mol Sci* 19(4):1044
- Helfrich BE, Lee C, Bristow DA, Xiao XH, Dong J, Alleyne AG, Salapaka SM, Ferreira PM (2009) Combined H_∞ -feedback control and iterative learning control design with application to nanopositioning systems. *IEEE Trans Control Syst Technol* 18(2):336–351
- Heron JT, Bosse JL, He Q, Gao Y, Trassin M, Ye L, Clarkson JD, Wang C, Liu J, Salahuddin S, Ralph DC, Schlom DG, Íñiguez J, Huey BD, Ramesh R (2014) Deterministic switching of ferromagnetism at room temperature using an electric field. *Nature* 516(7531):370–373
- Luo Y, Andersson SB (2015) A comparison of reconstruction methods for undersampled atomic force microscopy images. *Nanotechnology* 26(50):505703
- Rana MS, Pota HR, Petersen IR (2014) Spiral scanning with improved control for faster imaging of AFM. *IEEE Trans Nanotechnology* 13(3):541–550
- Teo YR, Yong Y, Fleming AJ (2016) A comparison of scanning methods and the vertical control implications for scanning probe microscopy. *Asian J Control* 28(2):65
- Tuma T, Lygeros J, Kartik V, Sebastian A, Pantazi A (2012) High-speed multiresolution scanning probe microscopy based on Lissajous scan trajectories. *Nanotechnology* 23(18):185501
- Walters DA, Cleveland JP, Thomson NH, Hansma PK, Wendman MA, Gurley G, Elings V (1998) Short cantilevers for atomic force microscopy. *Rev Sci Instrum* 67(10):3583–3590
- Yong YK, Leang KK (2016) Mechanical design of high-speed nanopositioning systems. In: *Nanopositioning technologies*. Springer, Cham, pp 61–121
- Yoshioka T, Matsushima H, Ueda M (2018) In situ observation of Cu electrodeposition and dissolution on Au(100) by high-speed atomic force microscopy. *Electrochem Commun* 92:29–32
- Zhang K, Hatano T, Tien T, Herrmann G, Edwards C, Burgess SC, Miles M (2015a) An adaptive non-raster scanning method in atomic force microscopy for simple sample shapes. *Meas Sci Technol* 26(3):035401
- Zhang Z, Xu Y, Yang J, Li X, Zhang D (2015b) A survey of sparse representation: algorithms and Applications. *IEEE Access* 3:490–530

Numerical Methods for Continuous-Time Stochastic Control Problems

George Yin

Department of Mathematics, Wayne State University, Detroit, MI, USA

Abstract

This expository article provides a brief review of numerical methods for stochastic control in continuous time. It concentrates on the methods of Markov chain approximation for controlled diffusions. Leaving most of the technical details out with the broad general audience in mind, it aims to serve as an introductory reference or a user's guide for researchers, practitioners, and students who wish to know something about numerical methods for stochastic control.

Keywords

Markov chain approximation · Numerical methods · Stochastic control

Introduction

This expository article provides a brief review of numerical methods for stochastic control in continuous time. Leaving most of the technical details out with the broad general audience in mind, it aims to serve as an introductory reference for researchers, practitioners, and students, who wish to know something about numerical methods for stochastic controls.

The study of stochastic control has witnessed tremendous progress in the last few decades; see,

for example, Fleming and Rishel (1975), Fleming and Soner (1992), Kushner (1977), and Yong and Zhou (1999) among others, for fundamentals of stochastic controls as well as historical remarks. Much of the development has been accompanied by the needs and progress in science, engineering, as well as finance. Typically, the problems are highly nonlinear, so a closed-form solution is very difficult to obtain. As a result, designing feasible numerical algorithms becomes vitally important. Among the many approximation methods, the Markov chain approximation methods have shown most promising features. Primarily for treating diffusions, the Markov chain approximation method was initiated in the 1970s (Kushner 1977) and substantially developed further in Kushner (1990b) and Kushner and Dupuis (1992). Nowadays, such method are used for more complex jump diffusions, or systems with random switchings. There were also efforts to incorporate the methods into an expert system so that the methods can be placed into an easily usable tool box (Chancelier et al. 1986, 1987). In addition to the existing applications in a wide variety of engineering problems, recently applications include such areas as insurance, quantile hedging for guaranteed minimum death benefits, dividend payment and investment strategies with capital injection, singular control, risk management, portfolio selection with bounded constraints, and production planning and manufacturing problems; see Jin et al. (2011, 2012, 2013), Sethi and Zhang (1994), and Yin et al. (2009) and references therein.

Let us begin with the controlled diffusion problem. We wish to minimize the cost function defined by

$$J(x, u(\cdot)) = E_x \left[\int_0^\tau R(X(t), u(t)) dt + B(X(\tau)) \right], \quad (1)$$

with the $\mathbb{R}^r$ -valued process $X(t)$ defined by the solution of the stochastic differential equation

$$dX(t) = b(X(t), u(t))dt + \sigma(X(t))dW,$$

$$X(0) = x \quad (2)$$

where $x \in \mathbb{R}^r$, $u(\cdot)$ is a U -valued, measurable process with $U \subset \mathbb{R}^d$ being a compact control set, $W(\cdot)$ is an r -dimensional standard Brownian motion, and τ is the first exit time of the diffusion from a bounded domain D , that is, $\tau = \min\{t : X(t) \notin D^0\}$ with D^0 denoting the interior of D , $b(\cdot, \cdot) : \mathbb{R}^r \times \mathbb{R}^d \mapsto \mathbb{R}^r$, $\sigma(\cdot) : \mathbb{R}^r \mapsto \mathbb{R}^r \times \mathbb{R}^r$, and $R(\cdot, \cdot) : \mathbb{R}^r \times \mathbb{R}^d \mapsto \mathbb{R}$ and $B(\cdot) : \mathbb{R}^r \mapsto \mathbb{R}$. In the above, $b(\cdot)$ is the control-dependent drift, $\sigma(\cdot)$ is the diffusion matrix, $R(\cdot)$ is the running cost, and $B(\cdot)$ is the terminal or boundary cost. Throughout the entry, we assume that the stopping time $\tau < \infty$ with probability one (w.p.1) for simplicity. Denote the value function by $V(x) = \inf_u J(x, u(\cdot))$, where the inf is taken over all admissible controls. Write the transpose of $Y \in \mathbb{R}^{d_1 \times d_2}$ as Y' with $d_1, d_2 \geq 1$, $a(x) = \sigma(x)\sigma'(x)$, and define the generator of the controlled Markov process by

$$\mathcal{L}^u f(x) = \frac{1}{2} \text{tr}(a(x) f_{xx}(x)) + b'(x, u) f_x(x), \quad (3)$$

for a suitably smooth function $f(\cdot)$, where $f_x(\cdot)$ and $f_{xx}(\cdot)$ denote the gradient and Hessian of $f(\cdot)$, respectively. Note that the operator is control dependent. Using ∂D to denote the boundary of D , then the associated Hamilton-Jacobi-Bellman (HJB) equation satisfied by the value function is given by

$$\begin{cases} \inf_u [\mathcal{L}^u V(x) + R(x, u)] = 0, & x \in D^0, \\ V(x) = B(x), & x \in \partial D. \end{cases} \quad (4)$$

The subject matter of this article is to solve the optimal stochastic control problem numerically.

The rest of the entry is arranged as follows. Section “[Markov Chain Approximation](#)” focuses on Markov chain approximation. It illustrates how one can construct the controlled Markov chain in discrete time for the approximation of the continuous-time stochastic control problems. Section “[Illustration: A One-Dimensional Problem](#)” uses a one-dimensional case as an example for illustration. Section “[Numerical](#)

Computation” discusses the implementation issues. We conclude the entry with a few further remarks.

Markov Chain Approximation

The main idea was initiated in Kushner (1977) and streamlined, extended, and further developed in Kushner and Dupuis (1992). An earlier paper describing how to discretize the elliptic HJB equation and then interpret it according to a controlled Markov chain can be found in Kushner and Kleinman (1968). This section illustrates the Markov chain approximation methods with simple setup. The reader is suggested to read the references mentioned above for a comprehensive treatment. To begin, let $h > 0$ be a small “step size” in the approximation. Instead of the domain D , we need to work with a finite set to ensure computational feasibility. Set $\mathbb{R}_h^r$ to be r -dimensional lattice cube, i.e., $\mathbb{R}_h^r = \{\dots, -2h, -h, 0, h, 2h, \dots\}^r$ (an r -dimensional product of the indicated set). Denote the interior of D by D^0 , and define $D_h^0 = D^0 \cap \mathbb{R}_h^r$.

We shall construct a controlled, discrete-time Markov chain, whose transition probabilities have desired properties in line with the controlled diffusion and whose values are in D_h^0 . Suppose that $\{\alpha_n^h\}$ is a time-homogeneous, discrete-time, controlled Markov chain with finite state space D_h^0 and transition probabilities $P = (p(x, y|v))$ with $x, y \in D_h^0$. Here we only consider the case that the Markov chain has a finite state space. This is sufficient for our computational purposes. At any time n , the control action is a random variable denoted by u_n^h taking values in a compact set U . Set the interpolation interval by $\Delta t^h(x, v) > 0$ and write $\Delta t_n^h = \Delta t^h(\alpha_n, u_n^h)$ such that $\sup_{x,v} \Delta t^h(x, v) \rightarrow 0$ as $h \rightarrow 0$ but $\inf_{x,v} \Delta t^h(x, v) > 0$ for each $h > 0$. The control is admissible if the Markov property $P(\alpha_{n+1}^h = y | \alpha_i^h, u_i^h; i \leq n) = P(\alpha_{n+1}^h = y | \alpha_n^h, u_n^h) = P(\alpha_n^h, y | u_n^h)$ holds. Use U^h to denote the collection of controls, which are determined by a sequence of measurable functions $F_n^h(\cdot)$ such that $u_n^h = F_n^h(\alpha_k^h, k \leq n; u_k^h, k < n)$. Denote the conditional expectation given $\{\alpha_j^h, u_j^h : j \leq n, \alpha_n^h = x, u_n^h = v\}$ by E_n^h . We say that a control policy is locally consistent if

$$\begin{aligned} E_n^h \alpha_n^h &= b(x, v) \Delta t^h(x, v) + o(\Delta t^h(x, v)), \\ E_n^h [\Delta \alpha_n^h - E_n^h \Delta \alpha_n^h] [\Delta \alpha_n^h - E_n^h \Delta \alpha_n^h]' &= a(x) \Delta t^h(x, v) + o(\Delta t^h(x, v)), \\ a(x) &= \sigma(x) \sigma'(x), \quad |\Delta \alpha_n^h| \rightarrow 0 \text{ as } h \rightarrow 0 \text{ uniformly in } n, \omega, \end{aligned} \quad (5)$$

where $\Delta \alpha_n^h = \alpha_{n+1}^h - \alpha_n^h$. The meaning of the local consistency can be seen from the corresponding controlled diffusion (2) (with $X(0) = x$ and $u(t) = v$ for $t \in [0, \delta]$, where $\delta > 0$ is a small parameter) in that $E_x(X(\delta) - x) = b(x, v)\delta + o(\delta)$, $E_x[X(\delta) - x][X(\delta) - x]' = a(x)\delta + o(\delta)$. Let τ_h be the first time that $\{\alpha_n^h\}$ leaves the set D_h^0 . We have an approximation for the cost function of the controlled diffusion (1) given by

$$J^h(x, u^h) = E_x^{u^h} \left[\sum_{j=0}^{\tau_h-1} R(\alpha_j^h, u_j^h) \Delta t_j^h + B(\alpha_{\tau_h}^h) \right]. \quad (6)$$

Define $t_n^h = \sum_{j=0}^{n-1} \Delta t_j^h$ and the continuous-time interpolations $\alpha^h(t) = \alpha_n^h$, $u_n^h = u_n^h$ for $t \in [t_n^h, t_{n+1}^h)$. Define the first exit time of $\alpha^h(\cdot)$ from D_h^0 by $\tau_h = t_{\tau_h}^h$. Corresponding to the continuous-time problems, the first term on the right-hand side of (6) represents the running cost and the last term gives the terminal cost. Denote the value function by $V^h(x)$. Then it satisfies the dynamic programming equation

$$V^h(x) = \begin{cases} \inf_{v \in U^h} [R(x, v) \Delta t^h(x, v) \\ + \sum_y p^h(x, y|v) V^h(y)], & x \in D_h^0, \\ B(x), & x \notin D_h^0. \end{cases} \quad (7)$$

Proving the convergence of the numerical algorithms is an important task. This requires the use of local consistency, interpolation of the approximating sequences in continuous time, as well as martingale representation. The proof is facilitated by the use of the so-called relaxed controls (Kushner and Dupuis 1992, p. 267), which enables us to characterize the limit under the framework of weak convergence. The detailed argument is beyond the scope of this entry. We refer the reader to Kushner and Dupuis (1992, Chapter 10) for further reading on the proof of convergence and the conditions needed.

Illustration: A One-Dimensional Problem

In this section, we use a one-dimensional example to illustrate the Markov chain approximation

methods, which enables us to present the results with a better visualization. Consider (2) with $x \in \mathbb{R}$. We proceed to find the transition probabilities and interpolation intervals for the Markov chain $\{\alpha_n^h\}$. To construct a controlled Markov chain that is locally consistent, we first consider a special case, namely, the control space has only one admissible control $u^h \in U^h$. In this case, min in (7) can be removed. Discretize the HJB equation using upwind finite difference method with step size $h > 0$ by

$$\begin{aligned} V(x) &\rightarrow V^h(x) \\ V_x(x) &\rightarrow \frac{V^h(x+h) - V^h(x)}{h} \text{ for } b(x, v) > 0, \\ V_x(x) &\rightarrow \frac{V^h(x) - V^h(x-h)}{h} \text{ for } b(x, v) < 0, \\ V_{xx}(x) &\rightarrow \frac{V^h(x+h) - 2V^h(x) + V^h(x-h)}{h^2}. \end{aligned}$$

For $x \in D_h^0$, it leads to

$$\begin{aligned} &\frac{V^h(x+h) - V^h(x)}{h} b^+(x, v) - \frac{V^h(x) - V^h(x-h)}{h} b^-(x, v) \\ &+ \frac{h}{2} \frac{V^h(x+h) - 2V^h(x) + V^h(x-h)}{h^2} + R(x, v) = 0, \end{aligned}$$

where b^+ and b^- are the positive and negative parts of b , respectively. Comparing with the dynamic programming equation, we obtain the transition probabilities

$$\begin{aligned} p^h(x, x+h|v) &= \frac{(a(x)/2) + hb^+(x, v)}{\tilde{\Delta}}, \\ p^h(x, x-h|v) &= \frac{(a(x)/2) + hb^-(x, v)}{\tilde{\Delta}}, \\ p^h(\cdot) &= 0, \text{ otherwise, } \Delta t^h(x, v) = \frac{h^2}{\tilde{\Delta}}, \end{aligned}$$

with $\tilde{\Delta} = a(x) + h|b(x, v)|$ being well defined. With the transition probabilities given above, we can proceed to verify the local consistency by straight forward calculations and prove the desired convergence.

Numerical Computation

To numerically approximate the controlled diffusions, frequently used methods are either value

iterations or policy iterations (iteration in policy space). Using Markov chain approximation in conjunction with either value iteration or iteration in policy space, we can further obtain a sequence of value functions $\{V^{h,n}\}$ such that $V^{h,n} \rightarrow V^h$ as $n \rightarrow \infty$. The procedures can be described as follows.

Value Iteration

1. Given a tolerance $\varepsilon > 0$, set $n = 0$; for $x \in D_h^0$, set $V^{h,0} = \text{constant}$ (for instance, 0).
2. Using $V^{h,n}$ obtained in (7) to obtain $V^{h,n+1}$.
3. If $|V^{h,n+1} - V^{h,n}| > \varepsilon$, go to Step 3 above with $n \rightarrow n + 1$.

Policy Iteration

1. Given a tolerance $\varepsilon > 0$, set $n = 0$; for $x \in D_h^0$, take an initial control $u_0^h(x) = \text{constant}$. Use $u_0^h(x)$ in lieu of v , solve (7) to find $V^{h,0}(\cdot)$.

2. Find an improved control by

$$u^{h,n+1}(x) := \operatorname{argmin}_{v \in U^h} [\sum_y p^h((x, y)|v) V^{h,n}(y) + R(x, v) \Delta t^h(x, v)].$$
3. Find $V^{h,n+1}(\cdot)$ with $u^{h,n+1}(\cdot)$ by solving (7). If $|V^{h,n+1} - V^{h,n}| > \varepsilon$, go to Step 2 above with $n \rightarrow n + 1$.

Further Remarks

Variations of the Problems. Variants of the problems can be considered. For example, one may consider nonlinear filtering problems or singularly perturbed control and filtering problems. For problems arising in manufacturing systems, one often needs to treat controlled Markov chain with no diffusion terms. Such a case can also be handled by the Markov chain approximation methods; see Sethi and Zhang (1994) for the problem and Yin and Zhang (2013, Chapter 9) for the numerical methods. In this article, we mainly discussed the approach by using probabilistic approach for getting the weak convergence of the interpolations of the controlled Markov chain. One can also use the so-called viscosity solution methods to treat the convergence; see Barles and Souganidis (1991) (also Kushner and Dupuis 1992, Chapter 11).

Variance Control. In this entry, only drift involves control term. When the diffusion term is also subject to controls, the problem becomes more difficult. In Peng (1990), the idea of using backward stochastic differential equations was initiated, which had significant impact in the development of such stochastic control problems. Detailed discussions can be found in Yong and Zhou (1999). The numerical problems for diffusion term involving controls can also be treated; see Kushner (2000) for further discussion. In this case, the so-called numerical noise or numerical viscosity can be introduced, so care must be taken.

Complex Models Involving Jump and Switching. Note that only controlled diffusions are considered in this entry. More complex models such as controlled jump diffusions (Kushner and

Dupuis 1992), switching diffusions (Yin and Zhu 2010), and switching jump diffusions can be treated (Song et al. 2006). Differential games can also be treated (Kushner 2002; Song et al. 2008).

Differential Delay Systems. Stochastic differential delay systems may come into play. The corresponding numerical algorithms have been studied extensively in Kushner (2008). Due to their inherent infinite dimensionality, a main issue here concerns suitable finite approximation to the memory segments.

Rates of Convergence. This entry mainly discusses the convergence of the approximation methods. There is also much interest in ascertaining rates of convergence. Such effort goes back to the paper Menaldi (1989) (see also Zhang 2006). Subsequently, it has been resurgent effort in dealing with this issue from a nonlinear partial differential equation point of view; see Krylov (2000). Our recent work Song and Yin (2009) complements the study by providing a probabilistic approach for treating switching diffusions.

Stochastic Approximation. In certain optimal control problems, the optimal controls or near-optimal controls turn out to be of threshold type. An alternative way of solving such problems leading to at least suboptimal or near-optimal control is to use a stochastic approximation approach; see Kushner and Yin (2003) for a comprehensive treatment of stochastic approximation algorithms. Some successful examples include manufacturing systems (Yin and Zhang 2013, Section 9.3) and liquidation decision making (Yin et al. 2002).

Cross-References

- Stochastic Dynamic Programming
- Stochastic Maximum Principle

Acknowledgments The research of this author was supported in part by the Army Research Office under grant W911NF-12-1-0223.

Bibliography

- Barles G, Souganidis P (1991) Convergence of approximation schemes for fully nonlinear second order equations. *J Asymptot Anal* 4:271–283
- Bertsekas DP, Castanon DA (1989) Adaptive aggregation methods for infinite horizon dynamic programming. *IEEE Trans Autom Control* 34:589–598
- Chancelier P, Gomez C, Quadrat J-P, Sulem A, Blankenship GL, La Vigna A, MaCenary DC, Yan I (1986) An expert system for control and signal processing with automatic FORTRAN program generation. In: *Mathematical systems symposium*, Stockholm. Royal Institute of Technology, Stockholm
- Chancelier P, Gomez C, Quadrat J-P, Sulem A (1987) Automatic study in stochastic control. In: Fleming W, Lions PL (eds) *IMA volume in mathematics and its applications*, vol 10. Springer, Berlin
- Crandall MG, Lions PL (1983) Viscosity solutions of Hamilton-Jacobi equations. *Trans Am Math Soc* 277:1–42
- Crandall MG, Ishii H, Lions PL (1992) User's guide to viscosity solutions of second order partial differential equations. *Bull Am Math Soc* 27:1–67
- Fleming WH, Rishel RW (1975) *Deterministic and stochastic optimal control*. Springer, New York
- Fleming WH, Soner HM (1992) *Controlled Markov processes and viscosity solutions*. Springer, New York
- Jin Z, Wang Y, Yin G (2011) Numerical solutions of quantile hedging for guaranteed minimum death benefits under a regime-switching-jump-diffusion formulation. *J Comput Appl Math* 235:2842–2860
- Jin Z, Yin G, Zhu C (2012) Numerical solutions of optimal risk control and dividend optimization policies under a generalized singular control formulation. *Automatica* 48:1489–1501
- Jin Z, Yang HL, Yin G (2013) Numerical methods for optimal dividend payment and investment strategies of regime-switching jump diffusion models with capital injections. *Automatica* 49:2317–2329
- Krylov VN (2000) On the rate of convergence of finite-difference approximations for Bellman's equations with variable coefficients. *Probab Theory Relat Fields* 117:1–16
- Kushner HJ (1977) *Probability methods for approximation in stochastic control and for elliptic equations*. Academic, New York
- Kushner HJ (1990a) Weak convergence methods and singularly perturbed stochastic control and filtering problems. Birkhäuser, Boston
- Kushner HJ (1990b) Numerical methods for stochastic control problems in continuous time. *SIAM J Control Optim* 28:999–1048
- Kushner HJ (2000) Consistency issues for numerical methods for variance control with applications to optimization in finance. *IEEE Trans Autom Control* 44:2283–2296
- Kushner HJ (2002) Numerical approximations for stochastic differential games. *SIAM J Control Optim* 40:457–486
- Kushner HJ (2008) *Numerical methods for controlled stochastic delay systems*. Birkhäuser, Boston
- Kushner HJ, Dupuis PG (1992) *Numerical methods for stochastic control problems in continuous time*. Springer, New York
- Kushner HJ, Kleinman AJ (1968) Numerical methods for the solution of the degenerate nonlinear elliptic equations arising in optimal stochastic control theory. *IEEE Trans Autom Control* AC-13:344–353
- Kushner HJ, Yin G (2003) *Stochastic approximation and recursive algorithms and applications*, 2nd edn. Springer, New York
- Menaldi J (1989) Some estimates for finite difference approximations. *SIAM J Control Optim* 27:579–607
- Peng S (1990) A general stochastic maximum principle for optimal control problems. *SIAM J Control Optim* 28:966–979
- Sethi SP, Zhang Q (1994) *Hierarchical decision making in stochastic manufacturing systems*. Birkhäuser, Boston
- Song QS, Yin G (2009) Rates of convergence of numerical methods for controlled regime-switching diffusions with stopping times in the costs. *SIAM J Control Optim* 48:1831–1857
- Song QS, Yin G, Zhang Z (2006) Numerical method for controlled regime-switching diffusions and regime-switching jump diffusions. *Automatica* 42:1147–1157
- Song QS, Yin G, Zhang Z (2008) Numerical solutions for stochastic differential games with regime switching. *IEEE Trans Autom Control* 53:509–521
- Warga J (1962) Relaxed variational problems. *J Math Anal Appl* 4:111–128
- Yan HM, Yin G, Lou SXC (1994) Using stochastic optimization to determine threshold values for control of unreliable manufacturing systems. *J Optim Theory Appl* 83:511–539
- Yin G, Zhang Q (2013) *Continuous-time Markov chains and applications: a two-time-scale approach*, 2nd edn. Springer, New York
- Yin G, Zhu C (2010) *Hybrid switching diffusions: properties and applications*. Springer, New York
- Yin G, Liu RH, Zhang Q (2002) Recursive algorithms for stock liquidation: a stochastic optimization approach. *SIAM J Optim* 13:240–263
- Yin G, Jin H, Jin Z (2009) Numerical methods for portfolio selection with bounded constraints. *J Comput Appl Math* 233:564–581
- Yong J, Zhou XY (1999) *Stochastic controls: Hamiltonian systems and HJB equations*. Springer, New York
- Zhang J (2006) Rate of convergence of finite difference approximations for degenerate ODEs. *Math Comput* 75:1755–1778

Numerical Methods for Nonlinear Optimal Control Problems

Lars Grüne

Mathematical Institute, University of Bayreuth,
Bayreuth, Germany

Abstract

In this article we describe the three most common approaches for numerically solving nonlinear optimal control problems governed by ordinary differential equations. For computing approximations to optimal value functions and optimal feedback laws, we present the Hamilton-Jacobi-Bellman approach. For computing approximately optimal open-loop control functions and trajectories for a single initial value, we outline the indirect approach based on Pontryagin's maximum principle and the approach via direct discretization.

Keywords

Direct discretization ·
Hamilton-Jacobi-Bellman equations ·
Optimal control · Ordinary differential
equations · Pontryagin's maximum principle

Introduction

This article concerns optimal control problems governed by nonlinear ordinary differential equations of the form

$$\dot{x}(t) = f(x(t), u(t)) \quad (1)$$

with $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$. We assume that for each initial value $x \in \mathbb{R}^n$ and measurable control function $u(\cdot) \in L^\infty(\mathbb{R}, \mathbb{R}^m)$, there exists a unique solution $x(t) = x(t, x, u(\cdot))$ of (1) satisfying $x(0, x, u(\cdot)) = x$.

Given a state constraint set $X \subseteq \mathbb{R}^n$ and a control constraint set $U \subseteq \mathbb{R}^m$, a running cost $g : X \times U \rightarrow \mathbb{R}$, a terminal cost $F : X \rightarrow \mathbb{R}$, and

a discount rate $\delta \geq 0$, we consider the optimal control problem

$$\text{minimize } J^T(x, u(\cdot)) \quad (2)$$

$$u(\cdot) \in \mathcal{U}^T(x)$$

where

$$J^T(x, u(\cdot)) := \int_0^T e^{-\delta s} g(x(s, x, u(\cdot)), u(s)) ds$$

$$+ e^{-\delta T} F(x(T, x, u(\cdot))) \quad (3)$$

and

$$\mathcal{U}^T(x) := \left\{ u(\cdot) \in L^\infty(\mathbb{R}, U) \mid \begin{array}{l} x(s, x, u(\cdot)) \in X \\ \text{for all } s \in [0, T] \end{array} \right\}. \quad (4)$$

In addition to this finite horizon optimal control problem, we also consider the infinite horizon problem in which T is replaced by “ ∞ ,” i.e.,

$$\text{minimize } J^\infty(x, u(\cdot)) \quad (5)$$

$$u(\cdot) \in \mathcal{U}^\infty(x)$$

where

$$J^\infty(x, u(\cdot)) := \int_0^\infty e^{-\delta s} g(x(s, x, u(\cdot)), u(s)) ds \quad (6)$$

and

$$\mathcal{U}^\infty(x) := \left\{ u(\cdot) \in L^\infty(\mathbb{R}, U) \mid \begin{array}{l} x(s, x, u(\cdot)) \in X \\ \text{for all } s \geq 0 \end{array} \right\}, \quad (7)$$

respectively.

The term “solving” (2), (3), (4) or (5), (6), (7) can have various meanings. First, the optimal value functions

$$V^T(x) = \inf_{u(\cdot) \in \mathcal{U}^T(x)} J^T(x, u(\cdot))$$

or

$$V^\infty(x) = \inf_{u(\cdot) \in \mathcal{U}^\infty(x)} J^\infty(x, u(\cdot))$$

may be of interest. Second, and often more importantly, one would like to know the optimal

control policy. This can be expressed in *open-loop* form $u^* : \mathbb{R} \rightarrow U$, in which the function u^* depends on the initial value x and on the initial time which we set to 0 here. Alternatively, the optimal control can be computed in state- and time-dependent *closed-loop* form, in which a feedback law $\mu^* : \mathbb{R} \times X \rightarrow U$ is sought. Via $u^*(t) = \mu^*(t, x(t))$, this feedback law can then be used in order to generate the time-dependent optimal control function for all possible initial values. Since the feedback law is evaluated along the trajectory, it is able to react to perturbations and uncertainties which may make $x(t)$ deviate from the predicted path. Finally, knowing u^* or μ^* , one can reconstruct the corresponding optimal trajectory by solving

$$\begin{aligned}\dot{x}(t) &= f(x(t), u^*(t)) \quad \text{or} \\ \dot{x}(t) &= f(x(t), \mu^*(t, x(t))).\end{aligned}$$

Hamilton-Jacobi-Bellman Approach

In this section we describe the numerical approach to solving optimal control problems via Hamilton-Jacobi-Bellman equations. We first describe how this approach can be used in order to compute approximations to the optimal value function V^T and V^∞ , respectively, and afterwards how the optimal control can be synthesized using these approximations. In order to formulate this approach for finite horizon T , we interpret $V^T(x)$ as a function in T and x . We denote differentiation with respect to T and x with subscript T and x , i.e., $V_x^T(x) = dV^T(x)/dx$, $V_T^T(x) = dV^T(x)/dT$, etc.

We define the Hamiltonian of the optimal control problem as

$$H(x, p) := \max_{u \in U} \{-g(x, u) - p \cdot f(x, u)\},$$

with $x, p \in \mathbb{R}^n$, f from (1), g from (3) or (6), and “ $\cdot$ ” denoting the inner product in $\mathbb{R}^n$. Then, under appropriate regularity conditions on the problem data, the optimal value functions V^T and V^∞

satisfy the first order partial differential equations (PDEs)

$$V_T^T(x) + \delta V^T(x) + H(x, V_x^T(x)) = 0$$

and

$$\delta V^\infty(x) + H(x, V_x^\infty(x)) = 0$$

in the viscosity solution sense. In the case of V^T , the equation holds for all $T \geq 0$ with the boundary condition $V^0(x) = F(x)$.

The framework of viscosity solutions is needed because in general the optimal value functions will not be smooth; thus, a generalized solution concept for PDEs must be employed (see Bardi and Capuzzo Dolcetta 1997). Of course, appropriate boundary conditions are needed at the boundary of the state constraint set X .

Once the Hamilton-Jacobi-Bellman characterization is established, one can compute numerical approximations to V^T or V^∞ by solving these PDEs numerically. To this end, various numerical schemes have been suggested, including various types of finite element and finite difference schemes. Among those, semi-Lagrangian schemes, see Falcone (1997) or Falcone and Ferretti (2013), allow for a particularly elegant interpretation in terms of optimal control synthesis, which we explain for the infinite horizon case.

In the semi-Lagrangian approach, one takes advantage of the fact that by the chain rule for $p = V_x^\infty(x)$ and constant control functions u , the identity

$$\begin{aligned}\delta V^\infty(x) - p \cdot f(x, u) &= \left. \frac{d}{dt} \right|_{t=0} \\ &\quad - (1 - \delta t) V^\infty(x(t, x, u))\end{aligned}$$

holds. Hence, the left-hand side of this equality can be approximated by the difference quotient

$$\frac{V^\infty(x) - (1 - \delta h) V^\infty(x(h, x, u))}{h}$$

for small $h > 0$. Inserting this approximation into the Hamilton-Jacobi-Bellman equation, replacing $x(h, x, u)$ by a numerical approximation

$\tilde{x}(h, x, u)$ (in the simplest case, the Euler method $\tilde{x}(h, x, u) = x + hf(x, u)$), multiplying by h , and rearranging terms, one arrives at the equation

$$V_h^\infty(x) = \min_{u \in U} \left\{ hg(x, u) + (1 - \delta h) V_h^\infty(\tilde{x}(h, x, u)) \right\}$$

defining an approximation $V_h^\infty \approx V^\infty$. This is now a purely algebraic dynamic programming-type equation which can be solved numerically, e.g., by using a finite element approach. The equation is typically solved iteratively using a suitable minimization routine for computing the “min” in each iteration (in the simplest case, U is discretized with finitely many values, and the minimum is determined by direct comparison). We denote the resulting approximation of V^∞ by $\tilde{V}_h^\infty$. Here, approximation is usually understood in the L^∞ sense (see Falcone 1997 or Falcone and Ferretti 2013).

The semi-Lagrangian scheme is appealing for synthesis of an approximately optimal feedback because V_h^∞ is the optimal value function of the auxiliary discrete-time problem defined by $\tilde{x}$. This implies that the expression

$$\mu_h^*(x) := \operatorname{argmin}_{u \in U} \left\{ hg(x, u) + (1 - \delta h) V_h^\infty(\tilde{x}(h, x, u)) \right\},$$

is an optimal feedback control value for this discrete-time problem for the next time step, i.e., on the time interval $[t, t + h]$ if $x = x(t)$. This feedback law will be approximately optimal for the continuous-time control system when applied as a discrete-time feedback law, and this approximate optimality remains true if we replace V_h^∞ in the definition of μ_h^* by its numerically computable approximation $\tilde{V}_h^\infty$. A similar construction can be made based on any other numerical approximation $\tilde{V}^\infty \approx V^\infty$, but the explicit correspondence of the semi-Lagrangian scheme to a discrete-time auxiliary system facilitates the

interpretation and error analysis of the resulting control law.

The main advantage of the Hamilton-Jacobi approach is that it directly computes an approximately optimal feedback law. Its main disadvantage is that the number of grid nodes needed for maintaining a given accuracy in a finite element approach to compute $\tilde{V}_h^\infty$ in general grows exponentially with the state dimension n . This fact – known as the *curse of dimensionality* – restricts this method to low-dimensional state spaces. Unless special structure is available which can be exploited, as, e.g., in the max-plus approach (see McEneaney 2006), it is currently almost impossible to go beyond state dimensions of about $n = 10$, typically less for strongly nonlinear problems.

Maximum Principle Approach

In contrast to the Hamilton-Jacobi-Bellman approach, the approach via Pontryagin’s maximum principle does not compute a feedback law. Instead, it yields an approximately open-loop optimal control u^* together with an approximation to the optimal trajectory x^* for a fixed initial value. We explain the approach for the finite horizon problem. For simplicity of presentation, we omit state constraints in our presentation, i.e., we set $X = \mathbb{R}^n$ and refer to, e.g., Vinter (2000), Bryson and Ho (1975), or Grass et al. (2008) for more general formulations as well as for rigorous versions of the following statements.

In order to state the maximum principle (which, since we are considering a minimization problem here, could also be called minimum principle), we define the non-minimized Hamiltonian as

$$\mathcal{H}(x, p, u) = g(x, u) + p \cdot f(x, u).$$

Then, under appropriate regularity assumptions, there exists an absolutely continuous function $p : [0, T] \rightarrow \mathbb{R}^n$ such that the optimal trajectory x^* and the corresponding optimal control function u^* for (2), (3) and (4) satisfy

$$\dot{p}(t) = \delta p(t) - \mathcal{H}_x(x^*(t), p(t), u^*(t)) \quad (8)$$

with terminal or transversality condition

$$p(T) = F_x(x^*(T)) \quad (9)$$

and

$$u^*(t) = \operatorname{argmin}_{u \in U} \mathcal{H}(x^*(t), p(t), u), \quad (10)$$

for almost all $t \in [0, T]$ (see Grass et al. 2008, Theorem 3.4). The variable p is referred to as the *adjoint* or *costate* variable.

For a given initial value $x_0 \in \mathbb{R}^n$, the numerical approach now consists of finding functions $x : [0, T] \rightarrow \mathbb{R}^n$, $u : [0, T] \rightarrow U$ and $p : [0, T] \rightarrow \mathbb{R}^n$ satisfying

$$\dot{x}(t) = f(x(t), u(t)) \quad (11)$$

$$\dot{p}(t) = \delta p(t) - \mathcal{H}_x(x(t), p(t), u(t)) \quad (12)$$

$$u(t) = \operatorname{argmin}_{u \in U} \mathcal{H}(x(t), p(t), u) \quad (13)$$

$$x(0) = x_0, \quad p(T) = F_x(x(T)) \quad (14)$$

for $t \in [0, T]$. Depending on the regularity of the underlying data, the conditions (11), (12), (13) and (14) may only be necessary but not sufficient for x and u being an optimal trajectory x^* and control function u^* , respectively. However usually x and u satisfying these conditions, are good candidates for the optimal trajectory and control, thus justifying the use of these conditions for the numerical approach. If needed, optimality of the candidates can be checked using suitable sufficient optimality conditions for which we refer to, e.g., Maurer (1981) or Malanowski et al. (2004). Due to the fact that in the maximum principle approach first optimality conditions are derived which are then discretized for numerical simulation, it is also termed *first optimize then discretize*.

Solving (11), (12), (13) and (14) numerically amounts to solving a boundary value problem, because the condition $x^*(0) = x_0$ is posed at the beginning of the time interval $[0, T]$, while the condition $p(T) = F_x(x^*(T))$ is required at

the end. In order to solve such a problem, the simplest approach is the *single shooting* method which proceeds as follows:

We select a numerical scheme for solving the ordinary differential equations (11) and (12) for $t \in [0, T]$ with initial conditions $x(0) = x_0$, $p(0) = p_0$ and control function $u(t)$. Then, we proceed iteratively as follows:

- (0) Find initial guesses $p_0^0 \in \mathbb{R}^n$ and $u^0(t)$ for the initial costate and the control, fix $\varepsilon > 0$, and set $k := 0$.
- (1) Solve (11) and (12) numerically with initial values x_0 and p_0^k and control function u^k . Denote the resulting trajectories by $\tilde{x}^k(t)$ and $\tilde{p}^k(t)$.
- (2) Apply one step of an iterative method for solving the zero-finding problem $G(p) = 0$ with

$$G(p_0^k) := \tilde{p}^k(T) - F_x(\tilde{x}^k(T))$$

for computing p_0^{k+1} . For instance, in case of the Newton method we get

$$p_0^{k+1} := p_0^k - DG(p_0^k)^{-1}G(p_0^k).$$

If $\|p_0^{k+1} - p_0^k\| < \varepsilon$, stop; else compute

$$u^{k+1}(t) := \operatorname{argmin}_{u \in U} \mathcal{H}(\tilde{x}^k(t), \tilde{p}^k(t), u),$$

set $k := k + 1$, and go to (1).

The procedure described in this algorithm is called *single shooting* because the iteration is performed on the single initial value p_0^k . For an implementable scheme, several details still need to be made precise, e.g., how to parameterize the function $u(t)$ (e.g., piecewise constant, piecewise linear or polynomial), how to compute the derivative DG and its inverse (or an approximation thereof), and the argmin in (2). The last task considerably simplifies if the structure of the optimal control, e.g., the number of switchings in case of a bang-bang control, is known.

However, even if all these points are settled, the set of initial guesses p_0^0 and u^0 for which the method is going to converge to a solution of (11), (12), (13) and (14) tends to be very small. One reason for this is that the solutions of (11) and (12) typically depend very sensitively on p_0^0 and u^0 . In order to circumvent this problem, *multiple shooting* can be used. To this end, one selects a time grid $0 = t_0 < t_1 < t_2 < \dots < t_N = T$ and in addition to p_0^k introduces variables $x_1^k, \dots, x_{N-1}^k, p_1^k, \dots, p_{N-1}^k \in$

$\mathbb{R}^n$. Then, starting from initial guesses p_0^0, u^0 , and $x_1^0, \dots, x_{N-1}^0, p_1^0, \dots, p_{N-1}^0$, in each iteration the Eqs. (11), (12), (13) and (14) are solved numerically on the intervals $[t_j, t_{j+1}]$ with initial values x_j^k and p_j^k , respectively. We denote the respective solutions in the k -th iteration by $\tilde{x}_j^k$ and $\tilde{p}_j^k$. In order to enforce that the trajectory pieces computed on the individual intervals $[t_j, t_{j+1}]$ fit together continuously, the map G is redefined as

$$G(x_1^k, \dots, x_{N-1}^k, p_0^k, p_1^k, \dots, p_{N-1}^k) = \begin{pmatrix} \tilde{x}_0^k(t_1) - x_1^k \\ \vdots \\ \tilde{x}_{N-2}^k(t_1) - x_{N-1}^k \\ \tilde{p}_0^k(t_1) - p_1^k \\ \vdots \\ \tilde{p}_{N-2}^k(t_1) - p_{N-1}^k \\ \tilde{p}_{N-1}^k(T) - F_x(\tilde{x}_{N-1}^k(T)) \end{pmatrix}.$$

The benefit of this approach is that the solutions on the shortened time intervals depend much less sensitively on the initial values and the control, thus making the problem numerically much better conditioned. The obvious disadvantage is that the problem becomes larger as the function G is now defined on a much higher dimensional space, but this additional effort usually pays off.

While the convergence behavior for the multiple shooting method is considerably better than for single shooting, it is still a difficult task to select good initial guesses x_j^0, p_j^0 and u^0 . In order to accomplish this, homotopy methods can be used (see, e.g., Pesch 1994), or the result of a direct approach as presented in the next section can be used as an initial guess. The latter can be reasonable as the maximum principle-based approach can yield approximations of higher accuracy than the direct method.

In the presence of state constraints or mixed state and control constraints, the conditions (12), (13) and (14) become considerably more technical and thus more difficult to be implemented numerically (cf. Pesch 1994).

Direct Discretization

Despite being the most straightforward and simple of the approaches described in this article, the direct discretization approach is currently the most widely used approach for computing single finite horizon optimal trajectories. In the direct approach, we first discretize the problem and then solve a finite dimensional nonlinear optimization problem (NLP), i.e., we *first discretize, then optimize*. The main reasons for the popularity of this approach are the simplicity with which constraints can be handled and the numerical efficiency due to the availability of fast and reliable NLP solvers.

The direct approach again applies to the finite horizon problem and computes an approximation to a single optimal trajectory $x^*(t)$ and control function $u^*(t)$ for a given initial value $x_0 \in X$. To this end, a time grid $0 = t_0 < t_1 < t_2 < \dots < t_N = T$ and a set $\mathcal{U}_d$ of control functions which are parameterized by finitely many values are selected. The simplest way to do so is to choose $u(t) \equiv u_j \in U$ for all $t \in [t_i, t_{i+1}]$. However, other approaches like piecewise linear

or piecewise polynomial control functions are possible, too. We use a numerical algorithm for ordinary differential equations in order to approximately solve the initial value problems

$$\dot{x}(t) = f(x(t), u_i), \quad x(t_i) = x_i \quad (15)$$

for $i = 0, \dots, N-1$ on $[t_i, t_{i+1}]$. We denote the exact and numerical solution of (15) by $x(t, t_i, x_i, u_i)$ and $\tilde{x}(t, t_i, x_i, u_i)$, respectively. Finally, we choose a numerical integration rule in order to compute an approximation

$$I(t_i, t_{i+1}, x_i, u_i) \approx \int_{t_i}^{t_{i+1}} e^{-\delta t} g(x(t, t_i, x_i, u), u(t)) dt.$$

In the simplest case, one might choose $\tilde{x}$ as the Euler scheme and I as the rectangle rule, leading to

$$\tilde{x}(t_{i+1}, t_i, x_i, u_i) = x_i + (t_{i+1} - t_i) f(x_i, u_i)$$

and

$$I(t_i, t_{i+1}, x_i, u_i) = (t_{i+1} - t_i) e^{-\delta t_i} g(x_i, u_i).$$

Introducing the optimization variables $u_0, \dots, u_{N-1} \in \mathbb{R}^m$ and $x_1, \dots, x_N \in \mathbb{R}^n$, the discretized version of (2), (3) and (4) reads

$$\underset{x_j \in \mathbb{R}^n, u_j \in \mathbb{R}^m}{\text{minimize}} \sum_{i=0}^{N-1} I(t_i, t_{i+1}, x_i, u) + e^{-\delta T} F(x_N)$$

subject to the constraints

$$\begin{aligned} u_j &\in U, & j &= 0, \dots, N-1 \\ x_j &\in X, & j &= 1, \dots, N \\ x_{j+1} &= \tilde{x}(t_{j+1}, t_j, x_j, u), & j &= 0, \dots, N \end{aligned}$$

This way, we have converted the optimal control problem (2), (3) and (4) into a finite dimensional nonlinear optimization problem (NLP). As such, it can be solved with any numerical method for solving such problems. Popular methods are, for instance, sequential quadratic

programming (SQP) or interior point (IP) algorithms. The convergence of this approach was proved in Malanowski et al. (1998); for an up-to-date account on theory and practice of the method; see Gerds (2012) and Betts (2010). These references also explain how information about the costates $p(t)$ can be extracted from a direct discretization, thus linking the approach to the maximum principle.

The direct method sketched here is again a multiple shooting method, and the benefit of this approach is the same as for solving boundary problems: thanks to the short intervals $[t_i, t_{i+1}]$, the solutions depend much less sensitively on the data than the solution on the whole interval $[0, T]$, thus making the iterative solution of the resulting discretized NLP much easier. The price to pay is again the increase of the number of optimization variables. However, due to the particular structure of the constraints guaranteeing continuity of the solution, the resulting matrices in the NLP have a particular structure which can be exploited numerically by a method called *condensing* (see Bock and Plitt 1984).

An alternative to multiple shooting is collocation, in which the internal variables of the numerical algorithm for solving (15) are also optimization variables. A popular subclass of these algorithms are pseudospectral methods, in which Gauss quadrature points are used for the collocation; see Garg et al. (2010). However, nowadays, the multiple shooting approach as described above is often preferred. For a more detailed description of various direct approaches, see also Binder et al. (2001), Sect. 5.

Further Approaches for Infinite Horizon Problems

The last two approaches only apply to finite horizon problems. While the maximum principle approach can be generalized to infinite horizon problems, the necessary conditions become weaker, and the numerical solution becomes considerably more involved (see Grass et al. 2008). Both the maximum principle and the direct approach can, however, be applied in a

receding horizon fashion, in which an infinite horizon problem is approximated by the iterative solution of finite horizon problems. The resulting control technique is known under the name of model predictive control (MPC; see Grüne and Pannek 2017). Under suitable assumptions, in which the classical turnpike property of optimal control problems plays an important role, rigorous approximation results can be established.

Summary and Future Directions

The three main numerical approaches to optimal control are:

- The Hamilton-Jacobi-Bellman approach, which provides a global solution in feedback form but is computationally expensive for higher dimensional systems
- The Pontryagin maximum principle approach which computes single optimal trajectories with high accuracy but needs good initial guesses for the iteration
- The direct approach which also computes single optimal trajectories but is less demanding in terms of the initial guesses at the expense of a somewhat lower accuracy

Currently, the main trends in numerical optimal control lie in the areas of Hamilton-Jacobi-Bellman equations and direct discretization. For the former, the development of discretization schemes suitable for increasingly higher dimensional problems is in the focus. For the latter, the popularity of these methods in online applications like MPC triggers continuing effort to make this approach faster and more reliable.

Beyond ordinary differential equations, the development of numerical algorithms for the optimal control of partial differential equations (PDEs) has attracted considerable attention during the last years. While many of these methods are still restricted to linear systems, in the near future, we can expect to see many extensions to (classes of) nonlinear PDEs. It is

worth noting that for PDEs, first-optimize-then-discretize approaches are more popular than for ordinary differential equations.

Cross-References

- [Discrete Optimal Control](#)
- [Economic Model Predictive Control](#)
- [Nominal Model-Predictive Control](#)
- [Optimal Control and Pontryagin's Maximum Principle](#)
- [Optimal Control and the Dynamic Programming Principle](#)
- [Optimization Algorithms for Model Predictive Control](#)

Bibliography

- Bardi M, Capuzzo Dolcetta I (1997) Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations. Birkhäuser, Boston
- Betts JT (2010) Practical methods for optimal control and estimation using nonlinear programming, 2nd edn. SIAM, Philadelphia
- Binder T, Blank L, Bock HG, Bulirsch R, Dahmen W, Diehl M, Kronseder T, Marquardt W, Schlöder JP, von Stryk O (2001) Introduction to model based optimization of chemical processes on moving horizons. In: Grötschel M, Krumke SO, Rambau J (eds) Online optimization of large scale systems: state of the art. Springer, Heidelberg, pp 295–340
- Bock HG, Plitt K (1984) A multiple shooting algorithm for direct solution of optimal control problems. In: Proceedings of the 9th IFAC world congress, Budapest. Pergamon, Oxford, pp 242–247
- Bryson AE, Ho YC (1975) Applied optimal control. Hemisphere Publishing Corp., Washington, DC. Revised printing
- Falcone M (1997) Numerical solution of dynamic programming equations. In: Appendix A in Bardi M, Capuzzo Dolcetta I (eds) Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations. Birkhäuser, Boston
- Falcone M, Ferretti R (2013) Semi-Lagrangian approximation schemes for linear and Hamilton-Jacobi equations. SIAM, Philadelphia
- Garg D, Patterson M, Hager WW, Rao AV, Benson DA, Huntington GT (2010) A unified framework for the numerical solution of optimal control problems using pseudospectral methods. *Automatica* 46(11):1843–1851
- Gerdts M (2012) Optimal control of ODEs and DAEs. De Gruyter textbook. Walter de Gruyter & Co., Berlin

- Grass D, Caulkins JP, Feichtinger G, Tragler G, Behrens DA (2008) Optimal control of nonlinear processes. Springer, Berlin
- Grüne L, Pannek J (2017) Nonlinear model predictive control: theory and algorithms, 2nd edn. Springer, Cham
- Malanowski K, Büskens C, Maurer H (1998) Convergence of approximations to nonlinear optimal control problems. In: Fiacco AV (ed) Mathematical programming with data perturbations. Lecture notes in pure and applied mathematics, vol 195. Dekker, New York, pp 253–284
- Malanowski K, Maurer H, Pickenhain S (2004) Second-order sufficient conditions for state-constrained optimal control problems. *J Optim Theory Appl* 123(3):595–617
- Maurer H (1981) First and second order sufficient optimality conditions in mathematical programming and optimal control. *Math Program Stud* 14: 163–177
- McEneaney WM (2006) Max-plus methods for nonlinear control and estimation. *Systems & control: foundations & applications*. Birkhäuser, Boston
- Pesch HJ (1994) A practical guide to the solution of real-life optimal control problems. *Control Cybern* 23(1–2):7–60
- Vinter R (2000) Optimal control. *Systems & control: foundations & applications*. Birkhäuser, Boston

Observer-Based Control

H.L. Trentelman¹ and Panos J. Antsaklis²

¹Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, Groningen, AV, The Netherlands

²Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

An observer-based controller is a dynamic feedback controller with a two-stage structure. First, the controller generates an estimate of the state variable of the system to be controlled, using the measured output and known input of the system. This estimate is generated by a state observer for the system. Next, the state estimate is treated as if it were equal to the exact state of the system, and it is used by a static state feedback controller. Dynamic feedback controllers with this two-stage structure appear in various control synthesis problems for linear systems. In this entry, we explain observer-based control in the context of internal stabilization by dynamic measurement feedback.

Keywords

Detectability · Dynamic output feedback control · Internal stabilization · Separation

principle · Stabilizability · State observers · Static state feedback

Introduction

In this entry, we explain the notion of observer-based feedback control. Given a to-be-controlled system in input-state-output form, together with a control objective, the problem is to design a feedback controller such that the closed-loop system meets the objective. In the case when all state variables of the system are available for control, the design problem is considered to be simpler, and often the controller can be chosen to be a static state feedback control law. In the more general case where the controller has access only to a linear function of the state variables, the problem is more involved and requires the design of a dynamic feedback control law. The key idea of observer-based feedback control is the following. As a first step, one determines a state observer for the system, i.e., a system that estimates the state of the system based on the measured outputs and inputs of the system. Next, the state estimate is treated as if it were exactly equal to the actual state of the system and is used by a static state feedback controller. In this way, a dynamic feedback controller is obtained that is composed of a (dynamic) state observer and a static feedback part.

Dynamic Output Feedback Control

Consider the controlled and observed system Σ :

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + Ed(t), \\ y(t) &= Cx(t), \\ z(t) &= Hx(t),\end{aligned}\tag{1}$$

with $x(t) \in \mathcal{X} = \mathbb{R}^n$ the state, $u(t) \in \mathbb{R}^m$ the control input, and $y(t) \in \mathbb{R}^p$ the measured output. The signal $d(t)$ may represent a disturbance input or a desired reference signal, while the signal $z(t)$ is a controlled output signal. A , B , C , E , and H are maps (or matrices). In general, a linear controller for this system is a finite-dimensional linear time-invariant system Γ represented by

$$\begin{aligned}\dot{w}(t) &= Kw(t) + Ly(t), \\ u(t) &= Mw(t) + Ny(t).\end{aligned}\tag{2}$$

The state space of the controller is assumed to be $\mathcal{W} = \mathbb{R}^q$ for some positive integer q . K , L , M , and N are assumed to be linear maps (or matrices). The controller (2) takes the observations y as its input and generates the control function u as its output. The closed-loop system resulting from the interconnection of Σ and Γ is described by the equations

$$\begin{aligned}\begin{pmatrix} \dot{x}(t) \\ \dot{w}(t) \end{pmatrix} &= \begin{pmatrix} A+BNC & BM \\ LC & K \end{pmatrix} \begin{pmatrix} x(t) \\ w(t) \end{pmatrix} + \begin{pmatrix} E \\ 0 \end{pmatrix} d(t), \\ z(t) &= \begin{pmatrix} H & 0 \end{pmatrix} \begin{pmatrix} x(t) \\ z(t) \end{pmatrix}.\end{aligned}\tag{3}$$

The control action of interconnecting the controller Γ with the system (1) is called *dynamic feedback*. The state space of the closed-loop system (3) is called the *extended state space* and is equal to the Cartesian product $\mathcal{X} \times \mathcal{W} = \mathbb{R}^{n+q}$. In general, a feedback control problem amounts to finding linear maps K , L , M , and N such that the closed-loop system (3) satisfies the control design specifications.

Observer-Based Controllers

Given the system (1) and a control objective, the problem thus arises on how to determine the maps K , L , M , and N so that the closed-loop systems meet the objective. As an example, take the special case when E in (1) is equal to zero (i.e., the system has no external disturbances or reference signals) and that we wish the closed-loop system (3) to be internally stable, i.e., we want to find the maps K , L , M , and N so that the eigenvalues λ_i of the system map of (3) are in the open left half-plane, i.e., satisfy $\text{Re}(\lambda_i) < 0$ for all i . If we had access to the entire state variable x (instead of only to the linear function $y = Cx$), then this problem would be simpler: assuming that the system is stabilizable (The system $\dot{x} = Ax + Bu$ is called stabilizable if there exists a map F such that $A + BF$ has all its eigenvalues in the open left half-plane), find a map F such that the eigenvalues of $A + BF$ are in the open left half-plane; then take the static state feedback controller $u = Fx$ as the control law. That is, we would choose the state space dimension of the controller Γ equal to 0 and the maps K , L , and M to be void, and we would take $N = F$.

In general, however, we only have access to a given linear function $y = Cx$ of x , determined by the output map C . The key idea of observer-based control is the following:

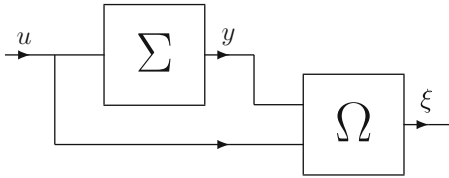
Use the theory of observer design to find an observer for the state x of the system (1), i.e., an observer that generates an estimate ξ of the system state x based on the measured output y and the control input u . Next, apply a static feedback $u = F\xi$ mimicking the (not permissible) control law $u = Fx$.

This idea leads to a dynamic feedback controller (2) of a very particular structure: the controller is the combination of a state observer (with a certain state space dimension) and a static control law acting on the state estimate. This *two-stage* structure, separating estimation and control, is often called the *separation principle*. We will work out this idea in more detail for the case when $E = 0$ (no external disturbances or reference signals) and the aim is to obtain internal

stability of the system. Before doing this, we first explain the most important material on observers that is needed in the sequel.

State Observers

If the state is not available for measurement, one can try to reconstruct it using a system, called observer, that takes the control input and the measured output of the original system as inputs and yields an output that is an estimate of the state of the original system. Again in case that in the system (1) we have $E = 0$, i.e., there are no disturbance signals. This is illustrated in the following picture:



The quantity ξ is supposed to be an estimate, in some sense, of the state, and w is the state variable of the observer. In general, the observer, denoted by Ω , has equations of the form

$$\begin{aligned}\dot{w}(t) &= Pw(t) + Qu(t) + Ry(t), \\ \xi(t) &= Sw(t).\end{aligned}\quad (4)$$

It turns out that particular choices for P , Q , R , and S , specifically $P = A - GC$ (where the map G has to be determined), $Q = B$, $R = G$, and $S = I$, lead to

$$\dot{\xi}(t) = (A - GC)\xi(t) + Bu(t) + Gy(t). \quad (5)$$

Introducing the *estimation error* $e := \xi - x$ and interconnecting the system (1) with (5), we find that the error e satisfies the differential equation

$$\dot{e}(t) = (A - GC)e(t). \quad (6)$$

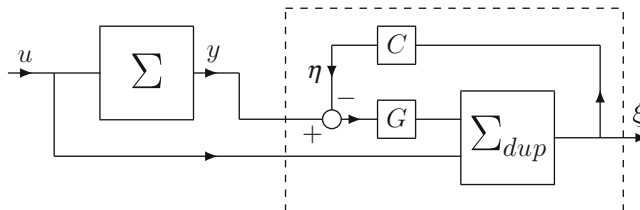
Hence all possible errors converge to 0 as t tends to infinity if and only if $A - GC$ is a stability matrix, i.e., has all its eigenvalues in the open left half-plane. In that case, we call (5) a *stable state observer*. Thus, a stable state observer exists if and only if G can be found such that $A - GC$ is a stability matrix. The problem of finding such a G is dual to the problem of finding a matrix F to a pair (A, B) such that $A + BF$ is a stability matrix.

Definition 1 The pair (C, A) is called *detectable* if there exists a matrix G such that $A - GC$ is a stability matrix, i.e., has all its eigenvalues in the open left half-plane.

Theorem 1 Given system Σ , the following statements are equivalent:

1. Σ has a stable state observer.
2. (C, A) is detectable.

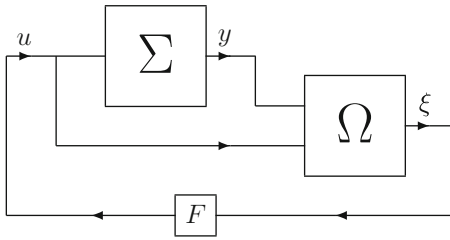
The equation for ξ can be rewritten using an artificial output $\eta = C\xi$ as $\dot{\xi} = A\xi + Bu + G(y - \eta)$. The interpretation of this is as follows. If ξ is the exact state, then $\eta = y$, and hence ξ obeys exactly the same differential equation as x . Otherwise, the equation for ξ has to be corrected by a term determined by the *output error* $y - \eta$. Consequently, the state observer consists of an exact replica Σ_{dup} of the original system with an extra input channel for incorporating the output error and an extra output, the state of the observer, which serves as the desired estimate for the state of the original system. The following diagram depicts the situation:



Observer-Based Stabilization

We now work out the ideas put forward in the previous sections for the special case of stabilization by dynamic measurement feedback, i.e., to find a controller (2) such that the closed-loop system (3) is internally stable; equivalently, the system mapping of (3) is a stability matrix. Again, we restrict ourselves to the case when $E = 0$.

We assume that we know how to stabilize by state feedback and how to build a state observer. If we have a plant of which we do not have the state available for measurement, we use a state observer to obtain an estimate of the state, and we apply the state feedback to this estimate rather than to the true state. This is illustrated by the following picture:



Again, consider the system Σ given by (1) and let the observer Ω be given by (5). Combining this with $u = F\xi$ yields

$$\begin{aligned}\dot{x}(t) &= Ax(t) + BF\xi, \\ \dot{\xi}(t) &= (A - GC + BF)\xi(t) + GCx(t).\end{aligned}\quad (7)$$

Introducing again $e := \xi - x$, we obtain, in accordance with the previous section,

$$\begin{aligned}\dot{x}(t) &= (A + BF)x(t) + BFe(t), \\ \dot{e}(t) &= (A - GC)e(t).\end{aligned}$$

That is, the equation $\dot{x}_e = A_e x_e$ with

$$x_e := \begin{pmatrix} x \\ e \end{pmatrix}, \quad A_e := \begin{pmatrix} A + BF & BF \\ 0 & A - GC \end{pmatrix}.$$

Assume that Σ is stabilizable and detectable. Then F and G can be found such that $A + BF$ and $A - GC$ are stability matrices. Since the set of eigenvalues of A_e is the union of those of $A + BF$ and $A - GC$, it follows that A_e is a stability matrix. Consequently, the system $\dot{x}_e = A_e x_e$ is asymptotically stable; equivalently, every solution $x_e = (x, e)$ converges to 0 as t tends to infinity. Of course, if (x, ξ) is a solution of (7), then $\xi = x + e$, with $x_e = (x, e)$ a solution of $\dot{x}_e = A_e x_e$. Hence (x, ξ) also converges to 0 as t goes to infinity. Thus we have proved the “if” part of the following theorem:

Theorem 2 *There exists an internally stabilizing dynamic feedback controller for Σ if and only if Σ is stabilizable and detectable. A controller is given by*

$$\begin{aligned}\dot{\xi}(t) &= (A - GC)\xi(t) + Bu(t) + Gy(t), \\ u(t) &= F\xi(t),\end{aligned}\quad (8)$$

where F is any map such that $A + BF$ is a stability matrix and G is any map such that $A - GC$ is a stability matrix.

The controller (8) is an *observer-based dynamic feedback controller*, since it is composed of a state observer and a static feedback part.

Summary and Future Directions

We have given an introduction to observer-based feedback controllers and have explained that such controllers are dynamic feedback controllers that can be represented as the composition of a state observer for the system, together with a static control law mimicking a (not permitted) static state feedback control law. We have given a detailed description of this principle for the case that the system to be controlled has no external disturbances or reference signals and the control objective is internal stability of the system. More intricate versions of the principle of observer-based feedback control appear in control design problems for linear systems with external disturbances and reference signals and with different, more sophisticated, control objectives. Examples

of these are the regulator problem, the problem of disturbance decoupling with internal stability, the $\mathcal{H}_2$ optimal control problem, and the $\mathcal{H}_\infty$ suboptimal control problem.

Cross-References

- [Linear State Feedback](#)
- [Observers in Linear Systems Theory](#)

Bibliography

- Antsaklis PJ, Michel AN (2007) A linear systems primer. Birkhäuser, Boston
- Trentelman HL, Stoorvogel AA, Hautus MLJ (2001) Control theory for linear systems. Springer, London
- Wonham WM (1979) Linear multivariable control: a geometric approach. Springer, New York
- Zadeh LA, Desoer CA (1963) Linear systems theory – the state-space approach. McGraw-Hill, New York

Observers for Nonlinear Systems

Laurent Praly
MINES ParisTech, PSL, Fontainebleau, France

Abstract

Observers are objects delivering estimation of variables which cannot be directly measured. The access to such *hidden* variables is made possible by combining modeling and measurements. But this is bringing face to face real world and its abstraction with, as a result, the need for dealing with uncertainties and approximations leading to difficulties in implementation and convergence.

Keywords

Estimation · Distinguishability · Detectability

Introduction

Observers are answers to the question of estimating, from observed/measured/empirical variables, denoted y , and delivered by sensors equipping a real world system, some “theoretical” variables, called hidden variables in this text, denoted z , which are involved in a mathematical model related to this system. The measured variables make what is called the a posteriori information on the hidden variables, whereas the model is part of the a priori information. Because a model cannot fit exactly a system, introduction of uncertainties is mandatory.

Typically this model describing the link between hidden and measured variables is made of three components:

- *a dynamic model* describes the dynamics/evolution ($\dot{x}$ denotes the time derivative $\frac{dx}{dt}$):

$$\begin{aligned}\dot{x}(t) &= f(x(t), t, \delta^s(t)) \quad \text{resp.} \\ x_{k+1} &= f_k(x_k, \delta_k^s),\end{aligned}\tag{1}$$

where t , in the continuous case, or k , in the discrete case, is an evolution parameter, called time in this text; x is a state, assumed finite dimensional in this text; and δ^s represents the uncertainties in the state dynamics. Any possible known inputs is represented here by the time-dependence of f .

- *a sensor model* relates state and measured variables:

$$\begin{aligned}y(t) &= h(x(t), t, \delta^m(t)) \quad \text{resp.} \\ y_k &= h_k(x_k, \delta_k^m)\end{aligned}\tag{2}$$

with δ_k^m representing the uncertainties in the measurements.

- a model which relates state and hidden variables:

$$\begin{aligned}z(t) &= \eta(x, t, \delta^h(t)) \quad \text{resp.} \\ z_k &= \eta_k(x_k, \delta_k^h).\end{aligned}\tag{3}$$

where again δ^h represents the uncertainties in the hidden variables.

In a deterministic setting, the a priori information on the uncertainties ($\delta^s, \delta^m, \delta^h$) may be that the values of δ^s, δ^m and δ^h are unknown but belong to known sets Δ^s, Δ^m and Δ^h . Namely, we have:

$$\begin{aligned} \delta^s(t) &\in \Delta^s(t), \quad \delta^m(t) \in \Delta^m(t), \quad \delta^h(t) \in \Delta^h(t), \\ \text{respectively} \quad \delta_k^s &\in \Delta_k^s, \quad \delta_k^m \in \Delta_k^m, \quad \delta_k^h \in \Delta_k^h. \end{aligned} \quad (4)$$

In a stochastic setting and more specifically in a Bayesian approach, it may be that δ^s, δ^m and δ^h are unknown realization of stochastic processes for which we know the probability distributions.

Similarly we may also know a priori that we have:

$$\begin{aligned} x(t) &\in \mathcal{X}(t), \quad z(t) \in \mathcal{Z}(t) \\ \text{respectively} \quad x_k &\in \mathcal{X}_k, \quad z_k \in \mathcal{Z}_k \end{aligned} \quad (5)$$

where the sets $\mathcal{X}$ and $\mathcal{Z}$ are known or we may have a priori probability distribution for x and z .

In this context, the a priori information is the data of the functions f, h , and q , of the sets $\Delta^s, \Delta^m, \Delta^h$ or the corresponding probability distribution and so maybe also of the sets $\mathcal{X}$ and $\mathcal{Z}$ or the corresponding a priori probability distribution.

In the next section, we state the observation problem and give the solutions which are direct consequences of the deterministic and stochastic setting given above. This will allow us to see that an observer is actually a dynamical system with the measurements as inputs and the estimate as output. But approximations in the implementation of these solutions, not knowing how to initialize, ... may lead to convergence problems even when the uncertainties disappear. The second part of this text is devoted to this convergence topic.

To ease the presentation we deal only with the discrete time case in section “[Set Valued and Conditional Probability Valued Observers](#)”, and the continuous time case in sections

“[An Optimization Approach](#)” and “[Convergent Observers](#)”.

Observation Problem and Its Solutions

The Observation Problem

Let $X^{\delta^s}(x, t, s)$, respectively $X_l^{\delta^s}(x, k)$, denote a solution of (1) at time s , respectively l , going through x at time t , respectively k , and under the action of δ^s .

Observation problem *At each time t , respectively k , given the function $s \in]t - T, t] \mapsto y(s)$, respectively the sequence $l \in \{k - K, \dots, k\} \mapsto y_l$, find an estimation $\hat{z}(t)$, respectively $\hat{z}_k$, of $z(t)$, respectively z_k , satisfying:*

$$\begin{aligned} \hat{z}(t) &= q(\hat{x}(t), t, \delta^h(t)) \quad \text{resp.} \\ \hat{z}_k &= q_k(\hat{x}_k, \delta^h). \end{aligned}$$

where $\hat{x}(t)$, respectively $\hat{x}_k$, is to be found as a solution of:

$$\begin{aligned} \hat{x}(t) &\in \mathcal{X}(t), \quad y(s) = h(X^{\delta^s}(\hat{x}(t), t, s), s, \delta^m(s)) \\ \forall s &\in]t - T, t], \end{aligned}$$

respectively

$$\begin{aligned} \hat{x}_k &\in \mathcal{X}_k, \quad y_l = h_l(X_l^{\delta^s}(\hat{x}_k, k), \delta_l^m) \\ \forall l &\in \{k - K, \dots, k\} \end{aligned}$$

and where the time functions δ^s, δ^m and δ^h must agree with the a priori (deterministic/stochastic) information or minimized in some way.

In this statement T , respectively K , quantify the time window length or memory length during which we record the measurement. The accumulation with time of measurements, together with the model equations (1) to (3) and the assumptions on ($\delta^s, \delta^m, \delta^h$), gives a redundancy of data compared with the number of unknowns that the hidden variables are. This is why it may be possible to solve this observation problem.

To simplify the following presentation, we restrict our attention on the case where the hidden variables are actually the full model state, i.e.,

$$z = \eta(x) = x.$$

When z differs from x , observers are called functional observers.

Set Valued and Conditional Probability Valued Observers

Conceptually the answer to this problem is easy at least when the memory increases with time ($\dot{T}(t) = 1$ resp. $K_{k+1} = K_k + 1$) leading to an infinite non-fading memory. It consists in starting from all what the a priori information makes possible and to eliminate what is not consistent with the a posteriori information. In the set valued observer setting, in the discrete time case, this gives the following observer. To ease its reading, we underline the data given by the a priori information. It requires the introduction of two sets ξ_k and $\xi_{k|k-1}$ which are updated at each time k when a new measurement y_k is made available. ξ_k is the set which x_k is guaranteed to belong to at time k , knowing all the measurements up to time k , and $\xi_{k|k-1}$ is the same but with measurements known up to time $k - 1$.

Set valued observer:

Initialization:

$$\xi_0 = \underline{X}_0$$

At each time k :

prediction (flowing)

$$\xi_{k|k-1} = \underline{f}_{k-1}(\xi_{k-1}, \underline{\Delta}_{k-1}^s)$$

restriction (consistency)

$$\xi_k = \left\{ x \in \left(\xi_{k|k-1} \cap \underline{X}_k \right) : y_k \in \underline{h}_k(x, \underline{\Delta}_k^m) \right\}$$

estimation

$$\widehat{x}_k \in \xi_k$$

A key feature here is that *this observer has a state ξ_k – a set – and is a dynamical system in the form:*

$$\xi_{k+1} = \varphi_k(\xi_k, y_k), \quad \widehat{x}_k \in \xi_k$$

with y as input and $\widehat{x}$ as output which is not single valued. Important also, the initial condition of the state ξ is given by the a priori information.

In the stochastic setting, following the Bayesian paradigm, the observer has the same structure but with the state ξ_k being a conditional probability. See (Jazwinski 2007, Theorem 6.4) or (Candy 2009, Table 2.1). In that setting too the observer is not a single state; it is the (a posteriori) conditional probability of the random variable x_k given the a priori information and the sequence of measurements $l \in \{k - K, \dots, k\} \mapsto y_l$.

Comments

Implementation: For the time being, except for very specific cases (Kalman filter, ...), the set valued and the conditional probability valued observers remain conceptual since we do not know how to manipulate numerically sets and probability laws. Their implementation requires approximations. For instance, see Milanese et al. (1996) and Witsenhausen (1966) for the set case and (Arulampalam et al. 2002); (Bucy and Joseph 1987); (Candy 2009), and (Jazwinski 2007) for the conditional probability case.

Need of finite or infinite but fading memory:

In these observers, model states x which are consistent with the a priori information but do not agree with the a posteriori information are eliminated (set intersection or probability product). But once a point is eliminated, this is for ever. As a consequence if there is, at some time, a misfit between a priori and a posteriori information, it is mistakenly propagated in future times. A way to round this problem is to keep the information memory finite or infinite but fading. In particular, with fixed length memory, consistent points which were disregarded due to measurements which are no more in the memory are reintroduced. This

says also that observers should not be sensitive to their initial condition.

Not single-valued estimate. The observers introduced above realize a lossless data compression with extracting and preserving all what concerns the hidden variables in the redundant data given by a priori and a posteriori information. But this “lossless compression” answer is not single valued (set valued or conditional probability valued) as a result of taking uncertainties into account. Actually, to get a single valued answer, *the observation problem must be complemented by making precise for what the estimation is made.* For instance, we may want to select the most likely or the average or more generally some cost-minimizing estimate $\hat{x}$ among all the possible ones given by ξ . In this way we obtain an observer giving a single-valued estimate:

$$\xi_{k+1} = \varphi_k(\xi_k, y_k), \quad \hat{x}_k = \tau_k(\xi_k)$$

respectively

$$\begin{aligned} \dot{\xi}(t) &= \varphi(\xi(t), y(t), t), \\ \hat{x}(t) &= \tau(\xi(t), t) \end{aligned} \quad (6)$$

But then, in general, we lose information and in particular we have no idea on the confidence level this estimate has. Also, since the function τ , at least, encodes for what the estimate $\hat{x}$ is used, for different uses, different functions τ may be needed.

An Optimization Approach

A shortcut to obtain directly an observer giving a single-valued estimate is to design it by trading off among a priori and a posteriori information (Cox 1964, pages 7–10), Alamir (2007),...). For example, in the continuous time case, we can select the estimate $\hat{x}(t)$ among the minimizers (in x) of:

$$C(\{s \mapsto \delta^s(s)\}, x, t) = \int_{-\infty}^t C(\delta^s(s), y(s), X^{\delta^s}(x, t, s), s) ds$$

where $X^{\delta^s}(x, t, s)$ is still the notation for a solution to (1) and $\{s \mapsto \delta^s(s)\}$, representing the unmodeled effect on the dynamics, is among the arguments for the minimization. The infinitesimal cost C is chosen to take nonnegative values and be such that $C(0, h(x, s), x, s)$ is zero. For instance, it can be:

$$C(\delta^s, y, x, s) = \|\delta^s\|_x^2 + d_y(y, h(x, s))^2$$

where $\|\cdot\|_x$ is a norm at the point x and d_y is a distance in the measurement space. In the same spirit, instead of optimization, a minimax approach can be followed. See, for instance, Bertsekas and Rhodes (1971), Başar (1995, Chapter 7), and Willems (2004).

With x fixed, the minimization of C is an infinite horizon optimal control problem in reverse time. Solving on line this problem is extremely difficult and again approximations are needed. We do not go on with this approach, but we remark that, under extra assumptions, the observer we obtain following this approach can also be implemented in the form of a dynamical system (6) but with the specificity that *the estimate $\hat{x}$ is part of the observer state ξ and its dynamics are a copy of the undisturbed model with a correction term which is zero when the estimated state reproduce the measurement.* Namely, we get:

$$\begin{aligned} \dot{\hat{x}}(t) &= f(\hat{x}(t), t, 0) + E(\{\sigma \mapsto y(\sigma)\}, \\ &\quad \hat{x}(t), y(t), t) \end{aligned}$$

where E is zero when $h(\hat{x}(t), t) = y(t)$. But, as opposed to what we saw in the previous section, the initial condition for the part $\hat{x}$ of the observer state is unknown. Hence we encounter again the need for the observer to forget its initial condition.

Convergent Observers

We have mentioned that often an observer can be implemented as a dynamical system, but without

knowing necessarily how to initialize it. Also approximation is involved both in its design and its implementation. So, at least when it gives a single valued estimate, we are facing the problem of convergence of this estimate to the “true” value, at least when there is no uncertainties. We concentrate now our attention on the study of this convergence, but, to simplify, in the continuous time case only.

Let the model and observer dynamics be:

$$\dot{x}(t) = f(x(t), t), \quad y(t) = h(x(t), t) \quad (7)$$

$$\dot{\xi} = \varphi(\xi(t), y(t), t), \quad \hat{x}(t) = \tau(\xi(t), y(t), t) \quad (8)$$

with the observer state ξ of finite dimension m . We denote by $(X(x, t, s), \Xi((x, \xi), t, s))$ a solution of (7)–(8).

Since we are dealing with convergence, the focus is on what is going on when the time becomes very large and in particular on the set Ω of model states which are accumulation points of some solution. Specifically we are interested in the stability properties of the set:

$$\mathfrak{Z}(t) = \left\{ (x, \xi) : x \in \Omega \text{ \& } x = \tau(\xi, h(x, t), t) \right\}$$

which is contained in the zero estimation error set associated with the given model-observer pair.

Definition 1 (Convergent observer) We say the observer (8) is convergent if, for each t , there exists a set $\mathfrak{Z}_a(t) \subset \mathfrak{Z}(t)$, such that, on the domain of existence of the solution, a distance between the point $(X(x, t, s), \Xi((x, \xi), t, s))$ and the set $\mathfrak{Z}_a(t)$ is upperbounded by a real function $s \mapsto \beta_{x, \xi, t}^c(s)$, maybe dependent on (x, ξ, t) , with nonnegative values, strictly decreasing and going to zero as s goes to infinity.

Necessary Conditions for Observer Convergence

No Restriction on τ

It is possible to prove that, *if the observer is convergent then,*

Necessity of detectability: When h and τ are uniformly continuous in x and ξ , respectively, the estimate $\hat{x}$ does converge to the model state x . In this case, two solutions of the model (7) which produce the same measurement must converge to each other. This is an asymptotic distinguishability property called detectability. If we are interested, not only in the asymptotic behavior but also in the transient (as for output feedback), a property stronger than detectability is needed. In particular instantaneous distinguishability (see section “[Observers Based on Instantaneous Distinguishability](#)”) is necessary if we want to be able to impose the decay rate of the function $\beta_{x, \xi, t}^c$.

Necessity of $m \geq n - p$: For each t , there exists a subset $X_a(t)$ of Ω , supposed to collect the model states which can be asymptotically estimated, and such that we can associate, to each of its point x , a set $\tau^i(x, t)$ allowing us to redefine the set $\mathfrak{Z}_a(t)$ as:

$$\mathfrak{Z}_a(t) = \left\{ (x, \xi) : x \in X_a(t) \text{ \& } \xi \in \tau^i(x, t) \right\}.$$

This implies that, for each t and each x in $X_a(t)$, there is a point ξ satisfying:

$$x = \tau(\xi, h(x, t), t). \quad (9)$$

This is a surjectivity property of the function τ but of a special kind since $h(x, t)$ is an argument of τ . We say that, for each t , *the function τ is surjective to $X_a(t)$ given h* . In a “generic” situation, this property requires the dimension m of the observer state ξ to be larger or equal to the dimension n of the model state x minus the dimension p of the measurement y .

τ Is Injective Given h

We consider now the case where the observer has been designed with a function τ which is injective given h , namely, we have the following implication, when x is in $X_a(t)$:

$$\left[\begin{array}{l} \tau(\xi_1, h(x, t), t) = \tau(\xi_2, h(x, t), t) \quad \& \\ \xi_1 \in \tau^i(x, t) \end{array} \right] \implies \xi_1 = \xi_2.$$

In a “generic” situation, this property, together with the surjectivity given h , implies that the

dimension m of the observer state ξ should be between $n - p$ and n .

If a convergent observer has a such a function τ , then $(x, t) \mapsto \tau^i(x, t)$, which is (of course) a (single valued) function, admits a Lie derivative $L_f \tau^i$ satisfying:

$$\begin{aligned} L_f \tau^i(x, t) &= \lim_{dt \rightarrow 0} \frac{\tau^i(X(x, t, t + dt), t + dt) - \tau^i(x, t)}{dt}, \\ &= \varphi(\tau^i(x, t), h(x, t), t) \quad \forall x \in \mathcal{X}_a(t). \end{aligned} \quad (10)$$

This says (very approximatively) that φ is nothing but the image of the vector field f , under the change of coordinates $(x, t) \mapsto (\tau^i(x, t), t)$ but again all this given h . As partly obtained in the optimization approach, the observer dynamics are then a copy of the model dynamics with maybe a correction term which is zero when the estimated state reproduce the measurement.

If moreover the functions h and τ are uniformly continuous in x and ξ , respectively, then, given ξ_1 and ξ_2 , a distance between $\Xi((x, \xi_1), t, s)$ and $\Xi((x, \xi_2), t, s)$ goes to zero as s goes to infinity. This property is related to what was called extreme stability (see Yoshizawa (1966)) in the 1950s and 1960s and is called incremental stability today (see Angeli (2002)). It holds when, with denoting by $\Xi^y(\xi, t, s)$ the solution at time s of the observer dynamics:

$$\dot{\xi}(t) = \varphi(\xi(t), y(t), t)$$

going through ξ at time t and under the action of y , the flow $\xi \mapsto \Xi^y(\xi, t, s)$ is a strict contraction (See Jouffroy (2005) for a bibliography on contraction) for each $s > t$ or, at least, if a distance between any two solutions $\Xi^y(\xi_1, t, s)$ and $\Xi^y(\xi_2, t, s)$, with the same input y , converges to 0.

Sufficient Conditions

Knowing now how a convergent observer should look like, we move to a quick description of some such observers.

Observers Based on Contraction

Since the flow generated by the observer should be a contraction, we may start its design by picking the function φ as:

$$\dot{\xi}(t) = \varphi(\xi(t), y(t), t) = A \xi(t) + B(y(t), t)$$

where A , not related to f , is a matrix whose eigen values have strictly negative real part. Under weak restriction, there exists a function τ^i satisfying (10), namely:

$$L_f \tau^i(x, t) = A \tau^i(x, t) + B(h(x, t), t). \quad (11)$$

To obtain a convergent observer, it is then sufficient that there exists a (uniformly continuous) function τ satisfying:

$$x = \tau(\tau^i(x, t), h(x, t), t)$$

For this to be possible, the function τ^i should be injective given h . This injectivity holds when the observer state has dimension $m \geq 2(n + 1)$, the model is distinguishable, and provided the eigen values of A have a sufficiently negative real part and are not in a set of zero Lebesgue measure.

Unfortunately, we are facing again a possible difficulty in the implementation since an expression for a function τ^i satisfying (11) is needed and the function $\tau : (\xi, y, t) \mapsto \hat{x}(t)$ is known implicitly only as:

$$\xi = \tau^i(\hat{x}(t), t).$$

See Andrieu and Praly (2006), Luenberger (1964), and Shoshitaishvili (1990).

Observers Based on Instantaneous

Distinguishability

Instantaneous distinguishability means that we can distinguish as quickly as we want two model states by looking at the paths of the measurements they generate. A sufficient condition to have this property can be obtained by looking at the Taylor expansion in s of $h(X(x, t, s), s)$. Indeed, we have:

$$h(X(x, t, s), s) = \sum_{i=0}^{m-1} h_i(x, t) \frac{(s-t)^i}{i!} + o((s-t)^{m-1})$$

where h_i is a function obtained recursively as:

$$\begin{aligned} h_0(x, t) &= h(x, t) \quad h_{i+1}(x, t) = \overline{\dot{h}_i(x, t)} \\ &= \frac{\partial h_i}{\partial x}(x, t) f(x, t) + \frac{\partial h_i}{\partial t}(x, t). \end{aligned}$$

If there exists an integer m such that, in some uniform way with respect to t , the function

$$x \mapsto H_m(x, t) = (h_0(x, t), \dots, h_{m-1}(x, t))$$

is injective, then we do have instantaneous distinguishability. We say the system is differentially observable of order m when this injectivity property holds. When a system has such a property, the model state space has a very specific structure as discussed in Isidori (1995, Section 1.9). It means that we can reconstruct x from the knowledge of y and its $m-1$ first time derivatives, i.e., there exists a function Φ such that we have:

$$x = \Phi(H_m(x, t), t).$$

This way, we are left with estimating the derivatives of y . This can be done as follows. With the notation $\eta_i = h_{i-1}(x, t)$, we obtain:

$$\dot{\eta}(t) = F \eta + G h_m(\Phi(\eta(t), t), t)$$

where

$$F \eta = (\eta_2, \dots, \eta_m, 0), \quad G = (0, \dots, 0, 1).$$

When the last term on the right-hand side is Lipschitz, we can find a convergent observer in the form:

$$\begin{aligned} \dot{\hat{\xi}}(t) &= F \hat{\xi}(t) + G h_m(\hat{x}(t), t) \\ &\quad + K(y(t) - \xi_1(t)), \quad \hat{x}(t) = \tau(\hat{\xi}(t), t), \end{aligned}$$

with $\hat{\xi}$ being actually an estimation of η and where K is a constant matrix and τ is a modified version of Φ keeping the estimated state in its a priori given set $\mathcal{X}(t)$.

This is the high-gain observer paradigm. See Gauthier and Kupka (2001) and Tornambe (1988). The implementation difficulty is in the function $\hat{\Phi}$, not to mention sensitivity to measurement uncertainty.

Observers With τ Bijective Given h

Case Where τ Is the Identity Function

A convergent observer whose function τ is the identity has the following form:

$$\begin{aligned} \dot{\hat{\xi}} &= f(\hat{\xi}, t) + E(\{\sigma \mapsto y(\sigma)\}, \hat{\xi}(t), y(t), t), \\ \hat{x}(t) &= \hat{\xi}(t). \end{aligned} \tag{12}$$

The only piece remaining to be designed is the correction term E . It has to ensure convergence and may be also other properties like symmetry preserving (see Bonnabel et al. 2008).

For this design, a first step is to exhibit some specific properties of the vector field f by writing it in some appropriate coordinates. For example, there may exist coordinates such that the expression of f takes the form $\hat{f}(x(t), h(x, t), t)$ and the corresponding observer (12) is such that there exists a positive definite matrix P for which the function $s \mapsto (X(x, t, d) - \hat{X}((x, \hat{x}), t, s))' P (X(x, t, d) - \hat{X}((x, \hat{x}), t, s))$ is strictly decaying (if not zero). A necessary condition for this to be possible is that f is

monotonic tangentially to the level sets of the function h , i.e., for all (x, y, v, t) satisfying $y = h(x, t)$ and $\frac{\partial h}{\partial x}(x, t)v = 0$, we have:

$$v^T P \frac{\partial f}{\partial x}(x, y, t) v \leq 0. \quad (13)$$

This is another way of expressing a detectability condition. This expression is coordinate dependent, hence the importance of choosing the coordinates properly.

When this condition is strict and uniform in t , it is sufficient to get a locally convergent observer and even a nonlocal one when h is linear in x , i.e., $h(x, t) = H(t)x$, again a coordinate-dependent condition. In this latter case the observer takes the form:

$$\begin{aligned} \dot{\xi}(t) &= f(\xi(t), y(t), t) + \ell(\xi(t)) P^{-1} H(t)^T \\ &\quad [y(t) - H(t)\xi(t)], \quad \hat{x}(t) = \xi(t), \end{aligned}$$

where ℓ is a real function to be chosen with sufficiently large values. If (13) is strict and uniform and holds for all v , the correction term is not needed.

There are many other results of this type, exploiting one or the other specificity of the dependence on x of the function f – monotonicity, convexity, See Fan and Arcak (2003), Krener and Isidori (1983), Respondek et al. (2004) and Sanfelice (2012), ...

Case Where $(x, t) \mapsto (\tau^i(x, t), h(x, t), t)$ Is a Diffeomorphism

At each time t we know already that the model state x we want to estimate satisfies $y(t) = h(x, t)$. So, as remarked in Luenberger (1964), when $(h(x, t), t)$ can be used as part of coordinates for (x, t) , we need to estimate the remaining part only. This can be done if we find a function τ^i , whose values are $n - p$ dimensional, such that $(x, t) \mapsto (y, \eta, t) = (h(x, t), \tau^i(x, t), t)$ is a diffeomorphism and the flow $\eta \mapsto \eta^y(\eta, t, s)$ generated by

$$\begin{aligned} \dot{\eta}(t) &= \frac{\partial \tau^i}{\partial x}(x(t), t) f(x(t), t) + \frac{\partial \tau^i}{\partial t}(x(t), t), \\ &= \varphi(\eta(t), y(t), t) \end{aligned}$$

is a strict contraction for all $s > t$. Indeed in this case the observer dynamics can be chosen as:

$$\dot{\xi}(t) = \varphi(\xi(t), y(t), t)$$

and the estimate $\hat{x}(t)$ is obtained as solution of:

$$\tau^i(\hat{x}(t), t) = \xi(t), \quad h(\hat{x}(t), t) = y(t).$$

This is the reduced order observer paradigm. See, for instance, Besançon (2000, Proposition 3.2), and Carnevale et al. (2008), and Luenberger (1964, Theorem 4).

Cross-References

- [Observer-Based Control](#)
- [Observers in Linear Systems Theory](#)

Bibliography

- Alamir M (2007) Nonlinear moving horizon observers: theory and real-time implementation. In: Nonlinear observers and applications. Lecture notes in control and information sciences. Springer, Berlin/Heidelberg
- Andrieu V, Praly L (2006) On the existence of Kazantzis-Kravaris/Luenberger observers. *SIAM J Control Optim* 45(2):432–456
- Angeli D (2002) A Lyapunov approach to incremental stability properties. *IEEE Trans Autom Control* 47(3):410–421
- Arulampalam M, Maskell S, Gordon N, Clapp T (2002) A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans Signal Process* 50(2):174–188
- Başar T, Bernhard P (1995) H^∞ optimal control and related minimax design problems: a dynamic game approach, Revised 2nd edn. Birkhäuser, Boston
- Besançon G (2000) Remarks on nonlinear adaptive observer design. *Syst Control Lett* 41(4):271–280
- Bertsekas DI, Rhodes IB (1971) On the minimax reachability of target sets and target tubes. *Automatica* 7: 233–213
- Bonnabel S, Martin P, Rouchon P (2008) Symmetry-preserving observers. *IEEE Trans Autom Control* 53(11):2514–2526

- Bucy R, Joseph P (1987) Filtering for stochastic processes with applications to guidance, 2nd edn. Chelsea Publishing Company, New York
- Candy J (2009) Bayesian signal processing: classical, modern, and particle filtering methods. Wiley series in adaptive learning systems for signal processing, communications and control. Wiley, Hoboken
- Carnevale D, Karagiannis D, Astolfi A (2008) Invariant manifold based reduced-order observer design for nonlinear systems. *IEEE Trans Autom Control* 53(11):2602–2614
- Cox H (1964) On the estimation of state variables and parameters for noisy dynamic systems. *IEEE Trans Autom Control* 9(1):5–12
- Fan X, Arcak M (2003) Observer design for systems with multivariable monotone nonlinearities. *Syst Control Lett* 50:319–330
- Gauthier J-P, Kupka I (2001) Deterministic observation theory and applications. Cambridge University Press, Cambridge
- Isidori A (1995) Nonlinear control systems, 3rd edn. Springer, London
- Jazwinski A (2007) Stochastic processes and filtering theory. Dover Publications, Mineola
- Jouffroy J (2005) Some ancestors of contraction analysis. *Proc IEEE Conf Decis Control* 6:5450–5455
- Krener A, Isidori A (1983) Linearization by output injection and nonlinear observer. *Syst Control Lett* 3(1):47–52
- Luenberger D (1964) Observing the state of a linear system. *IEEE Trans Mil Electron MIL-8*:74–80
- Milanese M, Norton J, Piet-Lahanier H, Walter E (eds) (1996) Bounding approaches to system identification. Plenum Press, New York
- Respondek W, Pogromsky A, Nijmeijer H (2004) Time scaling for observer design with linearizable error dynamics. *Automatica* 40:277–285
- Sanfelice R, Praly L (2012) Convergence of nonlinear observers on with a riemannian metric (Part I). *IEEE Trans Autom Control* 57(7):1709–1722
- Shoshitaishvili A (1990) Singularities for projections of integral manifolds with applications to control and observation problems. In: Arnol'd VI (ed) *Advances in soviet mathematics*, vol 1. Theory of singularities and its applications. American Mathematical Society, Providence
- Tornambe A (1988) Use of asymptotic observers having high gains in the state and parameter estimation. In: *Proceedings of the IEEE conference on decision and control*, 1791–1794
- Willems JC (2004) Deterministic least squares Filtering. *J Econ* 118:341–373
- Witsenhausen H (1966) Minimax control of uncertain systems. *Elec Syst Lab M.I.T. Rep. ESL-R-269M*, Cambridge
- Yoshizawa T (1966) Stability theory by Lyapunov's second method. The mathematical society of Japan

Observers in Linear Systems Theory

Alessandro Astolfi^{1,2} and Panos J. Antsaklis³

¹Department of Electrical and Electronic Engineering, Imperial College London, London, UK

²Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma Tor Vergata, Rome, Italy

³Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

Observers are dynamical systems which process the input and output signals of a given dynamical system and deliver an online estimate of the internal state of the system which asymptotically converges to the exact value of the state. For linear, finite-dimensional, time-invariant systems, observers can be designed provided a weak observability property, known as detectability, holds.

Keywords

State observer · Reduced order observer · Detectability

Introduction

Consider a linear, finite-dimensional, time-invariant system described by equations of the form

$$\begin{aligned}\sigma x &= Ax + Bu, \\ y &= Cx + Du,\end{aligned}\tag{1}$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$, $t \in T \subset \mathbb{R}$, and A , B , C , and D matrices of appropriate dimensions and with constant entries and the problem of estimating its state from

measurements of the input and output signals. In Eq. (1) $\sigma x(t)$ stands for $\dot{x}(t)$, if the system is continuous-time, and for $x(t+1)$, if the system is discrete-time. In addition, if the system is continuous-time, then $T \in \mathbb{R}^+$, i.e., the set of nonnegative real numbers, whereas if the system is discrete-time, then $T \in \mathbb{Z}^+$, i.e., the set of nonnegative integers.

We are interested in determining an online estimate $x_e(t) \in \mathbb{R}^n$, i.e., the estimate at time t has to be a function of the available information (input and output) at the same time instant. This implies that the estimate is generated by means of a device (known as a *filter*) processing the current input and output of the system and generating a state estimate. The filter may be instantaneous, i.e., the estimate is generated instantaneously by processing the available information. In this case we have a static filter. Alternatively, the state estimate can be generated processing the available information through a dynamical device. In this case we have a dynamic filter.

Assume, for simplicity, that $D = 0$. This assumption is without loss of generality: if $y = Cx + Du$ and u are measured, then also $\tilde{y} = Cx$ is measured. Assume, in addition, that the filter which generates the online estimate is linear, finite-dimensional, and time-invariant. Then we may have the following two configurations.

- *Static filter.* The state estimate is generated via the relation

$$x_e = My + Nu, \quad (2)$$

with M and N constant matrices of appropriate dimensions. The resulting interconnected system is described by the equations

$$\begin{aligned} \sigma x &= Ax + Bu, \\ x_e &= MCx + Nu. \end{aligned} \quad (3)$$

- *Dynamic filter.* The state estimate is generated by the system

$$\begin{aligned} \sigma \xi &= F\xi + Ly + Hu, \\ x_e &= M\xi + Ny + Pu, \end{aligned} \quad (4)$$

with F, L, H, M, N and P constant matrices of appropriate dimensions. The resulting interconnected system is described by the equations

$$\begin{aligned} \sigma x &= Ax + Bu, \\ \sigma \xi &= F\xi + LCx + Hu, \\ x_e &= M\xi + NCx + Pu. \end{aligned} \quad (5)$$

In what follows, we study in detail the dynamic filter configuration. This is mainly due to the fact that this configuration allows solving most estimation problems for linear systems. Moreover, while the use of a static filter is very appealing, it provides a useful alternative only in very specific situations.

State Observer

A state observer is a filter that allows to estimate, asymptotically or in finite time, the state of a system from measurements of the input and output signals.

The simplest possible observer can be constructed considering a copy of the system, the state of which has to be estimated. This means that a candidate observer for system (1) is given by

$$\begin{aligned} \sigma \xi &= A\xi + Bu, \\ x_e &= \xi. \end{aligned} \quad (6)$$

To assess the properties of this candidate state observer, let $e = x - x_e$ be the estimation error, and note that $\sigma e = Ae$. As a result, if $e(0) = 0$ then $e(t) = 0$ for all t and for any input signal u . However, if $e(0) \neq 0$, then, for any input signal u , $e(t)$ is bounded only if the system (1) is stable and converges to zero only if the system (1) is asymptotically stable. If these conditions do not hold, the estimation error is not bounded, and system (6) does not qualify as a state observer for system (1). The intrinsic limitation of the observer (6) is that it does not use all the available information, i.e., it does not use the knowledge of the output signal y . This observer is therefore an open-loop observer.

To exploit the knowledge of y we modify the observer (6) adding a term which depends upon the available information on the estimation error, which is given by $y_e = Cx_e - y$. This modification yields a candidate state observer described by

$$\begin{aligned}\sigma \xi &= A\xi + Bu + Ly_e, \\ x_e &= \xi.\end{aligned}\quad (7)$$

To assess the properties of this candidate state observer note that $e = x - x_e$ is such that

$$\sigma e = (A + LC)e. \quad (8)$$

The matrix L (known as *output injection gain*) can be used to shape the dynamics of the estimation error. In particular, we may select L to assign the characteristic polynomial $p(s)$ of $A + LC$. To this end, note that (the notation M' is used to denote the transpose of the matrix M)

$$p(s) = \det(sI - (A + LC)) = \det(sI - (A' + C'L')).$$

Hence, there is a matrix L which arbitrarily assigns the characteristic polynomial of $A + LC$ if and only if the system

$$\sigma \xi = A'\xi + C'v$$

is reachable or, equivalently, if and only if the system (1) is observable.

We summarize the above discussion with two formal statements.

Proposition 1 *Consider system (1) and suppose the system is observable. Let $p(s)$ be a monic polynomial of degree n . Then there is a matrix L such that the characteristic polynomial of $A + LC$ is equal to $p(s)$.*

Note that for single-output systems, the matrix L assigning the characteristic polynomial of $A + LC$ is unique.

Proposition 2 *System (1) is observable if and only if it is possible to arbitrarily assign the eigenvalues of $A + LC$.*

Detectability

The main goal of a state observer is to provide an online estimate of the state of a system. This goal may be achieved, as discussed in the previous section, if the system is observable. However, observability is not necessary to achieve this goal: in fact the unobservable modes (that is the eigenvalues of the unobservable part of the system) are not modified by the output injection gain. This implies that there exists a matrix L such that system (8) is asymptotically stable if and only if the unobservable modes of system (1) have negative real part, in the case of continuous-time systems, or have modulus smaller than one, in the case of discrete-time systems. To capture this situation we introduce a new definition.

Definition 1 (Detectability) System (1) is detectable if its unobservable modes have negative real part, in the case of continuous-time systems, or have modulus smaller than one, in the case of discrete-time systems.

Example (Dead-beat observer) Consider a discrete-time system described by equations of the form

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t),\end{aligned}$$

and the problem of designing a state observer, described by the Eqs. (7), such that, for any initial condition $x(0)$ and for any input u , $e(k) = 0$, for all $k \geq N$, and for some $N > 0$. A state observer achieving this goal is called a dead-beat state observer. To achieve this goal it is necessary to select L such that $(A + LC)^N = 0$ or, equivalently, such that the matrix $A + LC$ has all eigenvalues equal to 0. Note that $N \leq n$. $\diamond$

Reduced Order Observer

We have shown that, under the hypotheses of observability or detectability, it is possible to design an asymptotic observer of order n for the

system (1). However, this observer is somewhat oversized, i.e., it gives an estimate for the n components of the state vector, without making use of the fact that some of these components can be directly determined from the output function, e.g., if $y = x_1$ there is no need to reconstruct x_1 . It makes sense, therefore, to design a *reduced order observer*, i.e., a device that estimates only the part of the state vector which is not directly attainable from the output. To this end, consider the system (1) with $D = 0$, and assume that the matrix C has p independent rows. This is without loss of generality: in fact this is the case if $\text{rank } C = p$, whereas if $\text{rank } C < p$, it is always possible to eliminate redundant rows from C . Then there exists a matrix Q such that, possibly after reordering the state variables,

$$QC = [I \ C_2].$$

Let

$$v = Qy = QCx = x_1 + C_2x_2,$$

in which $x_1(t) \in \mathbb{R}^p$ and $x_2(t) \in \mathbb{R}^{n-p}$ denote the first p and the last $n - p$ components of x .

Observe that the vector v is measured. From the definition of v we conclude that if v and x_2 are known then x_1 can be easily computed, i.e., there is no need to construct an observer for x_1 .

Define now the new coordinates

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = Tx = \begin{bmatrix} I & C_2 \\ 0 & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

and note that, by construction, $v = Qy = \hat{x}_1$. In the new coordinates the system, with output v , is described by equations of the form

$$\begin{aligned} \sigma \hat{x}_1 &= \tilde{A}_{11}\hat{x}_1 + \tilde{A}_{12}\hat{x}_2 + \tilde{B}_1u, \\ \sigma \hat{x}_2 &= \tilde{A}_{21}\hat{x}_1 + \tilde{A}_{22}\hat{x}_2 + \tilde{B}_2u, \\ v &= \hat{x}_1. \end{aligned}$$

To construct an observer for $\hat{x}_2$ consider the system

$$\sigma \xi = F\xi + Hv + Gu,$$

with state ξ , driven by u and v , and with output

$$w = \xi + Lv.$$

The idea is to select the matrices F , H , G , and L in such a way that w is an estimate for $\hat{x}_2$. Let $w - \hat{x}_2$ be the observation error. Then

$$\begin{aligned} \sigma w - \sigma \hat{x}_2 &= F\xi + Hv + Gu + L \left[\tilde{A}_{11}\hat{x}_1 + \tilde{A}_{12}\hat{x}_2 + \tilde{B}_1u \right] - \left[\tilde{A}_{21}\hat{x}_1 + \tilde{A}_{22}\hat{x}_2 + \tilde{B}_2u \right] \\ &= F\xi + \left(H + L\tilde{A}_{11} - \tilde{A}_{12} \right) \hat{x}_1 + \left[L\tilde{A}_{12} - \tilde{A}_{22} \right] \hat{x}_2 + \left[G + L\tilde{B}_1 - \tilde{B}_2 \right] u. \end{aligned} \quad (9)$$

To have convergence of the estimation error to zero, regardless of the initial conditions and of the input signal, we must have

$$\sigma(w - \hat{x}_2) = F(w - \hat{x}_2) \quad (10)$$

and F must have all eigenvalues with negative real part, in the case of continuous-time systems, or with modulus smaller than one, in the case of discrete-time systems. Comparing Eqs. (9) and

(10), we obtain that the matrices F , H , G , and L must be such that

$$\begin{aligned} L\tilde{A}_{12} - \tilde{A}_{22} &= -F, \\ H + L\tilde{A}_{11} - \tilde{A}_{21} &= FL, \\ G + L\tilde{B}_1 - \tilde{B}_2 &= 0. \end{aligned} \quad (11)$$

We now show how the Eq. (11) can be solved and how the stability condition on F can be

enforced. Detectability of the system implies that the (reduced system) $\sigma \tilde{\xi} = \tilde{A}_{22} \tilde{\xi}$ with output $\tilde{y} = \tilde{A}_{12} \tilde{\xi}$ is detectable. As a result, there exists a matrix L such that the matrix

$$F = \tilde{A}_{22} - L \tilde{A}_{12}$$

has all eigenvalues with negative real part, in the case of continuous-time systems, or with modulus smaller than one, in the case of discrete-time systems. Then the remaining equations are solved by

$$\begin{aligned} H &= FL - L \tilde{A}_{11} + \tilde{A}_{21}, \\ G &= -L \tilde{B}_1 + \tilde{B}_2. \end{aligned}$$

Finally, from $\hat{x}_1 = v$ and the estimate w of $\hat{x}_2$, we build an estimate x_e of the state x inverting the transformation T , i.e.,

$$\begin{bmatrix} x_{1e} \\ x_{2e} \end{bmatrix} = \begin{bmatrix} I & -C_2 \\ 0 & I \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}.$$

Summary and Future Directions

The problem of online estimating the state of a linear system from input and output measurements can be solved provided a weak observability condition holds. The problem addressed in this entry is the simplest possible estimation problem: the underlying system is linear, and all variables are exactly measured. Observers for nonlinear systems and in the presence of signals corrupted by noise can also be designed exploiting some of the basic ingredients, such as the notions of error system and of output injection, discussed in this entry.

Cross-References

- [Controllability and Observability](#)
- [Hybrid Observers](#)
- [Kalman Filters](#)

- [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)
- [Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions](#)
- [Networked Control Systems: Estimation and Control over Lossy Networks](#)
- [Observer-Based Control](#)
- [Observers for Nonlinear Systems](#)
- [State Estimation for Batch Processes](#)

Recommended Reading and References

Classical references on observers for linear systems are given below.

Bibliography

- Antsaklis PJ, Michel AN (2007) A linear systems primer. Birkhäuser, Boston
- Brockett RW (1970) Finite dimensional linear systems. Wiley, London
- Brockett RW (2015) Finite dimensional linear systems. SIAM, Cambridge
- Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs
- Luenberger DG (1963) Observing the state of a linear system. IEEE Trans Mil Electron 8:74–80
- Trentelman HL, Stoorvogel AA, Hautus MLJ (2001) Control theory for linear systems. Springer, London
- Zadeh LA, Desoer CA (1963) Linear system theory. McGraw-Hill, New York

Opacity of Discrete Event Systems

Christoforos N. Hadjicostis
Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus

Abstract

We discuss ways to formulate, analyze, verify, and enforce different notions of opacity in discrete event systems. Opacity captures a class

of information flow properties that are important for security/privacy analysis and focus on the type of inferences that a passive intruder can make regarding a subset of the behavior of the system, called the secret behavior. The secret behavior represents critical aspects of the given system that need to be kept hidden from the passive intruder, which is modeled as an external observer with (partial) knowledge of the system model and access to certain observed system behavior.

Keywords

Discrete event systems · Opacity · Privacy · Security · Finite automata · Petri nets · Verification

Introduction

Opacity is an important security/privacy notion that has emerged as a major research direction in discrete event systems (DES) in the last two decades. Given an underlying DES, opacity aims to characterize the type of inferences that a passive observer can make regarding a subset of the behavior of the system that is considered critical and is referred to as the *secret behavior*. The secret behavior models activity in the system that reveals important details about the operation of the system and needs to be kept hidden, whereas the passive observer captures an external entity (intruder) that cannot issue commands to influence the operation of the system but can nevertheless observe (perhaps partially) activity in the system. By combining observations resulting from activity in the system together with its knowledge of the system model (e.g., partial knowledge of the transition structure and the initial state of the system), this intruder essentially acts as an external observer that tries to infer whether the system has (likely or certainly, depending on the notion of opacity that is under consideration) executed activity within the secret behavior. Many related privacy/security notions, such as transitive noninterference and anonymity, can be formulated as instances of opacity.

One of the reasons that opacity has emerged at the forefront of DES research is the ever-increasing dependence of many technological systems and applications on shared cyberinfrastructure (ranging from defense and banking to health care and power distribution systems), which has prompted increased attention on various aspects of privacy and security. As an example, consider a shopping district where a security guard patrols the area in order to monitor the status of different stores. This can be modeled as a transition system (e.g., a finite automaton) with states corresponding to the different locations (stores) and with transitions corresponding to the pathways (routes) that are available between locations. Locations and/or pathways may be equipped with sensors that indicate when a location/pathway (or a subset of locations/pathways) is used. A typical opacity problem may require that the intruder (external observer) cannot identify that the guard resides outside a subset of strategic locations (e.g., banks or stores that sell expensive goods), which would indicate a convenient time for the intruder to attempt to infiltrate a strategic location.

Not surprisingly, the range of applications has also prompted many variations of opacity in DES. *Strong* notions of opacity typically require that the intruder will *never* be able to establish with *certainty* that secret behavior has occurred in the system, i.e., regardless of the underlying activity that occurs in the system, the observations that are generated can always be explained with behavior that is non-secret (implying that the external observer cannot be absolutely certain about the presence of secret behavior). Relaxations of strong notions of opacity have been pursued in two distinct directions. *Weak* opacity only requires that, for *some* behavior in the system, the intruder will never be able to establish with *certainty* that secret behavior has occurred in the system (but for other behavior, this may not necessarily be true, indicating that perhaps one should carefully confine the system behavior to ensure opacity). In stochastic systems, where activity occurs with certain probability, researchers have also developed *probabilistic* notions of opacity.

Depending on the formulation, probabilistic notions of opacity aim at either (i) analyzing the prior probability of system behavior for which the intruder may be able to infer that secret behavior has occurred with certainty (i.e., analyzing the prior probability with which strong opacity will hold) or (ii) analyzing for each scenario the posterior probability (confidence) with which the intruder might be able to conclude that secret behavior has occurred (e.g., even though the intruder may not be certain about the occurrence of secret behavior based on the observations, it may be able to infer that secret behavior has occurred with high probability).

Unlike notions of fault diagnosis discussed in ▶ “[Diagnosis of Discrete Event Systems](#)”, opacity is concerned with the hiding of information rather than the revelation of information. In the fault diagnosis setting, we are given a system with certain special (typically unobservable) events, called faults, and the goal is to determine the occurrence of these fault events. In such setting, *diagnosability* is a property that holds for the system if, under all possible system behavior that includes at least one fault event, the external observer (called diagnoser in this case) can determine (at least after a finite number of observations, i.e., with some finite delay) that the fault has occurred. To compare opacity with diagnosability, we can take the system’s secret behavior to be behavior that includes the occurrence of one or more faults. In such case, opacity would hold in the given system if under all system behavior (and also behavior that includes the fault), the external observer can never be certain that a fault has occurred. It should be clear that there is an inverse relationship between opacity and diagnosability for a given system; however, the two notions are not exact opposites. If secret behavior is taken to be behavior that includes the occurrence of one or more faults, the system cannot be opaque if it is diagnosable; however, a system that is not opaque can be diagnosable or non-diagnosable. Some discussions on the connections between diagnosability and opacity can be found in Lafortune et al. (2018).

In our opacity discussions in this article, we focus primarily on DES that can be modeled as

finite automata and only briefly describe extensions to other types of discrete event systems (in particular, probabilistic finite automata and Petri nets). The focus on finite automata allows us to concisely describe various notions of opacity, though it might be constraining in terms of some engineering applications.

Opacity in Finite Automata

Notions of opacity have been studied quite extensively in non-deterministic finite automata (NFAs) under partial observation. In this section, we describe some of the basic opacity formulations that have been investigated in this context, as well as ways to verify that they hold. Opacity notions for NFAs can be classified in two categories, depending on how the secret behavior of the NFA is described. *State-based* notions of opacity define the secret behavior with respect to a given subset of secret states S , whereas *language-based* notions of opacity define the secret behavior as a sublanguage L_S of the given NFA (i.e., a subset of finite-length sequences of events that can be generated by the NFA).

Notation and Preliminaries

This section briefly describes some notation and preliminaries, for which one can also refer to “Modeling by Cassandra”.

Let Σ be an alphabet (set of events) and denote by Σ^* the set of all finite-length strings of elements of Σ (sequences of events), including the empty string ε . A language $L \subseteq \Sigma^*$ is a subset of finite-length strings in Σ^* , i.e., sequences of a certain number of events with the convention that the first event appears on the left. Given strings $s, t \in \Sigma^*$, the string st denotes the concatenation of s and t , i.e., the sequence of events captured by s followed by the sequence of events captured by t . For any set Q , we use $|Q|$ to denote the cardinality (number of elements) of Q .

Definition 1 (Non-deterministic Finite Automaton (NFA)) A non-deterministic finite automaton is captured by $G = (X, \Sigma, \delta, X_0)$, where $X = \{x_1, x_2, \dots, x_N\}$ is the set of states,

$\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_K\}$ is the set of events, $\delta : X \times \Sigma \rightarrow 2^X$ is the non-deterministic state transition function, and $X_0 \subseteq X$ is the set of possible initial states.

Remark 1 An NFA $G = (X, \Sigma, \delta, X_0)$ is called *deterministic* if for all $x \in X$ and all $\sigma \in \Sigma$, we have $|\delta(x, \sigma)| \leq 1$. In such case, we often write $\delta(x, \sigma) = x'$ if $\delta(x, \sigma) = \{x'\}$ or say that $\delta(x, \sigma)$ is undefined if $\delta(x, \sigma) = \emptyset$.

For a subset of states $Q \subseteq X$ and $\sigma \in \Sigma$, we define $\delta(Q, \sigma) = \cup_{q \in Q} \delta(q, \sigma)$; with this notation at hand, the function δ can be extended from the domain $X \times \Sigma$ to the domain $X \times \Sigma^*$ in the routine recursive manner $\delta(x, \sigma s) := \delta(\delta(x, \sigma), s)$ for $x \in X$, $\sigma \in \Sigma$, and $s \in \Sigma^*$ (note that we define $\delta(x, \varepsilon) := \{x\}$). The behavior of G is captured by its language $L(G) := \{s \in \Sigma^* \mid \exists x_0 \in X_0 \text{ such that } \delta(x_0, s) \neq \emptyset\}$. We use $L(G, x)$ to denote the set of all traces that originate from state x of G (so that $L(G) = \bigcup_{x_0 \in X_0} L(G, x_0)$).

A very common way of capturing the capability of the external observer (intruder) is to assume that it can only observe a subset Σ_{obs} ($\Sigma_{obs} \subseteq \Sigma$) of the events. This means that Σ is partitioned into the set of observable events Σ_{obs} and the set of unobservable events $\Sigma_{uo} := \Sigma \setminus \Sigma_{obs}$. The natural projection $P_{\Sigma_{obs}} : \Sigma^* \rightarrow \Sigma_{obs}^*$ can be used to map any trace executed in the system to the sequence of observations associated with it. This projection is defined recursively as $P_{\Sigma_{obs}}(s\sigma) = P_{\Sigma_{obs}}(s)P_{\Sigma_{obs}}(\sigma)$, $s \in \Sigma^*$, $\sigma \in \Sigma$, with

$$P_{\Sigma_{obs}}(\sigma) = \begin{cases} \sigma, & \text{if } \sigma \in \Sigma_{obs}, \\ \varepsilon, & \text{if } \sigma \in \Sigma_{uo} \cup \{\varepsilon\}, \end{cases}$$

where ε represents the empty (observation) trace. In the sequel, the subscript Σ_{obs} in $P_{\Sigma_{obs}}$ will be dropped when it is clear from context.

Current-State Opacity

Perhaps the most common way of defining opacity is in terms of the perceived current state of the system. In particular, given an NFA $G = (X, \Sigma, \delta, X_0)$, a natural projection mapping P with respect to a set of observable events Σ_{obs} , and a set of secret states S ($S \subseteq X$), the system G

is said to be *current-state opaque* if the entrance of the system state to the secret set S remains *opaque* (hidden) to an external observer – at least until the system leaves the set of secret states. Current-state opacity in NFA settings has been shown to be useful in characterizing security requirements in many applications (including encryption using pseudorandom generators and coverage properties in sensor networks Saboori and Hadjicostis 2007, 2011 and Dubreil et al. 2010). Formally, current-state opacity can be defined as follows (Saboori and Hadjicostis 2007).

Definition 2 (Current-State Opacity) Consider an NFA $G = (X, \Sigma, \delta, X_0)$, along with a natural projection map P with respect to the set of observable events Σ_{obs} . We say that G is *current-state opaque* with respect to a set of secret states $S \subseteq X$, if for all $t \in L(G)$ we can find an $s \in L(G)$ such that

$$\delta(X_0, s) \cap S \neq \delta(X_0, t) \text{ and } P(s) = P(t).$$

Remark 2 Note that string s in the above definition could also be t in case $\delta(X_0, t) \cap S \neq \delta(X_0, t)$. Also note that in the above definition we have $\delta(X_0, t) \neq \emptyset$ and $\delta(X_0, s) \neq \emptyset$ (since $t, s \in L(G)$).

The above definition of current-state opacity can be rephrased in terms of the current-state estimates that the external observer reaches following the execution of a sequence of events $t \in L(G)$. More specifically, t will lead to the sequence of observations $P(t)$, and the requirement is that, for all $t \in L(G)$, the set of possible current-state estimates defined as

$$\hat{x}(P(t)) = \{x \in X \mid \exists x_0 \in X_0, \exists s \in L(G, x_0) \text{ s.t. } x \in \delta(x_0, s) \text{ and } P(s) = P(t)\} \quad (1)$$

should not be a subset of the set of secret states S , i.e., $\hat{x}(P(t)) \cap S \neq \hat{x}(P(t))$.

Once the above connection is established, it is not surprising that current-state opacity for a given NFA G can be verified using its corresponding observer automaton. The observer automaton $G_{obs} = (X_{obs}, \Sigma_{obs}, \delta_{obs}, X_{0,obs})$

(see, e.g., the discussion in Cassandras and Lafortune 2009) of the given NFA G is a deterministic finite automaton, where $X_{obs} \subseteq 2^X$ and Σ_{obs} is the set of observable events, that satisfies the following:

1. The initial state satisfies $X_{0,obs} = \{x \in X \mid \exists x_0 \in X_0, \exists t \in \Sigma_{uo}^* \text{ s.t. } x \in \delta(x_0, t)\}$. In other words, $X_{0,obs}$ corresponds to the set of states of G that can be reached from at least one of the initial states in X_0 via zero, one, or more unobservable events; this is also known as the unobservable reach from the set of states X_0 .
2. For $x_{obs} \in X_{obs}$ and $\sigma_o \in \Sigma_{obs}$, the next state transition function δ_{obs} is defined as $x'_{obs} = \delta_{obs}(x_{obs}, \sigma_o)$ where

$$x'_{obs} = \{x' \in X \mid \exists x \in x_{obs}, s \in \Sigma^* \text{ such that } x' \in \delta(x, s) \text{ and } P(s) = \sigma_o\}.$$

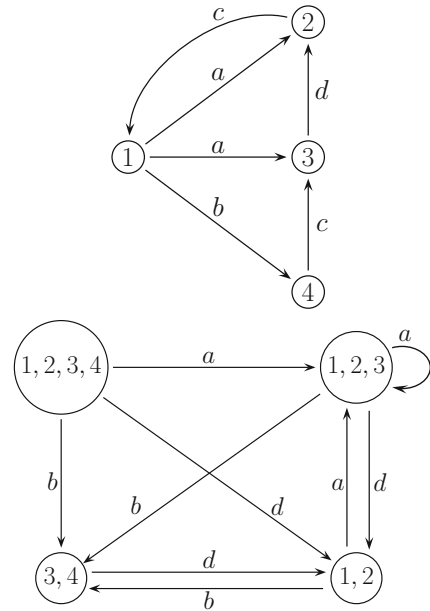
[If $x'_{obs} = \delta_{obs}(x_{obs}, \sigma_o)$ is the empty set, we often take $\delta_{obs}(x_{obs}, \sigma_o)$ to be undefined and not include $x'_{obs} = \emptyset$ in X_{obs} .]

The basic property of the observer automaton is that it can be used to obtain an estimate of the set of possible states of the system following any sequence of observations. More specifically, a sequence of events $t \in L(G)$ that occurs in the given system G generates the sequence of observations $P(t) \in \Sigma_{obs}^*$, and the corresponding set of state estimates, defined earlier as $\hat{x}(P(t))$, can be shown to satisfy

$$\hat{x}(P(t)) = \delta_{obs}(X_{0,obs}, P(t)),$$

where the definition of δ_{obs} is extended for strings of observable events (in Σ_{obs}^*) in the routine recursive manner described earlier.

Given that we have constructed the observer G_{obs} , we can verify current-state opacity for NFA G in a straightforward manner (Saboori and Hadjicostis 2007, 2011): if there exists a state x_{obs} of the observer that is nonempty and satisfies $x_{obs} \subseteq S$ (and is accessible from the initial state $X_{0,obs}$), then we know that the system is not



Opacity of Discrete Event Systems, Fig. 1 NFA G (top) and its observer G_{obs} (bottom) considered in Example 1

current-state opaque (and vice versa). Since the construction of the observer is worst-case exponential in the size of the given NFA, this approach for verifying current-state opacity has exponential complexity. It turns out that the verification of current-state opacity is a PSPACE-complete problem (see, e.g., Saboori and Hadjicostis 2011 and Cassez et al. 2012); therefore, it is unlikely that a polynomial algorithm for verifying current-state opacity exists.

Example 1 Consider NFA $G = (X, \Sigma, \delta, X_0)$ at the top of Fig. 1 with states $X = \{1, 2, 3, 4\}$, events $\Sigma = \{a, b, c, d\}$, transition function δ as shown in the figure, and initial states $X_0 = \{1, 2, 3, 4\}$. Assuming that the set of observable events $\Sigma_{obs} = \{a, b, d\}$, the observer of the system is shown at the bottom of Fig. 1. Each of the states of the observer is associated with a subset of X . In particular, the initial state of the observer is $X_{0,obs} = \{1, 2, 3, 4\}$ since initially NFA G could be in any state. From this initial observer state, there are three possible observations: (i) if a is observed the only possible states are states

in the set $\{1, 2, 3\}$ (state 1 can be reached from state 2 via the unobservable event c); (ii) if b is observed, the only possible states are states in the set $\{3, 4\}$ (state 3 can be reached from state 4 via the unobservable event c); and (iii) if d is observed, the only possible states are states in the set $\{1, 2\}$ (again, state 1 can be reached from state 2 via the unobservable event c). The observer construction can be completed by considering all possible observations (a and/or b and/or d) from each observer state.

Having constructed the observer, it is easy to verify current-state opacity with respect to a set of secret states S . For illustration purposes, we consider two different sets of secret states.

1. If $S = \{1, 2\}$, then NFA G is not current-state opaque. This is easy to see from the observer construction: the observer state that is associated with $\{1, 2\}$ is a subset of S and indicates a violation of current-state opacity. For instance, if we observe d (or bd or $bdbd$ or ad , etc.), the external observer will be certain that the system is in one of the secret states in the set $\{1, 2\}$. [Note that a sequence of observations gets generated in response to a sequence of events: for instance, the sequence of events bcd will generate the sequence of observations bd , which leads to a violation of current-state opacity.]
2. If $S = \{1, 3\}$ then NFA G is current-state opaque. This is easy to see from the observer construction as no (non-empty) observer state is associated with a subset of S (thus, a violation of current-state opacity is not possible).

Other State-Based Notions of Opacity

Apart from current-state opacity, many other state-based notions of opacity have been studied. These include current-state anonymity, initial-state opacity, K -step opacity, initial- and final-state opacity, infinite-step opacity, and many others. In all cases, the underlying system is captured by an NFA $G = (X, \Sigma, \delta, X_0)$ under a natural projection map P with respect to the set of observable events Σ_{obs} . The differences pertain to the way opacity violations are captured and the point(s) in time we are interested in. The work

in Wu and Lafortune (2013) has established that many of these notions (as well as language-based opacity which is discussed in the next section) are polynomially reducible from one to another.

In the remainder of this section, we briefly describe current-state anonymity and initial-state opacity.

Current-state anonymity does not involve a specific set of secret states S , but it is concerned with whether the external observer can determine the current state of the system exactly (the motivation in Wu et al. (2014) comes from the desire to maintain privacy when an agent attempts to use location-based services, with the state of the system being associated to the location of the agent). Formally, we have the following definition.

Definition 3 (Current-State Anonymity) Consider an NFA $G = (X, \Sigma, \delta, X_0)$, along with a natural projection map P with respect to the set of observable events Σ_{obs} . We say that current-state anonymity holds for G if for all $x_0 \in X_0$ and all $t \in L(G, x_0)$, the following is true: if $|\delta(x_0, t)| = 1$, then there exist $x'_0 \in X_0$ and $s \in L(G, x'_0)$, such that (i) $\delta(x'_0, s) \cap \delta(x_0, t) \neq \delta(x'_0, s)$ and (ii) $P(s) = P(t)$.

Another way of stating the current-state anonymity requirement is to say that

$$|\hat{x}(P(t))| > 1, \forall t \in L(G),$$

where $\hat{x}(P(t))$ was defined in (1) as the set of current-state estimates following the observation of $P(t)$. Recall that $\hat{x}(P(t))$ can be obtained as the set of states associated with the state of the observer G_{obs} reached via $\delta_{obs}(X_{0,obs}, P(t))$. Thus, one can verify current-state anonymity for a given system G by constructing its observer G_{obs} and checking that each observer state (that is reachable from the initial state) is associated with a set of state estimates of cardinality greater than unity (or zero, if we allow the observer to have a state associated with the empty set). Again, the complexity of performing this verification would be exponential in the size of the given NFA G .

Initial-state opacity is concerned with whether the external observer might be able to infer with

absolute certainty that the *initial* state of a given system definitely belonged in a set of secret *initial* states S_0 (which we can take without loss of generality to be a subset of the known possible initial states X_0). The intruder is assumed to have full knowledge of the system model and to be able to track the sequence of observations that are generated by underlying activity in the system. Initial-state opacity holds if, regardless of the underlying activity in the system, the external observer is never able to determine that the system definitely started from an initial state in the set S_0 . The following is the formal definition of initial-state opacity (Saboori and Hadjicostis 2008).

Definition 4 (Initial-State Opacity) Consider an NFA $G = (X, \Sigma, \delta, X_0)$, along with a natural projection map P with respect to the set of observable events Σ_{obs} . We say that G is initial-state opaque with respect to a set of secret initial states S_0 , $S_0 \subseteq X_0$, if for each $t \in L(G, S_0)$, we can find $s \in L(G, NS_0)$ (where $NS_0 = X_0 \setminus S_0$ is the set of non-secret initial states) such that $P(t) = P(s)$.

The verification of initial-state opacity in Saboori and Hadjicostis (2008, 2013) is done using an initial-state estimator, which acts much like an observer but for the initial-state estimates rather than the current-state estimates. Since the construction of the initial-state estimator is worst-case exponential with respect to the square of the number of states in the system, this approach has complexity that is generally exponential in the square of the size of the given NFA G (more precisely, exponential in $|X_0| \times |X|$). Initial-state opacity has been shown to be a PSPACE complete problem (see, e.g., Saboori and Hadjicostis 2013), and it is unlikely that a polynomial algorithm for its verification exists.

Example 2 Considering NFA G from Example 1 (top of Fig. 1), we easily see from its observer G_{obs} (at the bottom of the figure) that G is current-state anonymous (no observer state corresponds to a singleton set).

If $S_0 = \{1, 2\}$, then NFA G is *not* initial-state opaque. To see this, consider the sequence of observations ad ; the only state trajectories that

match the observation ad are $1 \xrightarrow{a} 3 \xrightarrow{d} 2$, $1 \xrightarrow{a} 3 \xrightarrow{d} 2 \xrightarrow{c} 1$, $2 \xrightarrow{c} 1 \xrightarrow{a} 3 \xrightarrow{d} 2$, and $2 \xrightarrow{c} 1 \xrightarrow{a} 3 \xrightarrow{d} 2 \xrightarrow{c} 1$ and clearly indicate that the initial-state estimate is $\{1, 2\}$, which violates initial-state opacity.

If $S'_0 = \{1, 3\}$, then the system is initial-state opaque. To verify this, one can construct the initial-state estimator. [In this particular example, we observe that (i) if the first observation is a or b , then the initial-state estimate will be $\{1, 2\}$ and will remain $\{1, 2\}$ regardless of subsequent observations or (ii) if the first observation is d , then the initial-state estimate will be $\{3, 4\}$ and will remain $\{3, 4\}$ regardless of subsequent observations. In both cases, initial-state opacity holds.]

Remark 3 State-based notions of opacity have an inverse relationship to notions of observability/detectability. For example, detectability in a DES requires that *eventually* (i.e., after a finite number of observations) the state of a given NFA becomes known exactly to the external observer (Shu et al. 2007). Clearly, detectability is inversely related to current-state anonymity; however, the two notions are not exact opposites: a system that is current-state anonymous cannot be detectable, but not necessarily the other way around.

Language-Based Opacity

Given an NFA G , language-based or trace-based opacity is defined with respect to secret behavior that is captured by a sublanguage of G .

Definition 5 (Language-Based Opacity) Consider an NFA $G = (X, \Sigma, \delta, X_0)$, along with a natural projection map P with respect to the set of observable events Σ_{obs} . Given a secret language L_S , $L_S \subseteq L(G)$, and a non-secret language L_{NS} , $L_{NS} \subseteq L(G)$ (typically, $L_{NS} = L(G) \setminus L_S$), we say that language-based opacity holds for G if for all $t \in L_S$, there exists another string $s \in L_{NS}$ such that $P(t) = P(s)$.

In words, any secret sequence of events $t \in L_S$ appears to the external observer in a way that is identical to the appearance of at least one other non-secret sequence of events $s \in L_{NS}$. For regular languages, one can verify language-

based opacity by reducing it (with polynomial complexity) to an initial-state opacity problem (Wu and Lafortune 2013).

Weak Notions of Opacity

Weak notions of opacity significantly relax the requirements for the (strong) opacity notions that were described earlier. Note that if a system is strongly opaque in some sense, then it is also weakly opaque in the same sense (but not necessarily the other way around). Unlike strong notions of opacity, most weak notions of opacity can be verified with polynomial complexity. Below, we provide the definition for weak language-based opacity; other weak notions of opacity can be defined in a similar manner. For technical reasons, we require that the underlying NFA is deadlock-free so that any sequence of events in its language can be extended arbitrarily.

Definition 6 (Weak Language-Based Opacity) Consider a deadlock-free NFA $G = (X, \Sigma, \delta, X_0)$, along with a natural projection map P with respect to the set of observable events Σ_{obs} . Given a secret language L_S , $L_S \subseteq L(G)$, and a non-secret language L_{NS} , $L_{NS} \subseteq L(G)$ (typically, $NS = L(G) \setminus S$), we say that weak language-based opacity holds for G if there exists an arbitrary long $t \in L_S$, for which we can find another string $s \in L_{NS}$ such that $P(t) = P(s)$.

In words, weak language-based opacity only requires the presence of two (arbitrarily long) strings with the same projection, one belonging to the secret language and one belonging to the non-secret language.

Opacity Enforcement and Label Insertion

Researchers have considered two distinct ways for dealing with systems in which an opacity notion of interest is violated. The first methodology relies on (minimally) restricting the behavior of the system via supervisory control techniques (refer to ► “Supervisory Control by Wonham”) in a way that ensures that opacity is not violated.

This is achieved by disabling the occurrence of certain events that may lead to opacity violations. The second methodology is to use obfuscation techniques to modify (as necessary) the sequence of outputs that is generated by the system, so that the external observer cannot deduce a violation of opacity; in other words, there is no restriction on the behavior of the system, but one has to be careful on how to modify the output sequence so as to consistently confuse the external observer regarding the occurrence of secret behavior.

Enforcement of opacity via supervisory control strategies was pursued for many notions of opacity in finite automata. For example, the work in Badouel et al. (2007) considers how opacity can be enforced via supervisory control in the presence of multiple intruders, whereas the work in Dubreil et al. (2010) considers a single intruder that might observe different events than the ones observed/controlled by the supervisor. The work in Saboori and Hadjicostis (2012) considers how to enforce current-state opacity in partially observed NFAs, whereas Tong et al. (2018) consider settings where the relationship between three important sets of events (events observable by the intruder, events observable by the controller, events controllable by the controller) can be more general than in earlier works.

While supervisory control methods can be successful in enforcing opacity, they restrict the possible behavior of the system. An alternative approach is to *alter* the observations that are emitted by the system, in an effort to confuse the intruder. For example, the works in Wu and Lafortune (2014, 2015) and Ji et al. (2018) devise insertion functions to insert one or more labels in between observable events. Naturally, what is challenging in this case is to ensure that the insertion of labels appears to the intruder to be consistent with (non-secret) system behavior (and that this consistency can be maintained regardless of subsequent system activity). A related approach that enforces opacity at runtime by exploiting delay (i.e., by presenting delayed information to the intruder) was considered in Falcone and Marchand (2015).

Summary and Future Directions

One limitation of the notions of opacity that have been described so far is that they fail to provide a *quantifiable* measure of opacity for a given system; instead they simply provide a binary characterization of the system (being opaque or not opaque). In stochastic settings, however, one may have additional information about the probability with which different behavior occurs in the system. This can help us refine notions of opacity in two distinct ways.

Prior Probability for Opacity Violation: One can still maintain the criteria used for the (strong) notions of opacity described earlier but ask about the (prior) probability with which these criteria will be violated. For example, if we are interested in violations of current-state opacity, we can identify all executions that lead to a violation of current-state opacity and calculate the total (prior) probability with which such problematic behavior will occur. This prior probability will be zero if the given system is strongly current-state opaque; however, even if the system is not current-state opaque, it could be that the probability of behavior that leads to current-state opacity violations is acceptably small. The notions of *step-based almost current-state opacity* and *almost current-state opacity* that appeared in Saboori and Hadjicostis (2014) for probabilistic finite automata fall in this category; both of these notions can be verified with complexity that is exponential in the size of the given PFA (Saboori and Hadjicostis 2014).

Posterior Probability of Opacity Violation: Another shortcoming of the strong notions of opacity described earlier is that they do not attempt to characterize the confidence of the intruder (whether opacity is violated or not). Consider, for example, current-state opacity with respect to a set of secret states S . In a probabilistic system, it might be that the observer cannot be absolutely certain that the current state of the system belongs to the secret set S , but it is possible that the posterior probability (conditioned on a particular sequence of observations) that the state of the

system does not belong to S is very small. The notion of *probabilistic current-state opacity* that appeared in Saboori and Hadjicostis (2014) for probabilistic finite automata falls in this category; probabilistic current-state opacity is generally undecidable (Saboori and Hadjicostis 2014).

Being an important and generally applicable security/privacy notion, opacity has naturally emerged in many other types of DES. Some of the very first opacity considerations in dynamical systems involved transition systems or Petri nets (see, e.g., Bryans et al. 2005, 2008) and led, in many cases, to undecidability results. Opacity considerations in Petri nets also resurfaced more recently: for instance, Tong et al. (2017b) exploit the notions of basis markings and basis reachability graphs to efficiently verify current-state opacity in bounded Petri nets, whereas Tong et al. (2017a) deal with decidable and undecidable problems in the verification of certain opacity properties in Petri nets.

Other DES models that have been adopted for opacity studies include timed models (see, e.g., Cassez 2009) and, more recently, timed stochastic models (Lefebvre and Hadjicostis 2019) and cyber-physical systems (Yin and Zamani 2019).

Cross-References

- [Applications of Discrete Event Systems](#)
- [Diagnosis of Discrete Event Systems](#)
- [Discrete Event Systems and Hybrid Systems, Connections Between](#)
- [Modeling, Analysis, and Control with Petri Nets](#)
- [Models for Discrete Event Systems: An Overview](#)
- [Supervisory Control of Discrete-Event Systems](#)

Recommended Reading

The interested reader is referred to Jacob et al. (2016), which provide a comprehensive review of the available literature on opacity up to the time the article was written.

Bibliography

- Badouel E, Bednarczyk M, Borzyszkowski A, Caillaud B, Darondeau P (2007) Concurrent secrets. *Discrete Event Dyn Syst* 17(4):425–446
- Bryans J, Koutny M, Ryan P (2005) Modelling opacity using Petri nets. *Electron Notes Theor Comput Sci* 121:101–115
- Bryans J, Koutny M, Mazare L, Ryan P (2008) Opacity generalised to transition systems. *Int J Inf Secur* 7(6):421–435
- Cassandras CG, Lafortune S (2009) Introduction to discrete event systems. Springer Science & Business Media, New York, USA
- Cassez F (2009) The dark side of timed opacity. In: *International Conference on Information Security and Assurance*, pp 21–30
- Cassez F, Dubreil J, Marchand H (2012) Synthesis of opaque systems with static and dynamic masks. *Form Methods Syst Des* 40(1):88–115
- Dubreil J, Darondeau P, Marchand H (2010) Supervisory control for opacity. *IEEE Trans Autom Control* 55(5):1089–1100
- Falcone Y, Marchand H (2015) Enforcement and validation (at runtime) of various notions of opacity. *Discret Event Dyn Syst* 25(4):531–570
- Jacob R, Lesage JJ, Faure JM (2016) Overview of discrete event systems opacity: models, validation, and quantification. *Ann Rev Control* 41:135–146
- Ji Y, Wu YC, Lafortune S (2018) Enforcement of opacity by public and private insertion functions. *Automatica* 93:369–378
- Lafortune S, Lin F, Hadjicostis CN (2018) On the history of diagnosability and opacity in discrete event systems. *Ann Rev Control* 45:257–266
- Lefebvre D, Hadjicostis CN (2019) Exposure time as a measure of opacity in timed discrete event systems. In: *Proceedings of European Control Conference (ECC)*
- Saboori A, Hadjicostis CN (2007) Notions of security and opacity in discrete event systems. In: *Proceedings of IEEE Conference on Decision and Control (CDC)*, pp 5056–5061
- Saboori A, Hadjicostis CN (2008) Verification of initial-state opacity in security applications of DES. In: *Proceedings of 9th International Workshop on Discrete Event Systems*, pp 328–333
- Saboori A, Hadjicostis CN (2011) Verification of K -step opacity and analysis of its complexity. *IEEE Trans Autom Sci Eng* 8(3):549–559
- Saboori A, Hadjicostis CN (2012) Opacity-enforcing supervisory strategies via state estimator constructions. *IEEE Trans Autom Control* 57(5):1155–1165
- Saboori A, Hadjicostis CN (2013) Verification of initial-state opacity in security applications of discrete event systems. *Inf Sci* 246:115–132
- Saboori A, Hadjicostis CN (2014) Current-state opacity formulations in probabilistic finite automata. *IEEE Trans Autom Control* 59(1):120–133
- Shu S, Lin F, Ying H (2007) Detectability of discrete event systems. *IEEE Trans Autom Control* 52(12):2356–2359
- Tong Y, Li Z, Seatzu C, Giua A (2017a) Decidability of opacity verification problems in labeled Petri net systems. *Automatica* 80:48–53
- Tong Y, Li Z, Seatzu C, Giua A (2017b) Verification of state-based opacity using Petri nets. *IEEE Trans Autom Control* 62(6):2823–2837
- Tong Y, Li Z, Seatzu C, Giua A (2018) Current-state opacity enforcement in discrete event systems under incomparable observations. *Discret Event Dyn Syst* 28(2):161–182
- Wu YC, Lafortune S (2013) Comparative analysis of related notions of opacity in centralized and coordinated architectures. *Discret Event Dyn Syst* 23(3):307–339
- Wu YC, Lafortune S (2014) Synthesis of insertion functions for enforcement of opacity security properties. *Automatica* 50(5):1336–1348
- Wu YC, Lafortune S (2015) Synthesis of optimal insertion functions for opacity enforcement. *IEEE Trans Autom Control* 61(3):571–584
- Wu YC, Sankararaman KA, Lafortune S (2014) Ensuring privacy in location-based services: an approach based on opacity enforcement. *IFAC Proc Vol* 47(2):33–38
- Yin X, Zamani M (2019) Towards approximate opacity of cyber-physical system. In: *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPs)*, pp 310–311

Optimal Control and Mechanics

Anthony Bloch

Department of Mathematics, The University of Michigan, Ann Arbor, MI, USA

Abstract

There are very natural close connections between mechanics and optimal control as both involve variational problems. This is a huge subject and we just touch on some interesting connections here. A survey and history may be found in Sussman and Willems (1997). Other aspects may be found in Bloch (2003).

Keywords

Nonholonomic integrator · Sub-Riemannian optimal control · Variational problems

Variational Nonholonomic Systems and Optimal Control

Variational nonholonomic problems (i.e., constrained variational problems) are equivalent to optimal control problems under certain regularity conditions. This issue was investigated in Bloch and Crouch (1994), employing the classical results of Rund (1966) and Bliss (1930), which relate classical constrained variational problems to Hamiltonian flows, although not optimal control problems. We outline the simplest relationship and refer to Bloch (2003) for more details.

Let Q be a smooth manifold and TQ its tangent bundle with coordinates $(q^i, \dot{q}^i)$. Let $L : TQ \rightarrow \mathbb{R}$ be a given smooth Lagrangian and let $\Phi : TQ \rightarrow \mathbb{R}^{n-m}$ be a given smooth function. We consider the classical Lagrange problem:

$$\min_{q(\cdot)} \int_0^T L(q, \dot{q}) dt \quad (1)$$

subject to the fixed endpoint conditions $q(0) = 0$, $q(T) = q_T$ and subject to the constraints

$$\Phi(q, \dot{q}) = 0.$$

Consider a modified Lagrangian $\Lambda(q, \dot{q}, \lambda) = L(q, \dot{q}) + \lambda \cdot \Phi(q, \dot{q})$ with Euler–Lagrange equations

$$\frac{d}{dt} \frac{\partial \Lambda}{\partial \dot{q}}(q, \dot{q}, \lambda) - \frac{\partial \Lambda}{\partial q}(q, \dot{q}, \lambda) = 0, \quad \Phi(q, \dot{q}) = 0. \quad (2)$$

We can rewrite this equation in Hamiltonian form and show that the resulting equations are equivalent to the equations of motion given by the maximum principle for a suitable optimal control problem. Set $p = \frac{\partial \Lambda}{\partial \dot{q}}(q, \dot{q}, \lambda)$ and consider this equation together with the constraints $\Phi(q, \dot{q}) = 0$. We can solve these two equations for $(\dot{q}, \lambda)$ under suitable conditions as discussed in Bloch (2003). We obtain the standard Hamiltonian equations with $H(q, p) = p \cdot \phi(q, p) - L(q, \phi(q, p))$.

We now compare this to the optimal control problem

$$\min_{u(\cdot)} \int_0^T g(q, u) dt \quad (3)$$

subject to $q(0) = 0$, $q(T) = q_T$, $\dot{q} = f(q, u)$, where $u \in \mathbb{R}^m$ and f, g are smooth functions.

Then we have the following:

Theorem 1 *The Lagrange problem and optimal control problem generate the same (regular) extremal trajectories, provided that:*

- (i) $\Phi(q, \dot{q}) = 0$ if and only if there exists a u such that $\dot{q} = f(q, u)$.
- (ii) $L(q, f(q, u)) = g(q, u)$.

For the proof and more details, see Bloch (2003).

The n -Dimensional Rigid Body

An interesting mechanical example is the n -dimensional rigid body. See Manakov (1976) and Ratiu (1980).

One can introduce a related system which we will call *the symmetric representation of the rigid body*; see Bloch et al. (2002).

By definition, *the left invariant representation of the symmetric rigid body system* is given by the first-order equations

$$\dot{Q} = Q\Omega; \quad \dot{P} = P\Omega \quad (4)$$

where $Q, P \in SO(n)$ and where Ω is regarded as a function of Q and P via the equations

$$\Omega := J^{-1}(M) \in \mathfrak{so}(n) \quad \text{and} \quad M := Q^T P - P^T Q.$$

One can check that differentiating M yields the classical form of the n -dimensional rigid body equations. For more on the precise relationship, see Bloch et al. (2002).

Now we can link the symmetric representation of the rigid body equations with the theory of optimal control. This work, developed in Bloch and Crouch (1996) and more generally in Bloch et al. (2002), has been further extended to optimal control problems for the infinitesimal generators

of group actions (so-called Clebsch optimal control problems) in Gay-Balmaz and Ratiu (2011) and Bloch et al. (2011, 2013) and even further to a class of embedded control problems in Bloch et al. (2011, 2013).

Let $T > 0$, $Q_0, Q_T \in \text{SO}(n)$ be given and fixed. Let the rigid body optimal control problem be given by

$$\min_{U \in \mathfrak{so}(n)} \frac{1}{4} \int_0^T \langle U, J(U) \rangle dt \quad (5)$$

subject to the constraint on U that there be a curve $Q(t) \in \text{SO}(n)$ such that

$$\dot{Q} = QU \quad Q(0) = Q_0, \quad Q(T) = Q_T. \quad (6)$$

Proposition 1 *The rigid body optimal control problem has optimal evolution equations (4) where P is the costate vector given by the maximum principle.*

The optimal controls in this case are given by

$$U = J^{-1}(Q^T P - P^T Q). \quad (7)$$

Kinematic Sub-Riemannian Optimal Control Problems

Optimal control of underactuated kinematic systems give rise to very interesting mechanical systems.

The problem is referred to as sub-Riemannian in that it gives rise to a geodesic flow with respect to a singular metric (see the work of Strichartz (1983, 1987) and Montgomery (2002) and references therein). This problem has an interesting history in control theory (see Brockett 1973, 1981; Baillieul 1975). See also Bloch et al. (1994) and Sussmann (1996) and further references below.

We consider control systems of the form

$$\dot{x} = \sum_{i=1}^m X_i u_i, \quad x \in M, \quad u \in \Omega \subset \mathbb{R}^m, \quad (8)$$

where Ω contains an open subset that contains the origin, M is a smooth manifold of dimension

n , and each of the vector fields in the collection $F := \{X_1, \dots, X_k\}$ is complete.

We assume that the system satisfies the accessibility rank condition and is thus controllable, since there is no drift term. Then we can pose the optimal control problem

$$\min_{u(\cdot)} \int_0^T \frac{1}{2} \sum_{i=1}^m u_i^2(t) dt \quad (9)$$

subject to the dynamics (8) and the endpoint conditions $x(0) = x_0$ and $x(T) = x_T$. These problems were studied by Griffiths (1983) from the constrained variational viewpoint and from the optimal control viewpoint by Brockett (1981, 1983). In the sub-Riemannian geodesic problem, abnormal extremals play an important role. See work by Strichartz (1983), Montgomery (1994, 1995), Sussmann (1996), and Agrachev and Sarychev (1996).

Example: Optimal Control and a Particle in a Magnetic Field The control analysis of the Heisenberg model or nonholonomic integrator goes back to Brockett (1981) and Baillieul (1975), while a modern treatment of the relationship with a particle in a magnetic field may be found in Montgomery (1993), for example. A nice treatment of the pure mechanical aspects of a particle in a magnetic field may be found in Marsden and Ratiu (1999).

The Heisenberg optimal control equations are a particular case of planar charged particle motion in a magnetic field. This may be seen by considering the slightly more general problem below.

We now consider the optimal control problem

$$\min \int (u^2 + v^2) dt \quad (10)$$

subject to the equations

$$\begin{aligned} \dot{x} &= u, \\ \dot{y} &= v, \\ \dot{z} &= A_1 u + A_2 v, \end{aligned} \quad (11)$$

where $A_1(x, y)$ and $A_2(x, y)$ are smooth functions of x and y . $A_1 = y$ and $A_2 = -x$ recover the Heisenberg/nonholonomic integrator equations. More generally we get the flow of a particle in a magnetic field – it is not hard to carry out the optimal control analysis to see this. Details are in Bloch (2003).

Cross-References

- [Discrete Optimal Control](#)
- [Optimal Control and Pontryagin's Maximum Principle](#)
- [Optimal Control with State Space Constraints](#)
- [Singular Trajectories in Optimal Control](#)

Bibliography

- Agrachev AA, Sarychev AV (1996) Abnormal sub-Riemannian geodesics: Morse index and rigidity. *Ann Inst H Poincaré Anal Non Linéaire* 13:635–690
- Baillieul J (1975) Some optimization problems in geometric control theory. Ph.D. thesis, Harvard University
- Baillieul J (1978) Geometric methods for nonlinear optimal control problems. *J Optim Theory Appl* 25:519–548
- Bliss G (1930) The problem of lagrange in the calculus of variations. *Am J Math* 52:673–744
- Bloch AM (2003) (with Baillieul J, Crouch PE, Marsden JE), *Nonholonomic mechanics and control*. Interdisciplinary applied mathematics. Springer, New York
- Bloch AM, Crouch PE (1994) Reduction of Euler–Lagrange problems for constrained variational problems and relation with optimal control problems. In: *Proceedings of the 33rd IEEE conference on decision and control*, Lake Buena Vista. IEEE, pp 2584–2590
- Bloch AM, Crouch PE (1996) Optimal control and geodesic flows. *Syst Control Lett* 28(2):65–72
- Bloch AM, Crouch PE, Ratiu TS (1994) Sub-Riemannian optimal control problems. *Fields Inst Commun AMS* 3:35–48
- Bloch AM, Crouch P, Marsden JE, Ratiu TS (2002) The symmetric representation of the rigid body equations and their discretization. *Nonlinearity* 15:1309–1341
- Bloch AM, Crouch PE, Nordkvist N, Sanyal AK (2011) Embedded geodesic problems and optimal control for matrix Lie groups. *J Geom Mech* 3:197–223
- Bloch AM, Crouch PE, Nordkvist N (2013) Continuous and discrete embedded optimal control problems and their application to the analysis of Clebsch optimal control problems and mechanical systems. *J Geom Mech* 5:1–38
- Brockett RW (1973) Lie theory and control systems defined on spheres. *SIAM J Appl Math* 25(2):213–225
- Brockett RW (1981) Control theory and singular Riemannian geometry. In: Hilton PJ, Young GS (eds) *New directions in applied mathematics*. Springer, New York, pp 11–27
- Brockett RW (1983) Nonlinear control theory and differential geometry. In: *Proceedings of the international congress of mathematicians*, Warsaw, pp 1357–1368
- Gay-Balmaz F, Ratiu TS (2011) Clebsch optimal control formulation in mechanics. *J Geom Mech* 3:41–79
- Griffiths PA (1983) *Exterior differential systems*. Birkhäuser, Boston
- Manakov SV (1976) Note on the integration of Euler's equations of the dynamics of an n -dimensional rigid body. *Funct Anal Appl* 10:328–329
- Marsden JE, Ratiu TS (1999) *Introduction to mechanics and symmetry*. Texts in applied mathematics, vol 17. Springer, New York. (1st edn. 1994; 2nd edn. 1999)
- Montgomery R (1993) Gauge theory of the falling cat. *Fields Inst Commun* 1:193–218
- Montgomery R (1994) Abnormal minimizers. *SIAM J Control Optim* 32:1605–1620
- Montgomery R (1995) A survey of singular curves in sub-Riemannian geometry. *J Dyn Control Syst* 1:49–90
- Montgomery R (2002) *A tour of sub-Riemannian geometries, their geodesics and applications*. Mathematical surveys and monographs, vol 91. American Mathematical Society, Providence
- Ratiu T (1980) The motion of the free n -dimensional rigid body. *Indiana U Math J* 29:609–627
- Rund H (1966) *The Hamiltonian–Jacobi theory in the calculus of variations*. Krieger, New York
- Strichartz R (1983) Sub-Riemannian geometry. *J Diff Geom* 24:221–263; see also *J Diff Geom* 30:595–596 (1989)
- Strichartz RS (1987) The Campbell–Baker–Hausdorff–Dynkin formula and solutions of differential equations. *J Funct Anal* 72:320–345
- Sussmann HJ (1996) A cornucopia of four-dimensional abnormal sub-Riemannian minimizers. In: Bellaïche A, Risler J-J (eds) *Sub-Riemannian geometry*. Progress in mathematics, vol 144. Birkhäuser, Basel, pp 341–364
- Sussmann HJ, Willems JC (1997) 300 years of optimal control: from the Brachystochrone to the maximum principle. *IEEE Control Syst Mag* 17:32–44

Optimal Control and Pontryagin's Maximum Principle

Richard Vinter
Imperial College, London, UK

Abstract

Pontryagin's Maximum Principle is a set of conditions providing information about solutions to optimal control problems; that is, optimization problems with differential equation constraints. It unifies and extends many classical necessary conditions from the calculus of variations. This article provides an overview of the Maximum Principle, including free end-time and nonsmooth versions. A time-optimal control problem is solved, to illustrate its application.

Keywords

Dynamic constraints · Hamiltonian system · Maximum principle · Nonlinear systems · Optimization

Optimal Control

A widely used framework for studying minimization problems, encountered in the optimal selection of flight trajectories and other areas of advanced engineering design and operation involving dynamic constraints, is to view them as special cases of the problem:

$$(P) \left\{ \begin{array}{l} \text{Minimize } J(x(\cdot), u(\cdot)) \\ \quad := \int_0^T L(t, x(t), u(t)) dt + g(x(0), x(T)) \\ \text{over measurable functions} \\ \quad u(\cdot) : [0, T] \rightarrow R^m \text{ and} \\ \quad \text{absolutely continuous functions} \\ \quad x(\cdot) : [0, T] \rightarrow R^n \text{ satisfying} \\ \quad \dot{x}(t) = f(t, x(t), u(t)) \text{ a.e.,} \\ \quad u(t) \in \Omega \text{ a.e.,} \\ \quad (x(0), x(T)) \in C, \end{array} \right.$$

the data for which comprise a number $T > 0$, functions $f : [0, T] \times R^n \times R^m \rightarrow R^n$, $L : [0, T] \times R^n \times R^m \rightarrow R$, and $g : R^n \times R^n \rightarrow R$ and sets $C \subset R^n \times R^n$ and $\Omega \subset R^m$.

It is assumed that set C has the functional inequality and equality constraint set representation

$$C = \{(x_0, x_1) \in R^n \times R^n : \\ \times \left\{ \begin{array}{l} \phi^i(x_0, x_1) \leq 0 \text{ for } i = 1, 2, \dots, k_1 \\ \psi^i(x_0, x_1) = 0 \text{ for } i = 1, 2, \dots, k_2 \end{array} \right\} \} \quad (1)$$

in which $\phi^i : R^n \times R^n \rightarrow R$, $i = 1, \dots, k_1$ and $\psi^i : R^n \times R^n \rightarrow R$, $i = 1, \dots, k_2$ are given functions.

A control function is a measurable function $u(\cdot) : [0, T] \rightarrow R^m$ satisfying $u(t) \in \Omega$ a.e. $t \in [0, T]$. A state trajectory $x(\cdot) : [0, T] \rightarrow R^n$ associated with a control function $u(\cdot)$ is a solution to the differential equation $\dot{x}(t) = f(t, x(t), u(t))$. A pair of functions $(x(\cdot), u(\cdot))$ comprising a control function $u(\cdot)$ and an associated state trajectory $x(\cdot)$ satisfying the condition $(x(0), x(T)) \in C$ is a feasible process. A feasible process $(\bar{x}(\cdot), \bar{u}(\cdot))$ which achieves the minimum of $J(x(\cdot), u(\cdot))$ over all feasible processes is called a minimizer.

Frequently, the initial state is fixed, i.e., C takes the form

$$C = \{x_0\} \times C_1 \text{ for some } x_0 \in R^n \text{ and some } C_1 \subset R^n.$$

In this case, the domain of the minimization problem (P) can be taken to be control functions since there is a unique state trajectory (with a fixed left endpoint) associated with a given control function. Allowing freedom in the choice of initial state introduces a flexibility into the formulation which is useful in some applications however.

Optimization problems involving dynamic constraints (such as, but not exclusively, those expressed as controlled differential equations) are known as optimal control problems. Various frameworks are available for studying such

problems. (P) is of special importance, since it embraces a wide range of significant dynamic optimization problems which are beyond the reach of traditional variational techniques and, at the same time, it is well suited to the derivation of general necessary conditions of optimality.

The Maximum Principle

Pontryagin's Maximum Principle, the centerpiece of optimal control theory, is a set of necessary conditions for a feasible process to be a minimizer. It came to prominence through a book by Pontryagin et al., published in 1961 and brought out in English translation, Pontryagin et al. (1962), one year later. It bears the name of Pontryagin, because of his position as leader of the research group at the Steklov Institute, Moscow, which achieved this advance. But the first proof is attributed to Boltyanskii.

For given $\lambda \geq 0$, define the Hamiltonian function $H_\lambda : [0, T] \times R^n \times R^n \times R^m \rightarrow R'$

$$H_\lambda(t, x, p, u) := p^T f(t, x, u) - \lambda L(t, x, u).$$

Theorem 1 (The Maximum Principle) *Let $(\bar{x}(\cdot), \bar{u}(\cdot))$ be a minimizer for (P). Assume that the following hypotheses are satisfied:*

- (i) g is continuously differentiable.
- (ii) $\phi^i, i = 1, \dots, k_1$ and $\psi^i, i = 1, \dots, k_2$, are continuously differentiable.
- (iii) L and f are continuous, $L(t, \cdot, u)$ and $f(t, \cdot, u)$ are continuously differentiable for each (t, u) , and there exist $\epsilon > 0$ and $k(\cdot) \in L^1$ such that

$$|L(t, x, u) - L(t, x', u)| + |f(t, x, u) - f(t, x', u)| \leq k(t)|x - x'|$$

for all $x, x' \in R^n$ such that $|x - \bar{x}(t)| \leq \epsilon$, $|x' - \bar{x}(t)| \leq \epsilon$, and $u \in \Omega$, a.e. $t \in [0, T]$

- (iv) Ω is a Borel set.

Then there exist a number λ ($\lambda = 0$ or 1), an absolutely continuous arc $p : [0, T] \rightarrow R^n$,

numbers $\alpha^i \geq 0$ for $i = 1, \dots, k_1$ and numbers β^i for $i = 1, \dots, k_2$ satisfying

$$(p(\cdot), \lambda, \{\alpha^i\}, \{\beta^i\}) \neq (0, 0, \{0, \dots, 0\}, \{0, \dots, 0\}) \quad (2)$$

and such that the following conditions are satisfied:

The Costate Equation:

$$\begin{aligned} -\dot{p}(t) &= \frac{\partial}{\partial x} f^T(t, \bar{x}(t), \bar{u}(t)) p(t) \\ &\quad - \lambda \frac{\partial}{\partial x} L^T(t, \bar{x}(t), \bar{u}(t)), \text{ a.e.,} \end{aligned}$$

The Maximization of the Hamiltonian Condition:

$$\begin{aligned} H_\lambda(t, \bar{x}(t), p(t), \bar{u}(t)) \\ = \max_{u \in \Omega} H_\lambda(t, \bar{x}(t), p(t), u) \text{ a.e.,} \end{aligned}$$

The Transversality Condition:

$$\begin{aligned} (p^T(0), -p^T(T)) &= \lambda \nabla g(\bar{x}(0), \bar{x}(T)) \\ &\quad + \sum_{i=1}^{k_1} \alpha^i \nabla \phi^i(\bar{x}(0), \bar{x}(T)) \\ &\quad + \sum_{i=1}^{k_2} \beta^i \nabla \psi^i(\bar{x}(0), \bar{x}(T)) \end{aligned}$$

and

$$\begin{aligned} \alpha^i &= 0 \text{ for all } i \in \{1, \dots, k_1\} \\ \text{s.t. } \phi^i(\bar{x}(0), \bar{x}(T)) &< 0, \end{aligned}$$

in which, for a generic function $h : R^n \times R^n \rightarrow R$,

$$\begin{aligned} \nabla h(x_0, x_1)(\bar{x}_0, \bar{x}_1) \\ := \left[\frac{\partial}{\partial x_0} h(\bar{x}_0, \bar{x}_1), \frac{\partial}{\partial x_1} h(\bar{x}_0, \bar{x}_1) \right]. \quad (3) \end{aligned}$$

If the functions $L(t, x, u)$ and $f(t, x, u)$ are independent of t , in which case H_λ no longer depends on t , the following condition is also satisfied

Constancy of the Hamiltonian for Autonomous Problems:

$H_\lambda(\bar{x}(t), p(t), \bar{u}(t)) = c \quad \text{a.e., for some constant } c.$

We allow the cases $k_1 = 0$ (no inequality constraints) and $k_2 = 0$ (no equality endpoint-constraints). In the first case, the non-degeneracy condition becomes $(p(\cdot), \lambda, \{\beta^i\}) \neq (0, 0, 0)$, and the summation involving the α^i 's is dropped from the transversality condition and non-triviality condition (2). The second case, or any combination of the two cases, is treated similarly.

Derivation of the costate equation and boundary conditions. A simple way to derive the differential equations for the $p_i(\cdot)$'s is, first, to construct the Hamiltonian $H_\lambda(t, x, p, u) = p^T f(t, x, u) - \lambda L(t, x, u)$ and, second, to use the fact that the i th component $p_i(\cdot)$ of the costate $p(t) = [p_1(t), \dots, p_n(t)]^T$ satisfies the equation:

$$-\dot{p}_i(t) = \frac{\partial}{\partial x_i} H_\lambda(t, \bar{x}(t), p(t), \bar{u}(t))$$

for $i = 1, \dots, n.$

The preceding equations are of course merely a component-wise statement of the costate equation above. In many applications the endpoint constraints take the form

$$\begin{aligned} x_i(0) &= \xi_0^i \text{ for } i \in J_0 \text{ and } x_i(0) \in R^n \text{ for } i \notin J_0 \\ x_i(T) &= \xi_1^i \text{ for } i \in J_1 \text{ and } x_i(T) \in R^n \text{ for } i \notin J_1 \end{aligned}$$

for given index sets $J_0, J_1 \subset \{0, \dots, n\}$ and n -vectors ξ_0^i for $i \in J_0$ and ξ_1^i for $i \in J_1$, i.e., the endpoints of each state trajectory component are either "fixed" or "free." In such cases the rules for setting up the boundary conditions on the $p_i(\cdot)$'s are

$$\begin{aligned} p_i(0) &\in R^n \text{ for } i \in J_0 \text{ and } p_i(0) = \lambda \frac{\partial}{\partial x_{0i}} g(\bar{x}(0), \bar{x}(T)) \text{ for } i \notin J_0 \\ p_i(T) &\in R^n \text{ for } i \in J_1 \text{ and } -p_i(T) = \lambda \frac{\partial}{\partial x_{1i}} g(\bar{x}(0), \bar{x}(T)) \text{ for } i \notin J_1, \end{aligned}$$

i.e., if $x_i(0)$ (respectively, $x_i(T)$) is fixed, then $p_i(0)$ (respectively, $p_i(T)$) is free, and if $x_i(0)$ (respectively, $x_i(T)$) is free, then $p_i(0)$ (respectively, $p_i(T)$) is fixed.

The optimal control problem (P) is a generalization of the following problem in the calculus of variations:

$$\left\{ \begin{array}{l} \text{Minimize } \int_0^T L(t, x(t), \dot{x}(t)) dt \\ \text{over absolutely continuous arcs } x(\cdot) : \\ [0, T] \rightarrow R^n \text{ satisfying} \\ (x(0), x(T)) = (a, b). \end{array} \right. \quad (4)$$

for given $L : [0, T] \times R^n \times R^n \rightarrow R$ and $(a, b) \in R^n \times R^n$. This problem is a special case of (P) in which $f(t, x, u) = u$, $\Omega = R^n$, $k_1 = 0$, $k_2 = 2n$ and

$$\begin{aligned} & \left((\psi^1(x_0, x_1), \dots, \psi^n(x_0, x_1)) \right), \\ & \left(\psi^{n+1}(x_0, x_1), \dots, \psi^{2n}(x_0, x_1) \right) \\ & = (x_0^T - a^T, x_1^T - b^T). \end{aligned}$$

It is a straightforward exercise to deduce from the Maximum Principle, in this special case, that a minimizer satisfies the classical Euler-Lagrange and Weierstrass conditions and also that the minimizer and associated costate arc satisfy Hamilton's system of equations, under an additional uniform convexity hypothesis on $L(t, x, \cdot)$. Thus, the Maximum Principle unifies many of the classical necessary conditions from the calculus of variations and, furthermore, validates them under reduced hypotheses. But it has far-reaching implications, beyond these

conditions, because it allows the presence of pathwise constraints on the velocities, expressed in terms of a controlled differential equation and a control constraint set, which are encountered in engineering design, econometrics, and other areas of application.

The Hamiltonian System

In favorable circumstances, we are justified in setting the cost multiplier $\lambda = 1$ and, furthermore, the maximization of the Hamiltonian condition permits us, for each t , to express u as a function $u^*(., .)$ of x and p :

$$u = u^*(t, x, p).$$

The Maximum Principle now asserts that a minimizing arc $\bar{x}(\cdot)$ is the first component of a pair of absolutely continuous functions $(\bar{x}(\cdot), p(\cdot))$ satisfying *Hamilton's system of equations*:

$$\begin{aligned} (-\dot{p}^T(t), \dot{\bar{x}}^T(t)) &= \nabla_{xp} H_1 \\ &\times (t, \bar{x}(t), p(t), u^*(t, \bar{x}(t), p(t))) \quad \text{a.e.,} \end{aligned} \quad (5)$$

in which $\nabla_{xp} H_1$ denotes the gradient of $H(t, x, p, u)$ w.r.t. the vector (x, p) variables for fixed (t, u) , together with the endpoint conditions

$$\begin{aligned} (\bar{x}(0), \bar{x}(T)) &\in C \text{ and} \\ (p^T(0), -p^T(T)) &= \lambda \nabla g(\bar{x}(0), \bar{x}(T)) \\ &+ \sum_{i=1}^{k_1} \alpha^i \nabla \phi^i(\bar{x}(0), \bar{x}(T)) + \sum_{i=1}^{k_2} \beta^i \nabla \psi^i(\bar{x}(0), \bar{x}(T)), \end{aligned}$$

for some nonnegative numbers $\{\alpha^i\}$ and numbers $\{\beta^i\}$ satisfying

$$\begin{aligned} \alpha^i &= 0 \text{ for all } i \in \{1, \dots, k_1\} \text{ s.t. } \phi^i(\bar{x}(0), \\ &\bar{x}(T)) < 0, \end{aligned}$$

where ∇g , $\nabla \phi$ and $\nabla \psi$ etc., are as defined in (3). The minimizing control satisfies the relation

$$\bar{u}(t) = u^*(t, \bar{x}(t), p(t)).$$

Notice that the first-order vector differential equation (5) is a system of $2n$ scalar, first-order differential equations. Let us suppose that $\bar{k}_1$ inequality endpoint constraints are active at $(\bar{x}(0), \bar{x}(T))$. Then, satisfaction of the active constraints and the transversality condition supply $2n + \bar{k}_1 + k_2$ equations for the boundary values of $(\bar{x}(\cdot), p(\cdot))$. Taking account of the fact that there are $\bar{k}_1 + k_2$ unknown endpoint multipliers, however, we see that the effective

number of endpoint constraints accompanying the differential equation (5) is

$$2n + \bar{k}_1 + k_2 - (\bar{k}_1 + k_2) = 2n.$$

Thus the set of $2n$ scalar first-order differential equations (5) giving rise to a “two-point boundary value problem” for $(\bar{x}, p)$ is fully determined in the sense that the number of individual boundary conditions matches the number first order scalar equations.

Refinements

Free-Time Problems Consider a variant on the “autonomous” case of problem (P) (L and f do not depend on t), call it (FT), in which the terminal time T is no longer fixed but is a choice variable along with the control function and the initial state, and the cost function is

$$\tilde{J}(T, x(\cdot), u(\cdot))$$

$$:= \int_0^T L(x(t), u(t))dt + \tilde{g}(T, x(0), x(T))$$

for some function $\tilde{g}(\cdot, \cdot, \cdot)$. Take a minimizer $(\bar{T}, \bar{x}(\cdot), \bar{u}(\cdot))$ for (FT) . Assume, in addition to hypotheses (i)–(iii), that $\mathcal{Q}$ is bounded and the function $k(\cdot)$ in (iii) is bounded. Then the Maximum Principle conditions (for data in which the end-time is frozen at $T = \bar{T}$) continue to be satisfied for some $p(\cdot) : [0, \bar{T}] \rightarrow R^n$ and λ , including the constancy of the Hamiltonian condition

$$H_\lambda(\bar{x}(t), p(t), \bar{u}(t)) = c \quad \text{a.e } t \in [0, \bar{T}]$$

for some constant c . But a new condition is required to reflect the extra degree of freedom in the new problem specification, namely, the free end-time. This is an additional transversality condition involving the constant value c of the Hamiltonian:

Free End-Time Transversality Condition: $c = \lambda \frac{\partial}{\partial T} g(\bar{T}, \bar{x}(0), \bar{x}(T))$.

Other Refinements Versions of the Maximum Principle are available to take account of pathwise functional inequality constraints on state variables (“pure” state constraints) and of both state and control variables (“mixed” constraints). Maximum Principle-like conditions have also been derived for optimal control problems in which the dynamic constraint takes the form of a retarded differential equation with control terms and in which the class of control functions is enlarged to include Dirac delta functions (“impulse” optimal control problems).

The Nonsmooth Maximum Principle

In early derivations of the Maximum Principle, it was assumed that the functions $f(t, x, u)$ and $L(t, x, u)$ were continuously differentiable

with respect to the x variable. A major research endeavor since the early 1970s has been to find versions of the Maximum Principle that remain valid when the functions $f(t, x, u)$ and $L(t, x, u)$ satisfy merely a “bounded slope” or, synonymously, a Lipschitz continuity condition with respect to x . Such functions are “nonsmooth” in the sense that they can fail to be differentiable, in the conventional sense, at some points in their domains. An overview of the Maximum Principle would be incomplete without reference to such advances.

The search for nonsmooth optimality conditions is motivated by a desire to solve optimal control problems where, in particular, the function $f(t, x, u)$ is a piecewise linear function of x (for fixed (t, u)). Such functions arise, for example, when the $f(t, x, u)$ is constructed empirically via a lookup table and linear interpolation. Nonsmooth cost integrands are encountered when they are constructed using “pointwise” supremum and/or “absolute value” operations. The function

$$J(x(\cdot)) = \int_0^T |x(t)|dt + \max\{x(1), 0\},$$

which penalizes the L^1 norm of the state trajectory and the terminal value of the scalar state, but only if this is nonnegative, is a case in point.

When attempting to generalize the Maximum Principle to allow for nonsmooth data, we encounter the challenge of interpreting the costate equation, which can be written as

$$-\dot{p}(t) = \frac{\partial}{\partial x} H_\lambda(t, \bar{x}(t), \bar{u}(t)p(t)),$$

in circumstances when the x -gradients of f and L are not defined, at least not in a conventional sense. One approach to dealing with this problem is via the Clarke generalized gradient ∂m of function $m: R^n \rightarrow R$ at a point $\bar{x}$:

$\partial m(\bar{x}) := \text{co} \{ \xi \mid \text{there exist sequences } x^i \rightarrow \bar{x}, \xi^i \rightarrow \xi \text{ s.t., for each } i,$

$m(\cdot)$ is Fréchet differentiable at x^i and $\xi_i = \frac{\partial}{\partial x} m(x^i) \}$.

Here, “co” means closed convex hull. In a landmark paper, *Clarke FH 76*, Clarke proved a necessary condition commonly referred to as the nonsmooth Maximum Principle, in which the costate equation is replaced by a differential inclusion involving the (partial) generalized gradient $\partial_x H(t, \bar{x}(t), p(t), \bar{u}(t))$ of $H(t, \cdot, p(t), \bar{u}(t))$ w.r.t x , evaluated at $\bar{x}(t)$, namely,

$$-\dot{p}^T(t) \in \partial_x H(t, \bar{x}(t), \bar{u}(t)) \quad \text{a.e. } t \in [0, T].$$

This formulation of the ‘costate inclusion’ for the nonsmooth Maximum Principle and the unrestrictive hypothesis under which it is derived in this paper remain state of the art. Another approach is based on the concept of ‘derivate containers’ *Warga J 83*.

Example

We illustrate the application of the Maximum Principle with a simple example. It has the fol-

lowing interpretation. A 1 kg mass is located 1 m along the line and has zero initial velocity. We seek a time $\bar{T} > 0$ s, which is the minimum over all times $T > 0$ having the property: there exists a time-varying force $u(t)$, $0 \leq t \leq T$ satisfying

$$-1 \leq u(t) \leq +1$$

such that, under the action of the force, the mass is located at the origin with zero velocity at time T . Note that, in consequence of Newton’s second law, the vector $x(t) = (x_1(t), x_2(t))$ comprising the displacement and velocity of the mass satisfies

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t). \quad (6)$$

This is a special case of the free end-time problem

$$\left\{ \begin{array}{l} \text{Minimize } T \\ \text{over times } T > 0, \text{ measurable functions } u(\cdot) : [0, T] \rightarrow R \text{ and} \\ \text{absolutely continuous functions } x(\cdot) : [0, T] \rightarrow R^2 \text{ such that} \\ \dot{x}(t) = Ax(t) + bu(t) \quad \text{a.e.} \\ u(t) \in \Omega \quad \text{a.e.} \\ (x_1(0), x_2(0)) = (1, 0) \text{ and } (x_1(T), x_2(T)) = (0, 0). \end{array} \right.$$

in which $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and $\Omega = [-1, +1]$.

The (free-time) Maximum Principle provides the following information about a minimizing end-time $\bar{T}$, control $\bar{u}(\cdot)$, and corresponding state $\bar{x}(\cdot) = (\bar{x}_1(\cdot), \bar{x}_2(\cdot))$. There exists an arc $p(\cdot) = [p_1(\cdot), p_2(\cdot)]^T$ such that

$$\dot{\bar{x}}_1(t) = \bar{x}_2(t) \quad \text{and} \quad \dot{\bar{x}}_2(t) = \bar{u}(t), \quad (7)$$

$$-\dot{p}_1(t) = 0 \quad \text{and} \quad -\dot{p}_2(t) = p_1(t), \quad (8)$$

$$\bar{u}(t) = \arg \max \{ p_2(t)u \mid u \in [-1, +1] \} \quad (9)$$

$$(\bar{x}_1, \bar{x}_2)(0) = (1, 0) \text{ and } (\bar{x}_1, \bar{x}_2)(T) = (0, 0) \quad (10)$$

$$p_1(t) \bar{x}_2(t) + |p_2(t)| = \lambda \quad \text{for all } t. \quad (11)$$

Condition (9) permits us to express $\bar{u}(\cdot)$ in terms of $p_2(\cdot)$, thus

$$\bar{u}(t) = \text{sign}\{p_2(t)\},$$

and thereby eliminate $\bar{u}(\cdot)$. It can be shown that relations (7), (8), (9), (10), and (11) have a unique solution for $\bar{T}$, $\bar{u}(t)$, $\bar{x}(t)$, $p(t)$ and $\lambda = 0$ or 1. Furthermore, these relations cannot be satisfied with $\lambda = 0$. The unique solution (with $\lambda = 1$) is

$$\bar{T} = 2$$

$$(\bar{x}_1(t), \bar{x}_2(t)) = \begin{cases} (1 - \frac{1}{2}t^2, -t) & \text{if } t \in [0, 1) \\ (\frac{1}{2} - (t-1) + \frac{1}{2}(t-1)^2 - 1 + \frac{1}{2}(t-1)), & \text{if } t \in [1, 2], \end{cases}$$

$$\bar{u}(t) = \begin{cases} -1 & \text{if } t \in [0, 1) \\ +1 & \text{if } t \in [1, 2], \end{cases}$$

$$p_1(t) = -1 \text{ and } p_2(t) = -1 + t \quad \text{for } t \in [0, 2].$$

The Maximum Principle is a necessary condition of optimality. Since a minimizer exists and since $(\bar{T}, \bar{x}(\cdot), \bar{u}(\cdot), p(\cdot))$ is a unique solution to the Maximum Principle relations, it follows that $(\bar{T}, \bar{x}(\cdot), \bar{u}(\cdot))$ is the solution to the problem.

This problem is amenable to simpler, more elementary, solution techniques. But the above solution is enlightening, because it highlights important generic features of the Maximum Principle. We see how the “maximization of the Hamiltonian condition” can be used to eliminate the control function and thereby to set up a two-point boundary problem for $\bar{x}(\cdot)$ and $p(\cdot)$ (a very nonclassical construction).

Cross-References

- [Numerical Methods for Nonlinear Optimal Control Problems](#)
- [Optimal Control and Mechanics](#)
- [Optimal Control and the Dynamic Programming Principle](#)
- [Optimal Control with State Space Constraints](#)

Recommended Reading

- Berkovitz LD (1974) Optimal control theory. Applied mathematical sciences, vol 12. Springer, New York
- Bryson AE, Ho Y-C (1975) Applied optimal control (Revised edn), Halstead Press (a division of John Wiley and Sons), New York

- Fleming WH, Rishel RW (1975) Deterministic and stochastic optimal control. Springer, New York
- Ioffe AD, Tihomirov VM (1979) Theory of extremal problems. North-Holland, Amsterdam

Reflections on the origins of the Maximum Principle appear in:

- Pesch HJ, Plail M (2009) The maximum principle of optimal control: a history of ingenious ideas and missed opportunities. Control Cybern 38:973–995

Expository texts that also cover advances in the theory of necessary conditions related to the Maximum Principle, based on techniques of Nonsmooth Analysis:

- Clarke FH (1983) Optimization and nonsmooth analysis. Wiley-Interscience, New York
- Clarke FH (2013) Functional analysis, calculus of variations and optimal control. Graduate texts in mathematics. Springer, London
- Vinter RB (2000) Optimal control. Birkhäuser, Boston

Engineering text illustrating the application of the Maximum Principle to solve problems of optimal control and design, in flight mechanics and other areas

- Bryson AE (1999) Dynamic optimization. Addison Wesley Longman, Menlo Park

Bibliography

- Clarke FH (1976) The maximum principle under minimal hypotheses. SIAM J Control Optim 14:1078–1091
- Pontryagin LS, Boltyanskii VG, Gamkrelidze RV, Mishchenko EF (1962) The mathematical theory of optimal processes. Tririgoff KN Transl., Neustadt LW Ed. Wiley, New York
- Warga J (1983) Optimization and controllability without differentiability assumptions. SIAM J Control Optim 21:239–260

Optimal Control and the Dynamic Programming Principle

Maurizio Falcone

Dipartimento di Matematica, SAPIENZA –
Università di Roma, Rome, Italy

Abstract

This entry illustrates the application of Bellman's dynamic programming principle within the context of optimal control problems for continuous-time dynamical systems. The approach leads to a characterization of the optimal value of the cost functional, over all possible trajectories given the initial conditions, in terms of a partial differential equation called the Hamilton–Jacobi–Bellman equation. Importantly, this can be used to synthesize the corresponding optimal control input as a state-feedback law.

Keywords

Continuous-time dynamics ·
Hamilton–Jacobi–Bellman equation ·
Optimization · Nonlinear systems · State
feedback

Introduction

The dynamic programming principle (DPP) is a fundamental tool in optimal control theory. It was largely developed by Richard Bellman in the 1950s (Bellman 1957) and has since been applied to various problems in deterministic and stochastic optimal control. The goal of optimal control is to determine the control function and the corresponding trajectory of a dynamical system which together optimize a given criterion usually expressed in terms of an integral along the trajectory (the cost functional) (Fleming and Rishel 1975; Macki and Strauss 1982). The function which associates with the initial condition

of the dynamical system the optimal value of the cost functional among all the possible trajectories is called the *value function*. The most interesting point is that via the dynamic programming principle, one can derive a characterization of the value function in terms of a nonlinear partial differential equation (the Hamilton–Jacobi–Bellman equation) and then use it to synthesize a feedback control law. This is the major advantage over the approach based on the Pontryagin Maximum Principle (PMP) (Boltyanskii et al. 1956; Pontryagin et al. 1962). In fact, the PMP merely gives necessary conditions for the characterization of the open-loop optimal control and of the corresponding optimal trajectory. The DPP has also been applied to construct approximation schemes for the value function although this approach suffers from the “curse of dimensionality” since one has to solve a nonlinear partial differential equation in a high dimension. Despite the elegance of the DPP approach, its practical application is limited by this bottleneck, and the solution of many optimal control problems has been accomplished instead via the two-point boundary value problem associated with the PMP.

The Infinite Horizon Problem

Let us present the main ideas for the classical *infinite horizon problem*. Let a controlled dynamical system be given by

$$\begin{cases} \dot{y}(s) = f(y(s), \alpha(s)) \\ y(t_0) = x_0. \end{cases} \quad (1)$$

where $x_0, y(s) \in \mathbb{R}^d$, and

$$\alpha : [t_0, T] \rightarrow A \subseteq \mathbb{R}^m,$$

with T finite or $+\infty$. Under the assumption that the control is measurable, existence and uniqueness properties for the solution of (1) are ensured by the Carathéodory theorem:

Theorem 1 (Carathéodory) Assume that:

1. $f(\cdot, \cdot)$ is continuous.
2. There exists a positive constant $L_f > 0$ such that

$$|f(x, a) - f(y, a)| \leq L_f |x - y|,$$

for all $x, y \in \mathbb{R}^d$, $t \in \mathbb{R}^+$ and $a \in A$.

3. $f(x, \alpha(t))$ is measurable with respect to t .

Then, there is a unique absolutely continuous function $y : [t_0, T] \rightarrow \mathbb{R}^d$ that satisfies

$$y(s) = x_0 + \int_{t_0}^s f(y(\tau), \alpha(\tau)) d\tau. \quad (2)$$

which is interpreted as the solution of (1).

Note that the solution is continuous, but only a.e. differentiable, so it must be regarded as a weak solution of (1). By the theorem above, fixing a control in the set of admissible controls

$$\alpha \in \mathcal{A} := \{\alpha : [t_0, T] \rightarrow A, \text{ measurable}\}$$

yields a unique trajectory of (1) which is denoted by $y_{x_0, t_0}(s; \alpha)$. Changing the control policy generates a family of solutions of the controlled system (1) with index α . Since the dynamics (1) are “autonomous,” the initial time t_0 can be shifted to 0 by a change of variable. So to simplify the notation for autonomous dynamics, we can set $t_0 = 0$ and we denote this family by $y_{x_0}(s; \alpha)$ (or even write it as $y(s)$ if no ambiguity over the initial state or control arises). It is customary in dynamic programming, moreover, to use the notations x and t instead of x_0 and t_0 (since x and t appear as variables in the Hamilton–Jacobi–Bellman equation).

Optimal control problems require the introduction of a *cost functional* $J : \mathcal{A} \rightarrow \mathbb{R}$ which is used to select the “optimal trajectory” for (1). In the case of the infinite horizon problem, we set $t_0 = 0$, $x_0 = x$, and this functional is defined as

$$J_x(\alpha) = \int_0^\infty g(y_x(s, \alpha), \alpha(s)) e^{-\lambda s} ds \quad (3)$$

for a given $\lambda > 0$. The function g represents the

running cost and λ is the *discount factor*, which can be used to take into account the reduced value, at the initial time, of future costs. From a technical point of view, the presence of the discount factor ensures that the integral is finite whenever g is bounded. Note that one can also consider the undiscounted problem ($\lambda = 0$) provided the integral is still finite. The goal of optimal control is to find an optimal pair (y^*, α^*) that minimizes the cost functional. If we seek optimal controls in open-loop form, i.e., as functions of t , then the Pontryagin Maximum Principle furnishes necessary conditions for a pair (y^*, α^*) to be optimal.

A major drawback of an open-loop control is that being constructed as a function of time, it cannot take into account errors in the true state of the system, due, for example, to model errors or external disturbances, which may take the evolution far from the optimal forecasted trajectory. Another limitation of this approach is that a new computation of the control is required whenever the initial state is changed.

For these reasons, we are interested in the so-called *feedback controls*, that is, controls expressed as functions of the state of the system. Under feedback control, if the system trajectory is perturbed, the system reacts by changing its control strategy according to the change in the state. One of the main motivations for using the DPP is that it yields solutions to optimal control problems in the form of feedback controls.

DPP for the Infinite Horizon Problem

The starting point of dynamic programming is to introduce an auxiliary function, the *value function*, which for our problem is

$$v(x) = \inf_{\alpha \in \mathcal{A}} J_x(\alpha), \quad (4)$$

where, as above, x is the initial position of the system. The value function has a clear meaning: it is the optimal cost associated with the initial position x . This is a reference value which can be useful to evaluate the efficiency of a control – if $J_x(\bar{\alpha})$ is close to $v(x)$, this means that $\bar{\alpha}$ is “efficient.”

Bellman's dynamic programming principle provides a first characterization of the value function.

Proposition 1 (DPP for the infinite horizon problem) *Under the assumptions of Theorem 1, for all $x \in \mathbb{R}^d$ and $\tau > 0$,*

$$v(x) = \inf_{\alpha \in \mathcal{A}} \left\{ \int_0^\tau g(y_x(s; \alpha), \alpha(s)) e^{-\lambda s} ds + e^{-\lambda \tau} v(y_x(\tau; \alpha)) \right\}. \quad (5)$$

Proof. Denote by $\bar{v}(x)$ the right-hand side of (5). First, we remark that for any $x \in \mathbb{R}^d$ and $\bar{\alpha} \in \mathcal{A}$,

$$\begin{aligned} J_x(\bar{\alpha}) &= \int_0^\infty g(\bar{y}(s), \bar{\alpha}(s)) e^{-\lambda s} ds \\ &= \int_0^\tau g(\bar{y}(s), \bar{\alpha}(s)) e^{-\lambda s} ds \\ &\quad + \int_\tau^\infty g(\bar{y}(s), \bar{\alpha}(s)) e^{-\lambda s} ds \\ &= \int_0^\tau g(\bar{y}(s), \bar{\alpha}(s)) e^{-\lambda s} ds + e^{-\lambda \tau} \\ &\quad \times \int_0^\infty g(\bar{y}(s + \tau), \bar{\alpha}(s + \tau)) e^{-\lambda s} ds \\ &\geq \int_0^\tau g(\bar{y}(s), \bar{\alpha}(s)) e^{-\lambda s} ds + e^{-\lambda \tau} v(\bar{y}(\tau)) \end{aligned}$$

(here, $y_x(s, \bar{\alpha})$ is abbreviated as $\bar{y}(s)$). Taking the infimum over all trajectories, first over the right-hand side and then the left of this inequality, yields

$$v(x) \geq \bar{v}(x) \quad (6)$$

To prove the opposite inequality, we recall that $\bar{v}$ is defined as an infimum, and so, for any $x \in \mathbb{R}^d$ and $\varepsilon > 0$, there exists a control $\bar{\alpha}_\varepsilon$ (and the corresponding evolution $\bar{y}_\varepsilon$) such that

$$\bar{v}(x) + \varepsilon \geq \int_0^\tau g(\bar{y}_\varepsilon(s), \bar{\alpha}_\varepsilon(s)) e^{-\lambda s} ds + e^{-\lambda \tau} v(\bar{y}_\varepsilon(\tau)). \quad (7)$$

On the other hand, the value function v being also defined as an infimum, for any $x \in \mathbb{R}^d$ and $\varepsilon > 0$, there exists a control $\tilde{\alpha}_\varepsilon$ such that

$$v(\bar{y}_\varepsilon(\tau)) + \varepsilon \geq J_{\bar{y}_\varepsilon(\tau)}(\tilde{\alpha}_\varepsilon). \quad (8)$$

Inserting (8) in (7), we get

$$\begin{aligned} \bar{v}(x) &\geq \int_0^\tau g(\bar{y}_\varepsilon(s), \tilde{\alpha}_\varepsilon(s)) e^{-\lambda s} ds \\ &\quad + e^{-\lambda \tau} J_{\bar{y}_\varepsilon(\tau)}(\tilde{\alpha}_\varepsilon) - (1 + e^{-\lambda \tau}) \varepsilon \\ &\geq J_x(\hat{\alpha}) - (1 + e^{-\lambda \tau}) \varepsilon \\ &\geq v(x) - (1 + e^{-\lambda \tau}) \varepsilon, \end{aligned} \quad (9)$$

where $\hat{\alpha}$ is a control defined by

$$\hat{\alpha}(s) = \begin{cases} \tilde{\alpha}_\varepsilon(s) & 0 \leq s < \tau \\ \tilde{\alpha}_\varepsilon(s - \tau) & s \geq \tau. \end{cases} \quad (10)$$

(Note that $\hat{\alpha}(\cdot)$ is still measurable). Since ε is arbitrary, (9) finally yields $\bar{v}(x) \geq v(x)$. $\square$

We observe that this proof crucially relies on the fact that the control defined by (10) still belongs to $\mathcal{A}$, being a measurable control. The possibility of obtaining an admissible control by joining together two different measurable controls is known as the *concatenation property*.

The Hamilton–Jacobi–Bellman Equation

The DPP can be used to characterize the value function in terms of a nonlinear partial differential equation. In fact, let $\alpha^* \in \mathcal{A}$ be the optimal control, and y^* the associated evolution (to simplify, we are assuming that the infimum is a minimum). Then,

$$v(x) = \int_0^\tau g(y^*(s), \alpha^*(s)) e^{-\lambda s} ds + e^{-\lambda \tau} v(y^*(\tau)),$$

that is,

$$v(x) - e^{-\lambda \tau} v(y^*(\tau)) = \int_0^\tau g(y^*(s), \alpha^*(s)) e^{-\lambda s} ds$$

so that adding and subtracting $e^{-\lambda \tau} v(x)$ and dividing by τ , we get

$$e^{-\lambda\tau} \frac{(v(x) - v(y^*(\tau)))}{\tau} + \frac{v(x)(1 - e^{-\lambda\tau})}{\tau} \\ = \frac{1}{\tau} \int_0^\tau g(y^*(s), \alpha^*(s)) e^{-\lambda s} ds.$$

Assume now that v is regular. By passing to the limit as $\tau \rightarrow 0^+$, we have

$$\lim_{\tau \rightarrow 0^+} -\frac{v(y^*(\tau)) - v(x)}{\tau} \\ = -Dv(x) \cdot \dot{y}^*(x) = -Dv(x) \cdot f(x, \alpha^*(0)) \\ \lim_{\tau \rightarrow 0^+} v(x) \frac{(1 - e^{-\lambda\tau})}{\tau} = \lambda v(x) \\ \lim_{\tau \rightarrow 0^+} \frac{1}{\tau} \int_0^\tau g(y^*(s), \alpha^*(s)) e^{-\lambda s} ds = g(x, \alpha^*(0))$$

where we have assumed that $\alpha^*(\cdot)$ is continuous at 0. Then, we can conclude

$$\lambda v(x) - Dv(x) \cdot f(x, \alpha^*) - g(x, \alpha^*) = 0 \quad (11)$$

where $\alpha^* = \alpha^*(0)$. Similarly, using the equivalent form

$$v(x) + \sup_{\alpha \in \mathcal{A}} \left\{ -\int_0^\tau g(y(s), \alpha(s)) e^{-\lambda s} ds \right. \\ \left. - e^{-\lambda\tau} v(y(\tau)) \right\} = 0$$

of the DPP and the inequality, this implies for any (continuous at 0) control $\alpha \in \mathcal{A}$,

$$\lambda v(x) - Dv(x) \cdot f(x, \alpha) - g(x, \alpha) \\ \leq 0, \quad \text{for every } \alpha \in \mathcal{A}. \quad (12)$$

Combining (11) and (12), we obtain the *Hamilton–Jacobi–Bellman equation* (or *dynamic programming equation*):

$$\lambda u(x) + \sup_{\alpha \in A} \{-f(x, \alpha) \cdot Du(x) - g(x, \alpha)\} = 0, \quad (13)$$

which characterizes the value function for the infinite horizon problem associated with minimizing (3). Note that given x , the value of a achieving the max (assuming it exists) corresponds to the control $a^* = \alpha^*(0)$, and this makes

it natural to interpret the argmax in (13) as the optimal feedback at x (see Bardi and Capuzzo Dolcetta (1997) for more details).

In short, (13) can be written as

$$H(x, u, Du) = 0$$

with $x \in \mathbb{R}^d$, and

$$H(x, u, p) = \lambda u(x) + \sup_{a \in A} \{-f(x, a) \cdot p - g(x, a)\}. \quad (14)$$

Note that $H(x, u, \cdot)$ is convex (being the sup of a family of linear functions) and that $H(x, \cdot, p)$ is monotone (since $\lambda > 0$). It is also easy to see that the solution u is not differentiable even when f and g are smooth functions (i.e., $f, g, \in C^\infty(\mathbb{R}^n, A)$), so we need to deal with weak solution of the Bellman equation. This can be done in the framework of viscosity solutions, a theory initiated by Crandall and Lions in the 1980s which has been successfully applied in many areas as optimal control, fluid dynamics, and image processing (see the books Barles (1994) and Bardi and Capuzzo Dolcetta (1997) for an extended introduction and numerous applications to optimal control). Typically viscosity solutions are Lipschitz continuous solutions so they are differentiable almost everywhere.

An Extension to the Minimum Time Problem

In the minimum time problem, we want to minimize the time of arrival of the state on a given *target set* $\mathcal{T}$. We will assume that $\mathcal{T} \subset \mathbb{R}^d$ is a closed set. Then our cost functional will be given by

$$J(x, \alpha) = t_x(\alpha)$$

where

$$t_x(\alpha) := \begin{cases} \min\{t : y_x(t, \alpha) \in \mathcal{T}\} & \text{if } y_x(t, \alpha) \in \mathcal{T} \\ & \text{for some } t \geq 0 \\ +\infty & \text{if } y_x(t, \alpha) \notin \mathcal{T} \\ & \text{for any } t \geq 0 \end{cases}$$

The corresponding value function is called the *minimum time function*

$$T(x) := \inf_{\alpha(\cdot) \in \mathcal{A}} t_x(\alpha(\cdot)). \quad (15)$$

The main difference with respect to the previous problem is that now the value function T will be finite valued only on a subset $\mathcal{R}$ which depends on the target, on the dynamics, and on the set of admissible controls.

Definition 1 The reachable set $\mathcal{R}$ is defined by

$$\mathcal{R} := \cup_{t>0} \mathcal{R}(t) = \{x \in \mathbb{R}^n : T(x) < +\infty\}$$

where, for $t > 0$, $\mathcal{R}(t) := \{x \in \mathbb{R}^n : T(x) < t\}$. The meaning is clear: $\mathcal{R}$ is the set of initial points which can be driven to the target in finite time. The system is said to be *controllable* on $\mathcal{T}$ if for all $t > 0$, $\mathcal{T} \subset \text{int}(\mathcal{R}(t))$ (here, $\text{int}(D)$ denotes the interior of the set D). Assuming controllability in a neighborhood of the target one gets the continuity of the minimum time function and under the assumptions made on f , A , and $\mathcal{T}$, one can prove some interesting properties:

- (i) $\mathcal{R}$ is open.
- (ii) T is continuous on $\mathcal{R}$.
- (iii) $\lim_{x \rightarrow x_0} T(x) = +\infty$, for any $x_0 \in \partial \mathcal{R}$.

Now let us denote by $\mathcal{X}_D$ the characteristic function of the set D . Using in $\mathcal{R}$ arguments similar to the proof of DPP in the previous section one can obtain the following DPP:

Proposition 2 (DPP for the minimum time problem) For any $x \in \mathcal{R}$, the value function satisfies

$$T(x) = \inf_{\alpha \in \mathcal{A}} \{t \wedge t_x(\alpha) + \mathcal{X}_{\{t \leq t_x(\alpha)\}} T(y_x(t, \alpha))\} \quad (16)$$

for any $t \geq 0$

and

$$T(x) = \inf_{\alpha \in \mathcal{A}} \{t + T(y_x(t, \alpha))\} \quad (17)$$

for any $t \in [0, T(x)]$

From the previous DPP, one can also obtain the following characterization of the minimum time function.

Proposition 3 Let $\mathcal{R} \setminus \mathcal{T}$ be open and $T \in C(\mathcal{R} \setminus \mathcal{T})$, then T is a viscosity solution of

$$\max_{a \in A} \{-f(x, a) \cdot \nabla T(x)\} = 1 \quad x \in \mathcal{R} \setminus \mathcal{T} \quad (18)$$

coupled with the natural boundary condition

$$\begin{cases} T(x) = 0 & x \in \partial \mathcal{T} \\ \lim_{x \rightarrow \partial \mathcal{R}} T(x) = +\infty \end{cases}$$

By the change of variable $v(x) = 1 - e^{-T(x)}$, one can obtain a simpler problem getting rid of the boundary condition on $\partial \mathcal{R}$ (which is unknown). The new function v will be the unique viscosity solution of an external Dirichlet problem (see Bardi and Capuzzo Dolcetta (1997) for more details), and the reachable set can be recovered a posteriori via the relation $\mathcal{R} = \{x \in \mathbb{R}^d : v(x) < 1\}$.

Further Extensions and Related Topics

The DPP has been extended from deterministic control problems to many other problems. In the framework of stochastic control problems where the dynamics are given by a diffusion, the characterization of the value function obtained via the DPP leads to a second-order Hamilton–Jacobi–Bellman equation (Fleming and Soner 1993; Kushner and Dupuis 2001). Another interesting extension has been made in differential games where the DPP is based on the delicate notion of nonanticipative strategies for the players and leads to a nonconvex nonlinear partial differential equation (the Isaacs equation) (Bardi and Capuzzo Dolcetta 1997). For a short introduction to numerical methods based on DP and exploiting the so-called “value iteration,” we refer the interested reader to the

Appendix A in Bardi and Capuzzo Dolcetta (1997) and to Kushner and Dupuis (2001) (see also the book Howard (1960) for the “policy iteration”).

Cross-References

- [Numerical Methods for Nonlinear Optimal Control Problems](#)
- [Optimal Control and Pontryagin’s Maximum Principle](#)

Bibliography

- Bardi M, Capuzzo Dolcetta I (1997) Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations. Birkhäuser, Boston
- Barles G (1994) Solutions de viscosité des équations de Hamilton-Jacobi. In: *Mathématiques et applications*, vol 17. Springer, Paris
- Bellman R (1957) Dynamic programming. Princeton University Press, Princeton
- Bertsekas DP (1987) Dynamic programming: deterministic and stochastic models. Prentice Hall, Englewood Cliffs
- Boltanskii VG, Gamkrelidze RV, Pontryagin LS (1956) On the theory of optimal processes (in Russian). *Doklady Akademii Nauk SSSR* 110, 7–10
- Fleming WH, Rishel RW (1975) Deterministic and stochastic optimal control. Springer, New York
- Fleming WH, Soner HM (1993) Controlled Markov processes and viscosity solutions. Springer, New York
- Howard RA (1960) Dynamic programming and Markov processes. Wiley, New York
- Kushner HJ, Dupuis P (2001) Numerical methods for stochastic control problems in continuous time. Springer, Berlin
- Macki J, Strauss A (1982) Introduction to optimal control theory. Springer, Berlin/Heidelberg/New York
- Pontryagin LS, Boltanskii VG, Gamkrelidze RV, Mishchenko EF (1961) *Matematicheskaya teoriya optimal’nykh protsessov*. Fizmatgiz, Moscow. Translated into English. The mathematical theory of optimal processes. John Wiley and Sons (Interscience Publishers), New York, 1962
- Ross IM (2009) A primer on Pontryagin’s principle in optimal control. Collegiate Publishers, San Francisco

Optimal Control via Factorization and Model Matching

Michael Cantoni

Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, VIC, Australia

Abstract

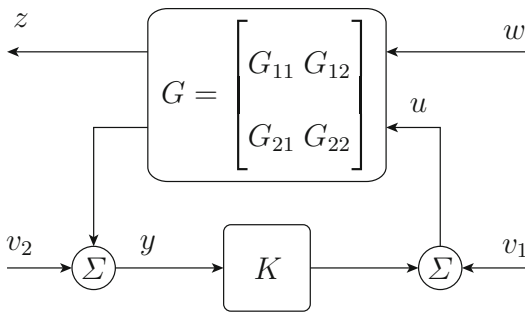
One approach to linear control system design involves the matching of input-output models with respect to a quantification of performance. The approach is based on a parametrization of all stabilizing feedback controllers for the given plant model. This parametrization, constructed from coprime factorizations of the plant, and spectral factorization methods for solving model-matching problems, are described in this article. Both impulse-response energy and worst-case energy-gain measures of controller performance are considered.

Keywords

Coprime factorization · $\mathcal{H}_2$ control · $\mathcal{H}_\infty$ control · Spectral factorization · Youla-Kučera controller parametrization

Introduction

Various linear control system design problems can be formulated in terms of the interconnection shown in Fig. 1; e.g., see Francis and Doyle (1987), Boyd and Barratt (1991), and Zhou et al. (1996). The linear system K is a *controller* (with input y and output $u - v_1$) to be designed for the given *generalized plant* model G . The generalized plant is constructed so that controller *performance* (i.e., the quality of K relative to



Optimal Control via Factorization and Model Matching, Fig. 1 Standard interconnection for controller design

specifications) can be quantified as a nonnegative functional of

$$H(G, K) = G_{11} + G_{12}K(I - G_{22}K)^{-1}G_{21}, \quad (1)$$

which relates the input w and the output z when $v_1 = 0$ and $v_2 = 0$. The control objective is to select K to minimize, or achieve a specified upper bound on, this measure of performance. Typically, *internal stability* is also required, which pertains to the signals v_1 and v_2 as discussed further below. The so-called $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems are well-known examples. In the $\mathcal{H}_2$ problem, performance is quantified as the energy (resp. power) of z when w is impulsive (resp. unit white noise). In the $\mathcal{H}_\infty$ problem, performance is quantified as the worst-case energy gain from w to z , which can be used to reflect robustness to model uncertainty, among other things; see Zhou et al. (1996).

The special case of $G_{22} = 0$ leads directly to a (weighted) *model-matching* problem, as the performance map becomes $H(G, K) = G_{11} + G_{12}KG_{21}$, which exhibits *affine* dependence on K . As such, the design of K can be interpreted in terms of matching $G_{12}KG_{21}$ to $-G_{11}$ with respect to the scalar quantification of performance. It turns out that every internally stabilizable problem with $G_{22} \neq 0$ can also

be converted into a model-matching problem. The key ingredients in this transformation are coprime factorizations of the plant model. The role of these and other factorizations in a model-matching approach to $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems is the focus of this entry.

For the sake of argument, finite-dimensional linear time-invariant systems are considered in terms of real-rational transfer functions in the *frequency domain*, as the existence of all factorizations employed is well understood in this setting. Indeed, constructions via state-space realizations of such transfer functions and corresponding Riccati equations are well known (Zhou et al. 1996). The merits of the model-matching approach pursued here are at least twofold: (i) the underlying algebraic input-output (i.e., transfer function) perspective extends to more abstract settings, including classes of distributed-parameter and time-varying systems (Desoer et al. 1980; Vidyasagar 1985; Curtain and Zwart 1995; Feintuch 1998; Quadrat 2006); and (ii) model matching is a convex problem for various measures of performance (including mixed indexes) and controller constraints. The latter can be exploited to devise numerical algorithms for controller optimization (Boyd and Barratt 1991; Dahleh and Diaz-Bobillo 1995; Qi et al. 2004).

First, some notational conventions regarding transfer functions and two measures of performance for control system design are defined. Coprime factorizations are then described within the context of a well-known parametrization of stabilizing controllers, originally discovered by Youla et al. (1976) and Kucera (1975). This yields an affine parametrization of performance maps for problems in standard form, and thus, a transformation to a model-matching problem. Finally, the role of spectral factorizations in solving model-matching problems with respect to impulse-response energy ($\mathcal{H}_2$) and worst-case energy-gain ($\mathcal{H}_\infty$) measures of performance is described.

Notation and Nomenclature

$\mathcal{R}$ generically denotes a linear space of matrices with given row and column dimensions (not denoted explicitly) and entries that are *proper* real-rational functions of the complex variable s ; i.e., $(\sum_{k=1}^m b_k s^k) / (\sum_{k=1}^n a_k s^k)$ for sets of real coefficients $\{a_k\}_{k=1}^n$ and $\{b_k\}_{k=1}^m$ with $m \leq n < \infty$. Henceforth, the compatibility of matrix dimensions is implicitly assumed in the matrix sums and products that arise. All matrices in $\mathcal{R}$ have (non-unique) “state-space” realizations of the form $C(sI - A)^{-1}B + D$, where A , B , C , and D are matrices with real valued entries. This form naturally arises in frequency-domain analysis of the *input-output* map associated with the time-domain model $\dot{x}(t) = Ax(t) + Bu(t)$, with initial condition $x(0) = 0$ and output equation $y(t) = Cx(t) + Du(t)$, where $\dot{x}$ denotes the time derivative of x and u is the input. The study of such linear time-invariant differential equation models via the Laplace transform and *multiplication* by real-rational transfer function matrices is fundamental in linear systems theory (Kailath 1980; Francis 1987; Zhou et al. 1996). $P \in \mathcal{R}$ has an inverse $P^{-1} \in \mathcal{R}$ (i.e., $PP^{-1} = P^{-1}P = I$ where I denotes the identity matrix) if and only if $\lim_{|s| \rightarrow \infty} P(s)$ is a nonsingular matrix (i.e., the matrix D in a state-space realization is invertible). The superscripts T and $*$ denote the transpose and complex conjugate transpose. For a matrix $Z = Z^*$ with complex entries, $Z > 0$ means $z^*Zz \geq \epsilon z^*z$ for some $\epsilon > 0$ and all complex vectors z of compatible dimension. $P^\sim(s) := P(-s)^T$, whereby $(P(j\omega))^* = P^\sim(j\omega)$ for all real ω with $j := \sqrt{-1}$. Zeros of transfer function denominators are called poles.

In subsequent sections, several subspaces of $\mathcal{R}$ are used to define and solve two standard linear control problems. The subspace $\mathcal{B} \subset \mathcal{R}$ comprises transfer functions that have no poles on the imaginary axis in the complex plane. For $P \in \mathcal{B}$, the scalar performance index

$$\|P\|_\infty := \max_{-\infty \leq \omega \leq \infty} \bar{\sigma}(P(j\omega)) \geq 0$$

is finite; the real number $\bar{\sigma}(Z)$ is the maximum singular value of the matrix argument Z . This index measures the worst-case energy-gain from an input signal u to the output signal $y = Pu$. Note that $\|P\|_\infty < \gamma$ if and only if $\gamma^2 I - P^\sim(j\omega)P(j\omega) > 0$ for all $-\infty \leq \omega \leq \infty$.

The subspace $\mathcal{S} \subset \mathcal{B} \subset \mathcal{R}$ consists of transfer functions that have no poles with positive real part. A transfer function in $\mathcal{S}$ is called *stable* because the corresponding input-output map is causal in the time domain, as well as bounded-in-bounded-out (in various senses). If $P \in \mathcal{S}$ is such that $P^\sim P = I$, then it is called *inner*. If $P, P^{-1} \in \mathcal{S}$, then both are called *outer*.

The symbol $\mathcal{L}$ denotes the subspace of *strictly-proper* transfer functions in $\mathcal{B}$; i.e., for every matrix entry the degree n of the denominator *exceeds* the degree m of the numerator. Observe that $P \in \mathcal{L}$ if and only if $P^\sim \in \mathcal{L}$ if and only if $P \in \mathcal{B}$ and $\lim_{|s| \rightarrow \infty} P(s) = 0$. Further, note that $P_1 P_2 \in \mathcal{L}$ and $P_3 P_1 \in \mathcal{L}$ for all $P_1 \in \mathcal{L}$ and $P_i \in \mathcal{B}, i = 2, 3$. The inner-product defined for every $P, P_1 \in \mathcal{L}$ by

$$\langle P_1, P \rangle := \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace}(P_1^\sim(j\omega)P(j\omega))d\omega < \infty$$

induces the scalar performance index $\|P\|_2 := \sqrt{\langle P, P \rangle} \geq 0$. This is equal to the root-mean-square (energy) measure of the impulse response, and also to the covariance (power) of the output signal $y = Pu$ when the input signal u is unit white noise. By the properties $\text{trace}(Z_1 + Z_2) = \text{trace}(Z_1) + \text{trace}(Z_2)$ and $\text{trace}(Z_1 Z_2) = \text{trace}(Z_2 Z_1)$ of the matrix trace operator, it follows that $\langle P_1 + P_2, P_3 \rangle = \langle P_1, P_3 \rangle + \langle P_2, P_3 \rangle$ and

$$\langle P_1, P_2 P_3 \rangle = \langle P_2^\sim P_1, P_3 \rangle$$

$$= \langle P_1 P_3^\sim, P_2 \rangle$$

$$= \langle P_3^\sim, P_1^\sim P_2 \rangle$$

$$\text{for } P_i \in \mathcal{L}, i = 1, 2, 3. \quad (2)$$

The subspace $\mathcal{L} \subset \mathcal{B} \subset \mathcal{R}$ can be expressed as the direct sum $\mathcal{L} = \mathcal{H} + \mathcal{H}_\perp$, where $\mathcal{H} = \mathcal{L} \cap \mathcal{S}$ and $\mathcal{H}_\perp$ is the subspace of transfer functions in $\mathcal{L}$ that have no poles with negative real part. That is, given $P \in \mathcal{L}$, there is a unique decomposition $P = \Pi_+(P) + \Pi_-(P)$, with $\Pi_+(P) \in \mathcal{H}$ and $\Pi_-(P) \in \mathcal{H}_\perp$. Observe that $P \in \mathcal{H}$ if and only if $P^\sim \in \mathcal{H}_\perp$. It can be shown via Plancherel's theorem that $\langle P_1, P_2 \rangle = 0$ for $P_1 \in \mathcal{H}_\perp$ and $P_2 \in \mathcal{H}$. Finally, note that $P_1 P_2 \in \mathcal{H}$ and $P_3 P_1 \in \mathcal{H}$ for all $P_i \in \mathcal{H}$ and $P_i \in \mathcal{S}$, $i = 2, 3$.

called an *inner-outer factorization*, and $P^\sim P = (M^{-1})^\sim M^{-1}$ is called a *spectral factorization*, since $M, M^{-1} \in \mathcal{S}$. More generally, if $\mathcal{E} = \mathcal{E}^\sim \in \mathcal{B}$ satisfies $\mathcal{E}(s) > 0$ for s on the extended imaginary axis, then there exists a (non-unique) spectral factor $\Sigma, \Sigma^{-1} \in \mathcal{S}$ such that $\mathcal{E} = \Sigma^\sim \Sigma$. Similarly, there exists a co-spectral factor $\tilde{\Sigma}, \tilde{\Sigma}^{-1} \in \mathcal{S}$ such that $\mathcal{E} = \tilde{\Sigma} \tilde{\Sigma}^\sim$. State-space realization-based constructions via the solution of Riccati equations can be found in Zhou et al. (1996, Chapter 13), for example.

Coprime and Spectral Factorizations

Given $P \in \mathcal{R}$, the factorizations $P = NM^{-1} = \tilde{M}^{-1}\tilde{N}$ are said to be *doubly coprime* over $\mathcal{S}$ provided $N, M, \tilde{N}, \tilde{M}$ are all in $\mathcal{S}$ and there exist $U_0, V_0, \tilde{U}_0, \tilde{V}_0$ in $\mathcal{S}$ such that $\tilde{V}_0 U_0 = \tilde{U}_0 V_0$,

$$\begin{bmatrix} \tilde{V}_0 & -\tilde{U}_0 \end{bmatrix} \begin{bmatrix} M \\ N \end{bmatrix} = I \quad \text{and} \quad \begin{bmatrix} -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} U_0 \\ V_0 \end{bmatrix} = I \quad (3)$$

all hold; i.e., $\begin{bmatrix} M^T & N^T \end{bmatrix}$ and $\begin{bmatrix} -\tilde{N} & \tilde{M} \end{bmatrix}$ are right invertible in $\mathcal{S}$. Importantly, if the factorizations are coprime and $P \in \mathcal{S}$, then $M^{-1} = \tilde{V}_0 - \tilde{U}_0 P$ and $\tilde{M}^{-1} = V_0 - P U_0$ are in $\mathcal{S}$, as sums of products of transfer functions in $\mathcal{S}$; i.e., M and $\tilde{M}$ are outer. Doubly coprime factorizations over $\mathcal{S}$ always exist, but these are not unique. Constructions from state-space realizations can be found in Zhou et al. (1996, Chapter 6) and Francis (1987), for example. As mentioned above, coprime factorizations play a role in transforming a general controller synthesis problem into the special model matching problem. Specifically, such factorizations underlie the Youla-Kučera parametrization of internally stabilizing controllers presented in the next section.

Subsequently, a special coprime factorization proves to be useful. If $P^\sim(s)P(s) = (M^\sim(s))^{-1}N^\sim(s)N(s)M^{-1}(s) > 0$ for s on the extended imaginary axis (i.e., for $s = j\omega$ with $-\infty \leq \omega \leq \infty$), then it is possible to choose the factor N to be inner. In this case, if P is also an element of $\mathcal{S}$, then $P = NM^{-1}$ is

Affine Controller/Performance-Map Parametrization

With reference to Fig. 1, a generalized plant model $G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \in \mathcal{R}$ is said to be *internally stabilizable* if there exists $K \in \mathcal{R}$ such that the nine transfer functions associated with the map from the vector of signals (w, v_1, v_2) to the vector of signals (z, u, y) are all elements of $\mathcal{S}$; only one of these is the performance map $H(G, K) = G_{11} + G_{12}K(I - G_{22}K)^{-1}G_{21}$. Accounting in this way for the influence of the fictitious signals v_1 and v_2 , and the behavior of the internal signals u and y , amounts to the following requirement: Given minimal state-space realizations, any nonzero initial condition response decays exponentially in the time domain when G and K are interconnected according to Fig. 1 with $w = 0$, $v_1 = 0$ and $v_2 = 0$. Not every $G \in \mathcal{R}$ is internally stabilizable in the sense just defined; for example, take G_{11} to have a pole with positive real part and $G_{21} = G_{12} = G_{22} = 0$. A necessary condition for stabilizability is $(I - G_{22}K)^{-1} \in \mathcal{R}$; i.e., the inverse must be proper. The latter always holds if G_{22} is strictly proper, as assumed henceforth to simplify the presentation. It is also assumed that G is internally stabilizable.

It can be shown that G is internally stabilized by K if and only if the standard feedback interconnection of G_{22} and K , corresponding to $w = 0$ in Fig. 1, is internally stable. That is, if and only if the transfer function

$$\begin{bmatrix} I & -K \\ -G_{22} & I \end{bmatrix} \in \mathcal{R}, \quad (4)$$

which relates u and y to v_1 and v_2 in line with the summing junctions shown at the interconnection points, has an inverse in $\mathcal{S}$; see Francis (1987, Theorem 4.2). Substituting the coprime factorizations $K = UV^{-1} = \tilde{V}^{-1}\tilde{U}$ and $G_{22} = NM^{-1} = \tilde{M}^{-1}\tilde{N}$, it follows that the inverse of (4) is an element of $\mathcal{S}$ if and only if

$$\begin{bmatrix} M & U \\ N & V \end{bmatrix}^{-1} \in \mathcal{S} \quad \Leftrightarrow \quad \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \in \mathcal{S}. \quad (5)$$

The equivalent characterizations of internal stability in (5) lead directly to affine parametrizations of controllers and performance maps. Specifically, following the approach of Desoer et al. (1980), Vidyasagar (1985), and Francis (1987), suppose that the factorizations $G_{22} = NM^{-1} = \tilde{M}^{-1}\tilde{N}$ are *doubly coprime* in the sense that (3) holds for some $U_0, V_0, \tilde{U}_0, \tilde{V}_0 \in \mathcal{S}$ with $\tilde{V}_0 U_0 = \tilde{U}_0 V_0$. Then it follows that

$$\begin{aligned} \begin{bmatrix} \tilde{V}_0 & -\tilde{U}_0 \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} M & U_0 \\ N & V_0 \end{bmatrix} &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} M & U_0 \\ N & V_0 \end{bmatrix} \begin{bmatrix} \tilde{V}_0 & -\tilde{U}_0 \\ -\tilde{N} & \tilde{M} \end{bmatrix}, \end{aligned} \quad (6)$$

since $0 = G_{22} - G_{22} = \tilde{M}^{-1}(\tilde{M}N - \tilde{N}M)M^{-1}$. Now using (6) and the internal stability condition (5), it can be shown that $K = UV^{-1}$ stabilizes G_{22} if and only if

$$U = (U_0 - MQ) \text{ and } V = (V_0 - NQ) \text{ with } Q \in \mathcal{S}.$$

Similarly, K stabilizes G_{22} if and only if $K = (\tilde{V}_0 - Q\tilde{N})^{-1}(\tilde{U}_0 - Q\tilde{M})$ with $Q \in \mathcal{S}$. Together, these constitute the so-called Youla-Kučera parametrizations of internally stabilizing controllers for G_{22} . Importantly, the coprime factors of the controller are affine functions of the stable but otherwise free parameter Q . Moreover, using (6), an affine parametrization of

the standard performance map (1) holds by direct substitution of either controller parametrization. Specifically,

$$\begin{aligned} H(G, K) &= G_{11} + G_{12}K(I - G_{22}K)^{-1}G_{21} \\ &= T_1 + T_2QT_3 \quad \text{with } Q \in \mathcal{S}, \end{aligned} \quad (7)$$

where $T_1 = G_{11} + G_{12}U_0\tilde{M}G_{21}$, $T_2 = -G_{12}M$ and $T_3 = \tilde{M}G_{21}$. Clearly, $T_1 \in \mathcal{S}$ since this is the performance map when $Q = 0 \in \mathcal{S}$. By the assumption that G is stabilizable, it follows that T_2 and T_3 are also elements of $\mathcal{S}$; see Francis (1987, Chapter 4). The so-called Q -parametrization of stable performance maps in (7) motivates the subsequent consideration of model-matching problems with respect to the standard measures of control system performance given by $\|\cdot\|_2$ and $\|\cdot\|_\infty$.

Model-Matching via Spectral Factorization

Bearing in mind the Q -parametrization (7), consider the following $\mathcal{H}_2$ model-matching problem, where $\inf$ denotes greatest lower bound (infimum) and $T_i \in \mathcal{S}$, $i = 1, 2, 3$:

$$\inf_{Q \in \mathcal{S}} \|T_1 + T_2QT_3\|_2. \quad (8)$$

Assume that $T_2(s)$ and $T_3(s)$ have full column and row rank, respectively, for s on the extended imaginary axis and that T_1 is strictly proper. Then Q must be strictly proper, and thus an element of $\mathcal{H} \subset \mathcal{S}$, in order for the $\|\cdot\|_2$ measure of performance to be finite. Under this standard collection of assumptions, the infimum in (8) is achieved, as shown below.

By the definition of $\|\cdot\|_2$, a minimizer of the nonnegative convex functional $f := Q \in \mathcal{H} \mapsto \langle (T_1 + T_2QT_3), (T_1 + T_2QT_3) \rangle$ is a solution of the model matching problem. Given spectral factorizations $\Phi^* \Phi = T_2^* T_2$ and $\Lambda \Lambda^* = T_3 T_3^*$, which exist by the rank assumptions on T_2 and T_3 , define $R := \Phi Q \Lambda$ and $W := \Phi^* T_2^* T_1 T_3^* \Lambda^*$. Then for $Q \in \mathcal{H}$,

which is equivalent to $R \in \mathcal{H}$ by the property $\Phi, \Phi^{-1}, \Lambda, \Lambda^{-1} \in \mathcal{S}$ of the spectral factors, it follows that

$$\begin{aligned} f(Q) &= \langle T_1, T_1 \rangle + \langle \Phi^{-\sim} T_2^{\sim} T_1 T_3^{\sim} \Lambda^{-\sim}, R \rangle \\ &\quad + \langle R, \Phi^{-\sim} T_2^{\sim} T_1 T_3^{\sim} \Lambda^{-\sim} \rangle + \langle R, R \rangle \\ &= \langle T_1, T_1 \rangle + \langle W + R, W + R \rangle \\ &\quad - \langle W, W \rangle \\ &= \langle T_1, T_1 \rangle - \langle \Pi_+(W), \Pi_+(W) \rangle \\ &\quad + \langle (\Pi_+(W) + R), (\Pi_+(W) + R) \rangle, \end{aligned} \quad (9)$$

where the second equality holds by “completion-of-squares” and the last equality holds since $\langle \Pi_+(W), \Pi_-(W) \rangle = 0 = \langle R, \Pi_-(W) \rangle$. From (9) it is apparent that

$$Q = -\Phi^{-1} \Pi_+(\Phi^{-\sim} T_2^{\sim} T_1 T_3^{\sim} \Lambda^{-\sim}) \Lambda^{-1}$$

is a minimizer of f . As above, spectral factorization is a key component of the so-called Wiener-Hopf approach of Youla et al. (1976) and DeSantis et al. (1978).

Now consider the $\mathcal{H}_\infty$ model-matching problem

$$\inf_{Q \in \mathcal{S}} \|T_1 + T_2 Q T_3\|_\infty,$$

given $T_i \in \mathcal{S}$ for $i = 1, 2, 3$. This is more challenging than the problem (8). While sufficient conditions are available for the infimum to be achieved, computing a minimizer is generally difficult; see Francis and Doyle (1987) and Glover et al. (1991). In view of this, nearly optimal solutions are often sought by considering the relaxed problem of finding the set of $Q \in \mathcal{S}$ that satisfy $\|T_1 + T_2 Q T_3\|_\infty < \gamma$ for a value of $\gamma > 0$ greater than, but close to, the infimum.

With a view to highlighting the role of factorization methods and simplifying the presentation, suppose that T_2 is inner, which is possible without loss of generality via inner-outer factorization when $T_2(s)$ has full column rank for s on the extended imaginary axis. Further, assume that $T_3 = I$. Now following the approach

of Francis (1987) and Green et al. (1990), let $X^{\sim} = [X_1^{\sim} \ X_2^{\sim}] := [T_2 \ I - T_2 T_2^{\sim}] \in \mathcal{B}$, so that $X^{\sim} X = I$ and $X T_2 = \begin{bmatrix} I \\ 0 \end{bmatrix}$. Then,

$$\begin{aligned} \|T_1 + T_2 Q\|_\infty &= \|X(T_1 + T_2 Q)\|_\infty \\ &= \left\| \begin{bmatrix} T_2^{\sim} T_1 + Q \\ (I - T_2 T_2^{\sim}) T_1 \end{bmatrix} \right\|_\infty < \gamma \end{aligned} \quad (10)$$

if and only if

$$\begin{aligned} 0 &< \gamma^2 I - T_1^{\sim} (I - T_2 T_2^{\sim}) T_1 \\ &\quad - (T_2^{\sim} T_1 + Q)^{\sim} (T_2^{\sim} T_1 + Q) \end{aligned} \quad (11)$$

on the extended imaginary axis. Observing that (11) implies $0 < \gamma^2 I - T_1^{\sim} (I - T_2 T_2^{\sim}) T_1$, it can be shown that there exists $Q \in \mathcal{S}$ such that (10) holds if and only if the following are both satisfied: (a) there exists spectral factorization $\gamma^2 \Psi^{\sim} \Psi = \gamma^2 I - T_1^{\sim} (I - T_2 T_2^{\sim}) T_1$; and (b) there exists $\bar{R} (= Q \Psi^{-1}) \in \mathcal{S}$ such that $\|\bar{W} + \bar{R}\|_\infty < \gamma$, where $\bar{W} := T_2^{\sim} T_1 \Psi^{-1} \in \mathcal{B}$. The condition (b) is a well-known extension problem. A solution exists if and only if the induced norm of the Hankel operator with symbol $\bar{W}$ is less than γ , which is part of a result known as Nehari’s theorem. In fact, (b) is equivalent to the existence of (J) -spectral factor $\Upsilon, \Upsilon^{-1} \in \mathcal{S}$ with $\Upsilon_{11}^{-1} \in \mathcal{S}$ such that

$$\Upsilon^{\sim} \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \Upsilon = \begin{bmatrix} I & \bar{W} \\ 0 & I \end{bmatrix}^{\sim} \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} I & \bar{W} \\ 0 & I \end{bmatrix}, \quad (12)$$

in which case $\|\bar{W} + \bar{R}\|_\infty \leq \gamma$ if and only if $\bar{R} = \bar{R}_1 \bar{R}_2^{-1}$ with $\begin{bmatrix} \bar{R}_1^T & \bar{R}_2^T \end{bmatrix} := \begin{bmatrix} \bar{S}^T & I \end{bmatrix} \Upsilon^{-T}$, $\bar{S} \in \mathcal{S}$ and $\|\bar{S}\|_\infty \leq \gamma$; see Ball and Ran (1987), Francis (1987), and Green et al. (1990) for details, including state-space realization-based constructions of the factors via Riccati equations. Noting that

$$\begin{aligned} &\begin{bmatrix} T_2 & T_1 \end{bmatrix}^{\sim} \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} T_2 & T_1 \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ 0 & \Psi \end{bmatrix}^{\sim} \begin{bmatrix} I & \bar{W} \\ 0 & I \end{bmatrix}^{\sim} \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} I & \bar{W} \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \Psi \end{bmatrix}, \end{aligned}$$

it follows using (12) that there exists $Q \in \mathcal{S}$ such that (10) holds if and only if there exists (J -)spectral factor Ω , $\Omega^{-1} \in \mathcal{S}$ with $\Omega_{11}^{-1} \in \mathcal{S}$ ($\Omega = \Upsilon \begin{bmatrix} I & 0 \\ 0 & \psi \end{bmatrix}$) that satisfies

$$\begin{bmatrix} T_2 & T_1 \\ 0 & I \end{bmatrix} \sim \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} T_2 & T_1 \\ 0 & I \end{bmatrix} \\ = \Omega \sim \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \Omega, \quad (13)$$

in which case $\|T_1 + T_2 Q\|_\infty \leq \gamma$ if and only if $Q = Q_1 Q_2^{-1}$, where $\begin{bmatrix} Q_1^T & Q_2^T \end{bmatrix} := \begin{bmatrix} S^T & I \end{bmatrix} \Omega^{-T}$, $S \in \mathcal{S}$ and $\|S\|_\infty \leq \gamma$; see Green et al. (1990). So-called J -spectral factorizations of the kind in (12) and (13) also appear in the chain-scattering/conjugation approach of Kimura (1989, 1997) and the factorization approach of Ball et al. (1991), for example.

Summary

The preceding sections highlight the role of coprime and spectral factorizations in formulating and solving model-matching problems that arise from standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems. The transformation of standard control problems to model-matching problems hinges on an affine parametrization of internally stabilized performance maps. Beyond the problems considered here, this parametrization can be exploited to devise numerical algorithms for various other control problems in terms of convex mathematical programs.

Cross-References

- [H₂ Optimal Control](#)
- [H-Infinity Control](#)
- [Polynomial/Algebraic Design Methods](#)
- [Spectral Factorization](#)

Bibliography

- Ball JA, Ran ACM (1987) Optimal Hankel norm model reductions and Wiener-Hopf factorization I: the canonical case. *SIAM J Control Optim* 25(2): 362–382
- Ball JA, Helton JW, Verma M (1991) A factorization principle for stabilization of linear control systems. *Int J Robust Nonlinear Control* 1(4): 229–294
- Boyd SP, Barratt CH (1991) Linear controller design: limits of performance. Prentice Hall, Englewood Cliffs
- Curtain RF, Zwart HJ (1995) An introduction to infinite-dimensional linear systems theory. Volume 21 of texts in applied mathematics. Springer, New York
- Dahleh MA, Diaz-Bobillo IJ (1995) Control of uncertain systems: a linear programming approach. Prentice Hall, Upper Saddle River
- DeSantis RM, Saeks R, Tung LJ (1978) Basic optimal estimation and control problems in Hilbert space. *Math Syst Theory* 12(1):175–203
- Desoer C, Liu R-W, Murray J, Saeks R (1980) Feedback system design: the fractional representation approach to analysis and synthesis. *IEEE Trans Autom Control* 25(3):399–412
- Feintuch A (1998) Robust control theory in Hilbert space. Applied mathematical sciences. Springer, New York
- Francis BA (1987) A course in H_∞ control theory. Lecture notes in control and information sciences. Springer, Berlin/New York
- Francis BA, Doyle JC (1987) Linear control theory with an H_∞ optimality criterion. *SIAM J Control Optim* 25(4):815–844
- Glover K, Limebeer DJN, Doyle JC, Kasenally EM, Safonov MG (1991) A characterization of all solutions to the four block general distance problem. *SIAM J Control Optim* 29(2):283–324
- Green M, Glover K, Limebeer DJN, Doyle JC (1990) A J -spectral factorization approach to $\mathcal{H}_\infty$ control. *SIAM J Control Optim* 28(6):1350–1371
- Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs
- Kimura H (1989) Conjugation, interpolation and model-matching in H_∞ . *Int J Control* 49(1):269–307
- Kimura H (1997) Chain-scattering approach to H^∞ control. Systems and control. Birkhäuser, Boston
- Kucera V (1975) Stability of discrete linear control systems. In: 6th IFAC world congress, Boston. Paper 44.1
- Qi X, Salapaka MV, Voulgaris PG, Khammash M (2004) Structured optimal and robust control with multiple criteria: a convex solution. *IEEE Trans Autom Control* 49(10):1623–1640
- Quadrat A (2006) On a generalization of the Youla–Kučera parametrization. Part II: the lattice approach to MIMO systems. *Math Control Signals Syst* 18(3): 199–235

- Vidyasagar M (1985) Control system synthesis: a factorization approach. Signal processing, optimization and control. MIT, Cambridge
- Youla D, Jabr H, Bongiorno J (1976) Modern Wiener-Hopf design of optimal controllers – Part II: the multi-variable case. IEEE Trans Autom Control 21(3):319–338
- Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River

Optimal Control with State Space Constraints

Heinz Schättler
Washington University, St. Louis, MO, USA

Abstract

Necessary and sufficient conditions for optimality in optimal control problems with state space constraints are reviewed with emphasis on geometric aspects.

Keywords

Admissible control · Bolza form · Mayer problem

Problem Formulation and Terminology

Many practical problems in engineering or of scientific interest can be formulated in the framework of optimal control problems with state space constraints. Examples range from the space shuttle reentry problem in aeronautics (Bonnard et al. 2003) to the problem of minimizing the base transit time in bipolar transistors in electronics (Rinaldi and Schättler 2003).

An optimal control problem with state space constraints in Bolza form takes the following form: minimize a functional

$$J(u) = \int_{t_0}^T L(t, x(t), u(t))dt + \Phi(T, x(T))$$

over all Lebesgue measurable functions $u : [t_0, T] \rightarrow U$ that take values in a control set $U \subset \mathbb{R}^m$, subject to the dynamics

$$\dot{x}(t) = F(t, x(t), u(t)), \quad x(t_0) = x_0,$$

terminal constraints

$$\Psi(T, x(T)) = 0,$$

and state space constraints

$$h_\alpha(t, x(t)) \leq 0 \quad \text{for } \alpha = 1, \dots, r.$$

The focus of this contribution is on state space constraints, and, for simplicity, in this formulation, we have omitted mixed control state space constraints of the form $g_\beta(t, x, u) \leq 0$. States x lie in $\mathbb{R}^n$ and controls in $\mathbb{R}^m$; typically, the control set $U \subset \mathbb{R}^m$ is compact and convex, often a polyhedron. The time-varying vector field $F : \mathbb{R} \times \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$ is continuously differentiable in (t, x) , and the terminal constraint $N = \{(t, x) : \Psi(t, x) = 0\}$ is defined by continuously differentiable mappings $\psi_i : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^k$ with the property that the gradients $\nabla \psi_i = (\frac{\partial \psi_i}{\partial t}, \frac{\partial \psi_i}{\partial x})$ (which we write as row vectors) are linearly independent on N . The terminal time T can be free or fixed; a fixed terminal time simply would be prescribed by one of the functions ψ_i . The state space constraints

$$M_\alpha = \{(t, x) : h_\alpha(t, x) = 0\}, \quad \alpha = 1, \dots, r,$$

are defined by continuously differentiable time-varying vector fields $h_\alpha : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$, $(t, x) \mapsto h_\alpha(t, x)$, and we assume that the gradients ∇h_α do not vanish on M_α . In particular, each set M_α thus is an embedded submanifold of codimension 1 of $\mathbb{R}^{n+1}$. We denote by $h = (h_1, \dots, h_r)^T$ the

time-varying vector field defining the state space constraints.

Terminology: *Admissible controls* are locally bounded Lebesgue measurable functions that take values in the control set, $u : [t_0, T] \rightarrow U$. Given any admissible control, the initial value problem $\dot{x}(t) = F(t, x(t), u(t))$, $x(t_0) = x_0$, has a unique solution defined on some maximal open interval of definition I . This solution is called the *trajectory* corresponding to the control u and the pair (x, u) is a *controlled trajectory*. An arc Γ of the graph of a trajectory defined over an open interval I for which none of the state space constraints is active is called an *interior arc*, and Γ is a *boundary arc* if at least one constraint is active on all of I . We call Γ an M_α -boundary arc over I if only the constraint $h_\alpha \leq 0$ is active on I . The times τ when interior arcs and boundary arcs meet are called *junction times* and the corresponding pairs $(\tau, x(\tau))$ *junction points*.

Despite the abundance and importance of practical problems that can be described as optimal control problems with state space constraints, for such problems the theory still lacks the coherence that the theory for problems without state space constraints has reached and there still exist significant gaps between the theories of necessary and sufficient conditions for optimality for optimal control problems with state space constraints. The theory of existence of optimal solutions differs little between optimal control problems with and without state space constraints, is well established, and will not be addressed here (e.g., see Cesari 1983 or the Filippov-Cesari theorem in Hartl et al. 1995).

Necessary Conditions for Optimality

First-order necessary conditions for optimality are given by the *Pontryagin maximum principle* (Pontryagin et al. 1962). The zero set of even a smooth (C^∞) function can be an arbitrary closed subset of the state space. As a result, in necessary conditions for optimality, the multipliers associated with the state space constraints a priori are only known to be nonnegative Radon measures

(Ioffe and Tikhomirov 1979; Vinter 2000). Let $u_* : [t_0, T] \rightarrow U$ be an optimal control with corresponding trajectory x_* and, for simplicity of presentation, also assume that no state constraints are active at the terminal time so that the standard transversality conditions apply. Then it follows that there exist a constant $\lambda_0 \geq 0$, an absolutely continuous function η , which we write as row-vector, $\eta : [t_0, T] \rightarrow (\mathbb{R}^n)^*$, and nonnegative Radon measures $\mu_\alpha \in C^*([t_0, T]; \mathbb{R})$, $\alpha = 1, \dots, r$, with support in the sets $R_\alpha = \{t \in [t_0, T] : h_\alpha(t, x_*(t)) = 0\}$, which do not all vanish simultaneously, i.e.,

$$\lambda_0 + \|\eta\|_\infty + \sum_{\alpha=1}^r \mu_\alpha([t_0, T]) > 0,$$

such that with

$$\lambda(t) = \eta(t) - \sum_{\alpha=1}^r \int_{[t_0, t]} \frac{\partial h_\alpha}{\partial x}(s, x_*(s)) d\mu_\alpha(s),$$

and

$$H = H(t, \lambda_0, \lambda, x, u) = \lambda_0 L(t, x, u) + \lambda F(t, x, u)$$

the following conditions hold:

(a) The adjoint equation holds in the form

$$\begin{aligned} \dot{\eta}(t) &= -\frac{\partial H}{\partial x}(t, \lambda_0, \lambda(t), x_*(t), u_*(t)) \\ &= -\lambda_0 \frac{\partial L}{\partial x}(t, x_*(t), u_*(t)) \\ &\quad - \lambda(t) \frac{\partial F}{\partial x}(t, x_*(t), u_*(t)), \end{aligned}$$

and there exists a row-vector $\mu \in (\mathbb{R}^k)^*$ such that

$$\lambda(T) = \lambda_0 \frac{\partial \Phi}{\partial x}(T, x_*(T)) + \mu \frac{\partial \Psi}{\partial x}(T, x_*(T))$$

and

$$\begin{aligned} 0 &= H(T, \lambda_0, \lambda(T), x_*(T), u_*(T)) \\ &\quad + \lambda_0 \frac{\partial \Phi}{\partial t}(T, x_*(T)) + \mu \frac{\partial \Psi}{\partial t}(T, x_*(T)). \end{aligned}$$

- (b) The optimal control minimizes the Hamiltonian over the control set U along $(\lambda(t), x_*(t))$:

$$\begin{aligned} & H(t, \lambda_0, \lambda(t), x_*(t), u_*(t)) \\ &= \min_{v \in U} H(t, \lambda_0, \lambda(t), x_*(t), v). \end{aligned}$$

Furthermore,

$$\begin{aligned} H(t, \lambda_0, \lambda(t), x_*(t), u_*(t)) \\ &= H(T, \lambda_0, \lambda(t), x_*(t), u_*(t)) \\ &\quad - \int_{[t, T]} \frac{\partial H}{\partial t}(s, \lambda_0, \lambda(s), x_*(s), u_*(s)) ds \\ &\quad + \sum_{\alpha=1}^r \int_{[t, T]} \frac{\partial h_\alpha}{\partial t}(s, x_*(s)) d\mu_\alpha(s) \end{aligned}$$

Controlled trajectories (x, u) for which there exist multipliers such that these conditions are satisfied are called *extremals*. In general, it cannot be excluded that λ_0 vanishes and extremals with $\lambda_0 = 0$ are called *abnormal*, while those with $\lambda_0 > 0$ are called *normal*. In this case, the multiplier can be normalized, $\lambda_0 = 1$.

Special Case: A Mayer Problem for Single-Input Control Linear Systems

Under the general assumptions formulated above, the sets $R_\alpha \subset [t_0, T]$ when a particular constraint is active can be arbitrarily complicated. But in many practical applications, state constraints have strong geometric properties – often they are embedded submanifolds – and it is possible to strengthen these necessary conditions for optimality in the sense of specifying the measures further. We formulate the conditions for a particular case of common interest.

We consider an optimal control problem in *Mayer form* (i.e., $L \equiv 0$) for a single-input control linear system with dynamics

$$\dot{x} = F(t, x, u) = f(t, x) + ug(t, x)$$

and the control set U a compact interval, $U = [a, b]$. Adjoining time as extra state variable, $i \equiv 1$, and defining

$$F_0(t, x) = \begin{pmatrix} 1 \\ f(t, x) \end{pmatrix} \text{ and } G(t, x) = \begin{pmatrix} 0 \\ g(t, x) \end{pmatrix},$$

for a continuously differentiable function $k : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$, the expressions

$$\begin{aligned} & \mathcal{L}_{F_0} k : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n, \\ & (t, x) \mapsto (\mathcal{L}_{F_0} k)(t, x) \\ &= \frac{\partial k}{\partial t}(t, x) + \frac{\partial k}{\partial x}(t, x) f(t, x) \end{aligned}$$

and

$$\begin{aligned} & \mathcal{L}_G k : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n, \\ & (t, x) \mapsto (\mathcal{L}_G k)(t, x) = \frac{\partial k}{\partial x}(t, x) g(t, x) \end{aligned}$$

represent the Lie (or directional) derivatives of the function k along the vector fields F_0 and G , respectively. In terms of this notation, the derivative of the function h_α (defining the manifold M_α) along trajectories of the system is given by

$$\begin{aligned} \dot{h}_\alpha(t, x(t)) &= \frac{d}{dt} h_\alpha(t, x(t)) \\ &= \mathcal{L}_{F_0} h_\alpha(t, x(t)) + u(t) \mathcal{L}_G h_\alpha(t, x(t)). \end{aligned}$$

If the function $\mathcal{L}_G h_\alpha$ does not vanish at a point $(\tilde{t}, \tilde{x}) \in M_\alpha$, then there exists a neighborhood V of $(\tilde{t}, \tilde{x})$ such that there exists a unique control $u_\alpha = u_\alpha(t, x)$ which solves the equation $\dot{h}_\alpha(t, x) = 0$ on V and u_α is given in feedback form as

$$u_\alpha(t, x) = -\frac{\mathcal{L}_{F_0} h_\alpha(t, x)}{\mathcal{L}_G h_\alpha(t, x)}.$$

The manifold M_α is said to be *control invariant of relative degree 1* if the Lie derivative of h_α with respect to G , $\mathcal{L}_G h_\alpha$, does not vanish anywhere on M_α and if the function $u_\alpha(t, x)$ is admissible, i.e., takes values in the control set $[a, b]$.

Thus, for a control-invariant submanifold of relative degree 1, the control that keeps the manifold invariant is unique, and the corresponding dynamics induce a unique flow on the constraint. This assumption corresponds to the least degenerate, i.e., in some sense most generic or common,

scenario and is satisfied for many practical problems.

Suppose the reference extremal is normal and let Γ_α be an M_α -boundary arc defined over an open interval I with corresponding boundary control u_α that takes values in the interior of the control set along Γ_α . Then the Radon measure μ_α is absolutely continuous with respect to Lebesgue measure on I with continuous and nonnegative Radon-Nikodym derivative $v_\alpha(t)$ given by

$$v_\alpha(t) = \frac{\lambda(t) \left(\frac{\partial g}{\partial t}(t, x_*(t)) + [f, g](t, x_*(t)) \right)}{\mathcal{L}_G h_\alpha(t, x_*(t))}$$

where $[f, g]$ denotes the Lie bracket of the time-varying vector fields f and g in the variable x ,

$$[f, g](t, x) = \frac{\partial g}{\partial x}(t, x)f(t, x) - \frac{\partial f}{\partial x}(t, x)g(t, x).$$

In particular, in this case, the adjoint equation can be expressed in the more common form

$$\dot{\lambda}(t) = -\lambda(t) \frac{\partial F}{\partial x}(t, x_*, u_*) - v_\alpha(t) \frac{\partial h_\alpha}{\partial x}(t, x_*),$$

with all partial derivatives evaluated along the reference trajectory. Furthermore, the multiplier λ remains continuous at entry or exit if the controlled trajectory (x_*, u_*) meets the constraint M_α transversally (e.g., see Schättler 2006). This follows from the following characterization of transversal connections between interior and boundary arcs due to Maurer (1977): if τ is an entry or exit junction time between an interior arc and an M_α -boundary arc for which the reference control u_* has a limit at τ along the interior arc, then the interior arc is transversal to M_α at entry or exit if and only if the control u_* is discontinuous at τ .

Informal Formulation of Necessary Conditions

In order to ensure the practicality of necessary conditions for optimality, it is essential that

besides atomistic structures at junctions that lead to computable jumps in the multipliers, the Radon measures μ_α have no singular parts with respect to Lebesgue measure. If it is *assumed* a priori that optimal controlled trajectories are finite concatenations of interior and boundary arcs, and if the constraint sets have a reasonably regular structure (embedded submanifolds and transversal intersections thereof) and satisfy a rather technical *constraint qualification* (see Hartl et al. 1995) that guarantees that the restrictions of the system to active constraints have solutions, then it is possible to specify the above necessary conditions further and formulate more user friendly versions for the determination of the multipliers. Such formulations have become the standard for numerical computations, but they still have not always been established rigorously and somewhat carry the stigma of a heuristic nature. Nevertheless, it is often this more concrete set of conditions that allow to solve problems numerically and analytically. If then, in conjunction with sufficient conditions for optimality, it is possible to verify the optimality of the computed extremal solutions, this generates a satisfactory theoretical procedure. Such conditions, following Hartl et al. (1995), generally are referred to as the “informal theorem”.

Suppose (x_*, u_*) is a normal extremal controlled trajectory defined over the interval $[t_0, T]$ with the property that the graph of x_* is a finite concatenation of interior and boundary arcs with junction times τ_i , $i = 1, \dots, k$, $t_0 = \tau_0 < \tau_1 < \dots < \tau_k < \tau_{k+1} = T$. Under an appropriate constraint qualification, there exist a multiplier λ , $\lambda : [t_0, T] \rightarrow (\mathbb{R}^n)^*$, which is absolutely continuous on each subinterval $[\tau_i, \tau_{i+1}]$; multipliers $v_\alpha, v_\alpha : [t_0, T] \rightarrow (\mathbb{R}^n)^*$, which are continuous on each interval $[\tau_i, \tau_{i+1}]$; a vector $\mu \in (\mathbb{R}^k)^*$; and vectors $\eta(\tau_i) \in (\mathbb{R}^r)^*$, $i = 1, \dots, k$, with nonnegative entries such that:

- (a) (adjoint equation) On each interval (τ_i, τ_{i+1}) , $i = 0, \dots, r$, λ satisfies the adjoint equation in the form

$$\begin{aligned}\dot{\lambda}(t) = & -\frac{\partial L}{\partial x}(t, x_*(t), u_*(t)) \\ & -\lambda(t) \frac{\partial F}{\partial x}(t, x_*(t), u_*(t)) \\ & -\sum_{\alpha=1}^r v_{\alpha}(t) \frac{\partial h_{\alpha}}{\partial x}(t, x_*),\end{aligned}$$

with $v_{\alpha}(t) = 0$ if the constraint M_{α} is not active at time t . Assuming that no state space constraint is active at the terminal time, the value of the multiplier λ at the terminal time is given by the transversality condition

$$\lambda(T) = \frac{\partial \Phi}{\partial x}(T, x_*(T)) + \mu \frac{\partial \Psi}{\partial x}(T, x_*(T)).$$

At any junction time τ_i between an interior arc and a boundary arc, the multiplier λ may be discontinuous satisfying a jump condition of the form

$$\lambda(\tau_i-) = \lambda(\tau_i+) + \eta(\tau_i) \frac{\partial h}{\partial x}(\tau_i, x_*(\tau_i))$$

and the complementary slackness condition

$$\eta(\tau_i) \frac{\partial h}{\partial x}(\tau_i, x_*(\tau_i)) = 0$$

holds.

- (b) The optimal control minimizes the Hamiltonian over the control set U along $(\lambda(t), x_*(t))$:

$$\begin{aligned}H(t, \lambda(t), x_*(t), u_*(t)) \\ = \min_{v \in U} H(t, \lambda(t), x_*(t), v)\end{aligned}$$

and at the junction times τ_i we have that

$$\begin{aligned}H(\tau_i, \lambda(\tau_i-), x_*(\tau_i), u_*(\tau_i-)) \\ = H(\tau_i, \lambda(\tau_i+), x_*(\tau_i), u_*(\tau_i+)) \\ - \eta(\tau_i) \frac{\partial h}{\partial t}(\tau_i, x_*(\tau_i)).\end{aligned}$$

Sufficient Conditions for Optimality

The literature on sufficient conditions for optimality for optimal control problems with state space constraints is limited. The value function for an optimal control problem at a point (t, x) in the extended state space, $V = V(t, x)$, is defined as the infimum over all admissible controls u for which the corresponding trajectory starts at the point x at time t and satisfies all the constraints of the problem,

$$V(t, x) = \inf_{u \in \mathcal{U}} J(u).$$

Any sufficiency theory for optimal control problems, one way or another, deals with the solution of the corresponding Hamilton-Jacobi-Bellman (HJB) equation:

$$\begin{aligned}\frac{\partial V}{\partial t}(t, x) + \min_{u \in U} \left\{ \frac{\partial V}{\partial x}(t, x) F(t, x, u) \right. \\ \left. + L(t, x, u) \right\} \equiv 0,\end{aligned}$$

$$V(T, x) = \Phi(T, x) \text{ whenever } \Psi(T, x) = 0.$$

Value functions for optimal control problems rarely are differentiable everywhere, but generally have singularities along lower-dimensional submanifolds. Nevertheless, under some technical assumptions and with proper interpretations of the derivatives, this equation describes the evolution of the value function of an optimal control problem and, if an appropriate solution can be constructed, indeed solves the optimal control problem.

There exists a broad theory of viscosity solutions to the HJB equation (e.g., Fleming and Soner 2005; Bardi and Capuzzo-Dolcetta 2008) that is also applicable to problems with state space constraints (Soner 1986) and, under varying technical assumptions, characterizes the value function V as the unique viscosity solution to the HJB equation. This has led to the development of algorithms that can be used to compute numerical solutions.

A more classical and more geometric approach to solving the HJB equation is based on the method of characteristics and goes back to the work of Boltyansky on a regular synthesis for optimal control problems without state space constraints (Boltyanskii 1966). This work follows classical ideas of fields of extremals from the calculus of variations and imposes technical conditions that allow to handle the singularities that arise in the value functions (e.g., see, Schättler and Ledzewicz 2012). Stalford's results in Stalford (1971) follow this approach for problems with state space constraints, but a broadly applicable theory of regular synthesis, as it was developed by Piccoli and Sussmann in (2000) for problems without state space constraints, does not yet exist for problems with state space constraints. Results that embed a controlled reference extremal into a local field of extremals have been given by

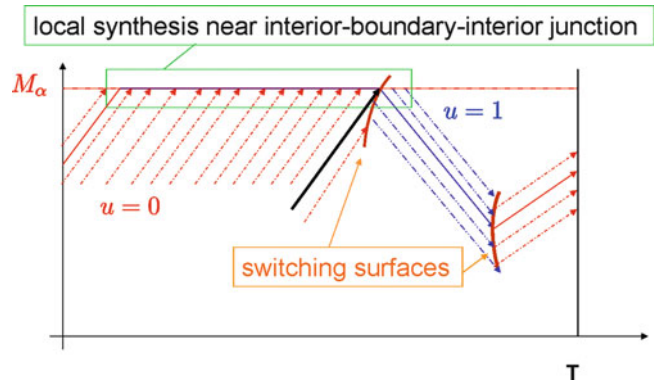
Bonnard et al. (2003) or Schättler (2006), and these constructions show the applicability of the concepts of a regular synthesis to problems with state space constraints as well.

Examples of Local Embeddings of Boundary Arcs

We illustrate the typical, i.e., in some sense most common, generic structures of local embeddings of boundary arcs in Figs. 1 and 2. The state constraint M_α is a control-invariant submanifold of relative degree 1 and represented by a horizontal line as it arises when limits on the size of a particular state are imposed. Figure 1 shows the typical entry-boundary-exit concatenations of an interior arc followed by a boundary arc and another interior arc. The local embedding of the boundary arc differs substantially from classical local imbeddings for unconstrained problems in the sense that this field necessarily contains

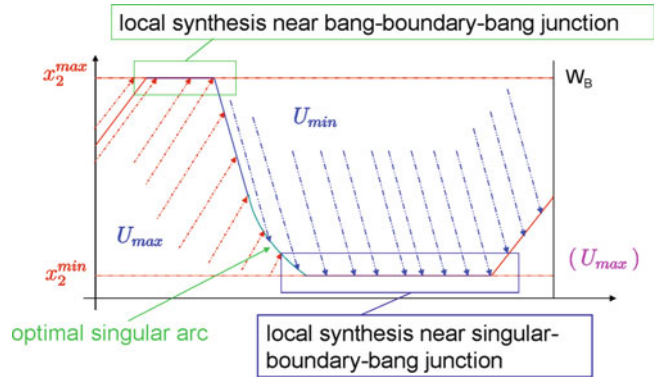
Optimal Control with State Space Constraints,

Fig. 1 A typical local synthesis around a boundary arc when no terminal constraints are present



Optimal Control with State Space Constraints,

Fig. 2 A typical local synthesis around a boundary arc when terminal constraints are present



small pieces of trajectories which, when propagated backward, are not close to the reference trajectory. This, however, does not affect the memoryless properties required for a synthesis forward in time, and strong local optimality of the reference trajectory can be proven combining synthesis type arguments with homotopy type approximations of the synthesis (Schättler 2006). The one trajectory marked as black line in Fig. 1 corresponds to an optimal trajectory that meets the constraint only at the junction point and immediately bounces back into the interior. Such a trajectory arises as the limit when the concatenation structure of optimal controlled trajectories changes from interior-boundary-interior arcs to trajectories that do not meet the constraint. These structures are one of the extra sources for singularities in the value function that come up in optimal control problems with state space constraints. Switching surfaces for the interior arcs, as one is also shown in this figure, do not cause such a loss of differentiability if they are crossed transversally by the extremal trajectories of the field.

Figure 2 depicts the structure of an optimal synthesis for a problem from electronics, the problem of minimizing the base transit time of bipolar homogeneous transistors. The electrical field that determines the transit time is controlled by tailoring a distribution of dopants in the base region, and this dopant profile becomes an important design parameter determining the speed of the device. But due to physical and engineering limitations, the variables describing the dopants need to be limited, and thus this becomes an optimal control problem with state space constraints represented by hard limits on the variables. The constraints here are control invariant of relative degree 1. Optimal solutions, in the presence of initial and terminal constraints, have both portions along the upper and lower control limits of the constrained variable and typically proceed from the upper to the lower values along an optimal singular control (which takes values in the interior of the control set) in the interior of the admissible domain, possibly with saturation if the control limits are reached.

Cross-References

- ▶ [Numerical Methods for Nonlinear Optimal Control Problems](#)
- ▶ [Optimal Control and Pontryagin's Maximum Principle](#)

Bibliography

- Bardi M, Capuzzo-Dolcetta I (2008) Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations. Springer, New York
- Boltyanskii VG (1966) Sufficient conditions for optimality and the justification of the dynamic programming principle. *SIAM J Control Optim* 4:326–361
- Bonnard B, Faubourg L, Launay G, Trélat E (2003) Optimal control with state space constraints and the space shuttle re-entry problem. *J Dyn Control Syst* 9:155–199
- Cesari L (1983) Optimization – theory and applications. Springer, New York
- Fleming WH, Soner HM (2005) Controlled Markov processes and viscosity solutions. Springer, New York
- Frankowska H (2006) Regularity of minimizers and of adjoint states in optimal control under state constraints. *Convex Anal* 13:299–328
- Hartl RF, Sethi SP, Vickson RG (1995) A survey of the maximum principles for optimal control problems with state constraints. *SIAM Rev* 37:181–218
- Ioffe AD, Tikhomirov VM (1979) Theory of extremal problems. North-Holland, Amsterdam/New York
- Maurer H (1977) On optimal control problems with bounded state variables and control appearing linearly. *SIAM J Control Optim* 15:345–362
- Piccoli B, Sussmann H (2000) Regular synthesis and sufficient conditions for optimality. *SIAM J Control Optim* 39:359–410
- Pontryagin LS, Boltyanskii VG, Gamkrelidze RV, Mishchenko EF (1962) Mathematical theory of optimal processes. Wiley-Interscience, New York
- Rinaldi P, Schättler H (2003) Minimization of the base transit time in semiconductor devices using optimal control. In: Feng W, Hu S, Lu X (eds) Dynamical systems and differential equations. Proceedings of the 4th international conference on dynamical systems and differential equations. Wilmington, May 2002, pp 742–751
- Schättler H (2006) A local field of extremals for optimal control problems with state space constraints of relative degree 1. *J Dyn Control Syst* 12:563–599
- Schättler H, Ledzewicz U (2012) Geometric optimal control. Springer, New York
- Soner HM (1986) Optimal control with state-space constraints I. *SIAM J Control Optim* 24:552–561
- Stalford H (1971) Sufficient conditions for optimal control with state and control constraints. *J Optim Theory Appl* 7:118–135
- Vinter RB (2000) Optimal control. Birkhäuser, Boston

Optimal Deployment and Spatial Coverage

Sonia Martínez

Department of Mechanical and Aerospace
Engineering, University of California, La Jolla,
San Diego, CA, USA

Abstract

Optimal deployment refers to the problem of how to allocate a finite number of resources over a spatial domain to maximize a performance metric that encodes certain quality of service. Depending on the deployment environment, the type of resource, and the metric used, the solutions to this problem can greatly vary.

Keywords

Coverage control algorithms · Facility location problems

Introduction

The problem of deciding what are optimal geographic locations to place a set of facilities has a long history and is the main subject in operations research and management science; see Drezner (1995). A facility can be broadly understood as a service such as a school; a hospital; an airport; an emergency service, such as a fire station; or, more generally, routes of a vehicle, from buses to aircraft, an autonomous vehicle, or a mobile sensor.

The specific formulation of facility location problems depends very much on the particular underlying application. A distinguishing feature is that all involve strategic planning, accounting for the long-term impact on the facility operating cost and their fast response to the demand. Thus, these problems lead to constrained optimization formulations which are typically very hard to solve optimally. The computational complexity of such problems, which, even in their most basic

formulations, typically lead to NP-hard problems, has made their solution largely intractable until the advent of high-speed computing.

Locational optimization techniques have also been employed to solve optimal estimation problems by static sensor networks, mesh and grid optimization design, clustering analysis, data compression, and statistical pattern recognition; see Du et al. (1999). However, these solutions typically require centralized computations and availability of information at all times.

When the facilities are multiple vehicles or mobile sensors, the underlying dynamics may require additional changes and further analysis that guarantee the overall system stability. In what follows, we review a particular coverage control problem formulation in terms of the so-called expected-value multicenter functions that makes the analysis tractable leading to robust, distributed algorithm implementations employing computational geometric objects such as Voronoi partitions.

Basic Ingredients from Computational Geometry

In order to formulate a basic optimal deployment problem and algorithm, we require of several notions from computational geometry; see Bullo et al. (2009) for more information.

Let S be a measurable set of $\mathbb{R}^m$, for $m \in \mathbb{N}$, consider a distance function d on $\mathbb{R}^m$, and let $P = \{p_1, \dots, p_n\}$ be n distinct points of S , corresponding to *locations* of certain facilities. The *Voronoi partition* of S generated by P and associated with d is given by $\mathcal{V}(P) = \{V_1(P), \dots, V_n(P)\}$, where

$$V_i(P) = \{q \in S \mid d(p_i, q) \leq d(p_j, q), \\ j \in P \setminus \{i\}\}, \quad i \in \{1, \dots, n\}.$$

Given $r \in \mathbb{R}_{>0}$, denote by $\overline{B}(p_i, r)$ the closed ball of center p_i and radius r . The *r-limited Voronoi partition* of S generated by P and associated with d is the Voronoi partition of

the set $S \cap \bigcup_{i=1}^n \overline{B}(p_i, r)$, denoted as $\mathcal{V}_r(P) = \{V_{1,r}(P), \dots, V_{n,r}(P)\}$.

Let $\phi : S \rightarrow \mathbb{R}_{\geq 0}$ be a measurable *density* function on S . The *area* and the *centroid* (or *center of mass*) of $W \subseteq S$ with respect to ϕ are the values

$$A_\phi(W) = \int_W \phi(q) dq,$$

$$CM_\phi(W) = \frac{1}{A_\phi(W)} \int_W q \phi(q) dq.$$

We say that the set of distinct points P in S is a *centroidal Voronoi configuration* (resp., a *r-limited centroidal Voronoi configuration*) if each p_i is at the centroid of its own Voronoi cell. That is, $p_i = CM_\phi(V_i(P))$, $i \in \{1, \dots, n\}$ (resp., $p_i = CM_\phi(V_{i,r}(P))$, and $i \in \{1, \dots, n\}$). Voronoi partitions and centroidal Voronoi configurations help assess the distribution of locations in a spatial domain as we establish below.

A Voronoi partition induces a natural *proximity graph*, called the *Delaunay graph*, over the set of points P . We recall that a *graph* G is a pair $G = (V, E)$ where V is a set of n vertices and E is a set of ordered pair of *vertices*, $E \subset V \times V$, called *edge set*. A *proximity graph* is a graph function defined on the set S , which assigns a set of distinct points $P \subset S$ to a graph $G(P) = (P, E(P))$, where $E(P)$ is a function of the relative locations of the point set. Example graphs include the following:

1. The *r-disk graph*, $\mathcal{G}_{\text{disk},r}$, for $r \in \mathbb{R}_{>0}$. Here, $(p_i, p_j) \in E_{\text{disk},r}(P)$ if $d(p_i, p_j) \leq r$.
2. The *Delaunay graph*, $\mathcal{G}_D$. We have $(p_i, p_j) \in E_D(P)$ if $V_i(P) \cap V_j(P) \neq \emptyset$.
3. The *r-limited Delaunay graph*, $\mathcal{G}_{\text{LD},r}$, for $r \in \mathbb{R}_{>0}$. Here, $(p_i, p_j) \in E_{\text{LD},r}(P)$ if $V_{i,r}(P) \cap V_{j,r}(P) \neq \emptyset$.

Expected-Value Multicenter Functions

Facility location problems consist of spatially allocating a number of sites to provide certain

quality of service. Problems of this class are formulated in terms of multicenter functions and, in particular, expected-value multicenter functions.

To define these, consider $\phi : S \rightarrow \mathbb{R}_{\geq 0}$ a density function over a bounded measurable set $S \subset \mathbb{R}^m$. One can regard ϕ as a function measuring the probability that some event takes place over the environment. The larger the value of $\phi(q)$, the more important the location q will have. We refer to a nonincreasing and piecewise continuously differentiable function $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, possibly with finite jump discontinuities, as a *performance function*.

Performance functions describe the utility of placing a node at a certain distance from a location in the environment. The smaller the distance, the larger the value of f , that is, the better the performance. For instance, in sensing problems, performance functions can encode the signal-to-noise ratio between a source with an unknown location and a sensor attempting to locate it. Without loss of generality, it can be assumed that $f(0) = 0$.

An *expected-value multicenter function* models the expected value of the coverage over any point in S provided by a set of points $p_1, \dots, p_n$. Formally,

$$\mathcal{H}(p_1, \dots, p_n) = \int_S \max_{i \in \{1, \dots, n\}} f(\|q - p_i\|_2) \phi(q) dq, \quad (1)$$

where $\|\cdot\|_2$ denotes the 2-norm of $\mathbb{R}^m$. This definition can be understood as follows: consider the best coverage of $q \in S$ among those provided by each of the nodes $p_1, \dots, p_n$, which corresponds to the value $\max_{i \in \{1, \dots, n\}} f(\|q - p_i\|_2)$. Then, modulate the performance by the importance $\phi(q)$ of the location q . Finally, the infinitesimal sum of this quantity over the environment S gives rise to $\mathcal{H}(p_1, \dots, p_n)$ as a measure of the overall coverage provided by $p_1, \dots, p_n$.

From here, we can formulate the following geometric optimization problem, known as the *continuous p-median problem*, see Drezner (1995):

$$\max_{\{p_1, \dots, p_n\} \subset S} \mathcal{H}(p_1, \dots, p_n). \quad (2)$$

The expected-value multicenter function can be alternatively described in terms of the Voronoi partition of S generated by $P = \{p_1, \dots, p_n\}$. Let us define the set

$$\mathcal{C} = \{(p_1, \dots, p_n) \in (\mathbb{R}^m)^n \mid p_i = p_j \text{ for some } i \neq j\},$$

consisting of tuples of n points, where some of them are repeated. Then, for $(p_1, \dots, p_n) \in S^n \setminus \mathcal{C}$, one has

$$\mathcal{H}(p_1, \dots, p_n) = \sum_{i=1}^n \int_{V_i(P)} f(\|q - p_i\|_2) \phi(q) dq. \quad (3)$$

This expression of $\mathcal{H}$ is appealing because it clearly shows the result of the overall coverage of the environment as the aggregate contribution of all individual nodes. If $(p_1, \dots, p_n) \in \mathcal{C}$, then a similar decomposition of $\mathcal{H}$ can be written in terms of the distinct points $P = \{p_1, \dots, p_n\}$.

Inspired by (3), a more general version of the expected-value multicenter function is given next. Given $(p_1, \dots, p_n) \in S^n$ and a partition $\{W_1, \dots, W_n\}$ of S , let

$$\mathcal{H}(p_1, \dots, p_n, W_1, \dots, W_n) = \sum_{i=1}^n \int_{W_i} f(\|q - p_i\|_2) \phi(q) dq. \quad (4)$$

For all $(p_1, \dots, p_n) \in S^n \setminus \mathcal{C}$, we have that $\mathcal{H}(p_1, \dots, p_n) = \mathcal{H}(p_1, \dots, p_n, V_1(P), \dots, V_n(P))$. With respect to, e.g., sensor networks, this function evaluates the performance associated with an assignment of the sensors' locations at $(p_1, \dots, p_n)$ and a region assignment $(W_1, \dots, W_n)$.

Moreover, one can establish that the Voronoi partition (Du et al. 1999) $\mathcal{V}(P)$ is optimal for $\mathcal{H}$ among all partitions of S . That is, let $P = \{p_1, \dots, p_n\} \in S$. For any performance function f and for any partition $\{W_1, \dots, W_n\}$ of S ,

$$\mathcal{H}(p_1, \dots, p_n, V_1(P), \dots, V_n(P)) \geq$$

$$\mathcal{H}(p_1, \dots, p_n, W_1, \dots, W_n),$$

with a strict inequality if any set in $\{W_1, \dots, W_n\}$ differs from the corresponding set in $\{V_1(P), \dots, V_n(P)\}$ by a set of positive measure.

Next, we characterize the smoothness of the expected-value multicenter function (Cortés et al. 2005). Before stating the precise properties, let us introduce some useful notation. For a performance function f , let $\text{discont}(f)$ denote the (finite) set of points where f is discontinuous. For each $a \in \text{discont}(f)$, define the limiting values from the left and from the right, respectively, as

$$f_-(a) = \lim_{x \rightarrow a^-} f(x), \quad f_+(a) = \lim_{x \rightarrow a^+} f(x).$$

Recall that the line integral of a function $g : \mathbb{R}^2 \rightarrow \mathbb{R}$ over a curve C parameterized by a continuous and piecewise continuously differentiable map $\gamma : [0, 1] \rightarrow \mathbb{R}^2$ is defined as follows:

$$\int_C g = \int_C g(\gamma) d\gamma := \int_0^1 g(\gamma(t)) \|\dot{\gamma}(t)\|_2 dt,$$

and is independent of the selected parameterization.

Now, given a set $S \subset \mathbb{R}^m$ that is bounded and measurable, a density $\phi : S \rightarrow \mathbb{R}_{\geq 0}$, and a performance function $f : \mathbb{R} \rightarrow_{\geq 0} \mathbb{R}$, the expected-value multicenter function $\mathcal{H} : S^n \rightarrow \mathbb{R}$ is globally Lipschitz (Given $S \subset \mathbb{R}^h$, a function $f : S \rightarrow \mathbb{R}^k$ is globally Lipschitz if there exists $K \in \mathbb{R}_{>0}$ such that $\|f(x) - f(y)\|_2 \leq K\|x - y\|_2$ for all $x, y \in S$). on S^n ; and continuously differentiable on $S^n \setminus \mathcal{C}$, where for $i \in \{1, \dots, n\}$

$$\begin{aligned} \frac{\partial \mathcal{H}}{\partial p_i}(P) &= \int_{V_i(P)} \frac{\partial}{\partial p_i} f(\|q - p_i\|_2) \phi(q) dq \\ &+ \sum_{a \in \text{discont}(f)} (f_-(a) - f_+(a)) \\ &\int_{V_i(P) \cap \partial \bar{B}(p_i, a)} n_{\text{out}}(q) \phi(q) dq, \end{aligned} \quad (5)$$

where n_{out} is the outward normal vector to $\bar{B}(p_i, a)$.

Different performance functions lead to different expected-value multicenter functions. Let us examine some important cases.

Distortion Problem

Consider the performance function $f(x) = -x^2$. Then, on $S^n \setminus \mathcal{C}$, the expected-value multicenter function takes the form

$$\mathcal{H}_{\text{distor}}(p_1, \dots, p_n) = - \sum_{i=1}^n \int_{V_i(P)} \|q - p_i\|_2^2 \phi(q) dq.$$

In signal compression $-\mathcal{H}_{\text{distor}}$ is referred to as the *distortion function* and is relevant in many disciplines where including vector quantization, signal compression, and numerical integration; see Gray and Neuhoff (1998) and Du et al. (1999). Here, distortion refers to the average deformation (weighted by the density ϕ) caused by reproducing $q \in S$ with the location p_i in $P = \{p_1, \dots, p_n\}$ such that $q \in V_i(P)$. By means of the Parallel Axis Theorem (see Hibbeler 2006), it is possible to express $\mathcal{H}_{\text{distor}}$ as a sum

$$\begin{aligned} \mathcal{H}_{\text{distor}}(p_1, \dots, p_n, W_1, \dots, W_n) \\ = \sum_{i=1}^n -J_\phi(W_i, CM_\phi(W_i)) \\ - A_\phi(W_i) \|p_i - CM_\phi(W_i)\|_2^2, \quad (6) \end{aligned}$$

where $J_\phi(W, p) = \int_W \|q - p\|^2 \phi(q) dq$ is the so-called moment of inertia of the region W about p with respect to ϕ . In this way, the terms $J_\phi(W_i, CM_\phi(W_i))$ only depend on the partition of S , whereas the second terms multiplied by $A_\phi(W_i)$ include the particular location of the points. As a consequence of this observation, the optimality of the centroid locations for $\mathcal{H}_{\text{distor}}$ follows Bullo et al. (2009). More precisely, let $\{W_1, \dots, W_n\}$ be a partition of S . Then, for any set points $P = \{p_1, \dots, p_n\}$ in S ,

$$\begin{aligned} \mathcal{H}_{\text{distor}}(CM_\phi(W_1), \dots, CM_\phi(W_n), W_1, \dots, W_n) \\ \geq \mathcal{H}_{\text{distor}}(p_1, \dots, p_n, W_1, \dots, W_n), \end{aligned}$$

and the inequality is strict if there exists $i \in \{1, \dots, n\}$ for which W_i has nonvanishing area and $p_i \neq CM_\phi(W_i)$. In other words, the centroid locations $CM_\phi(W_1), \dots, CM_\phi(W_n)$ are optimal for $\mathcal{H}_{\text{distor}}$ among all configurations in S .

Note that when $n = 1$, the node location that optimizes $p \mapsto \mathcal{H}_{\text{distor}}(p)$ is the centroid of the set S , denoted by $CM_\phi(S)$.

Recall that the gradient of $\mathcal{H}_{\text{distor}}$ on $S^n \setminus \mathcal{C}$ takes the form,

$$\begin{aligned} \frac{\partial \mathcal{H}_{\text{distor}}}{\partial p_i}(P) = 2A_\phi(V_i(P))(CM_\phi(V_i(P)) - p_i), \\ i \in \{1, \dots, n\}, \end{aligned}$$

that is, the i th component of the gradient points in the direction of the vector going from p_i to the centroid of its Voronoi cell. The critical points of $\mathcal{H}_{\text{distor}}$ are therefore the set of centroidal Voronoi configurations in S . This is a natural generalization of the result for the case $n = 1$, where the optimal node location is the centroid $CM_\phi(S)$.

Area Problem

For $r \in \mathbb{R}_{>0}$, consider the performance function $f(x) = 1_{[0,r]}(x)$, that is, the indicator function of the closed interval $[0, r]$. Then, the expected-value multicenter function becomes

$$\begin{aligned} \mathcal{H}_{\text{area},r}(p_1, \dots, p_n) = \sum_{i=1}^n A_\phi(V_i(P) \cap \overline{B}(p_i, r)) \\ = A_\phi(\cup_{i=1}^n \overline{B}(p_i, r)), \end{aligned}$$

which corresponds to the area, measured according to ϕ , covered by the union of the n balls $\overline{B}(p_1, r), \dots, \overline{B}(p_n, r)$.

Let us see how the computation of the partial derivatives of $\mathcal{H}_{\text{area},r}$ specializes in this case. Here, the performance function is differentiable everywhere except at a single discontinuity, and its derivative is identically zero. Therefore, the first term in (5) vanishes. The gradient of $\mathcal{H}_{\text{area},r}$ on $S^n \setminus \mathcal{C}$ then takes the form, for each $i \in \{1, \dots, n\}$,

$$\frac{\partial \mathcal{H}_{\text{area},r}}{\partial p_i}(P) = \int_{V_i(P) \cap \partial \bar{B}(p_i,r)} \mathbf{n}_{\text{out}}(q) \phi(q) dq,$$

where $\mathbf{n}_{\text{out}}$ is the outward normal vector to $\bar{B}(p_i, r)$. The critical points of $\mathcal{H}_{\text{area},r}$ correspond to configurations with the property that each p_i is a local maximum for the area of $V_{i,r}(P) = V_i(P) \cap \bar{B}(p_i, r)$ at fixed $V_i(P)$. We refer to these configurations as *r-limited area-centered Voronoi configurations*.

Optimal Deployment Algorithms

Once a set of optimal deployment configurations have been characterized, the next step is to devise a distributed algorithm that allows a group of mobile robots to converge to such configurations. Gradient algorithms are the first of the options that should be explored.

For the expected-value multicenter functions, robots whose dynamics can be described by first-order integrator dynamics and which can communicate at predetermined *communication rounds* of a fixed time schedule, these laws present a similar structure, loosely described as follows:

[*Informal description*] In each communication round, each robot performs the following tasks: (i) it transmits its position and receives its neighbors' positions; (ii) it computes a notion of the geometric center of its own cell, determined according to some notion of partition of the environment. (iii) Between communication rounds, each robot moves toward this center.

The notions of geometric center and of partition of the environment differ depending on what is the type of expected-value multicenter function used. In the *Voronoi-center deployment algorithm*, the geometric center just reduces to $CM_\phi(V_i)$. In the *limited-Voronoi-normal* deployment problem in (ii), each agent computes the direction of $v = \frac{\partial \mathcal{H}_{\text{area},r}}{\partial p_i}$ for some r and (iii) moves for a maximum step size in this direction to ensure the area function will be decreased.

The Voronoi-center deployment algorithm achieves convergence of a set of nodes to a

centroidal Voronoi configuration, thus maximizing the expected-value multicenter function $\mathcal{H}_{\text{distor}}$. The algorithm is distributed over the proximity graph $\mathcal{G}_D$, as the computation of the centroids requires information in $\mathcal{N}_{\mathcal{G}_D}(p_i)$, for each $i \in \{1, \dots, n\}$. Additional properties of this algorithm are that the algorithm is adaptive to agent departures or arrivals and amenable to asynchronous implementations.

On the other hand, the limited-Voronoi-normal deployment algorithm achieves convergence to a set that locally maximizes the area covered by the set of sensing balls. The algorithm is distributed in the sense that agents only need to know information from neighbors in the proximity graph $\mathcal{G}_{2r}$ or, more precisely, $\mathcal{G}_{LD,r}$. Thus, it can be implemented by agents that employ range-limited interactions. It enjoys similar robustness properties as the Voronoi-center deployment algorithm.

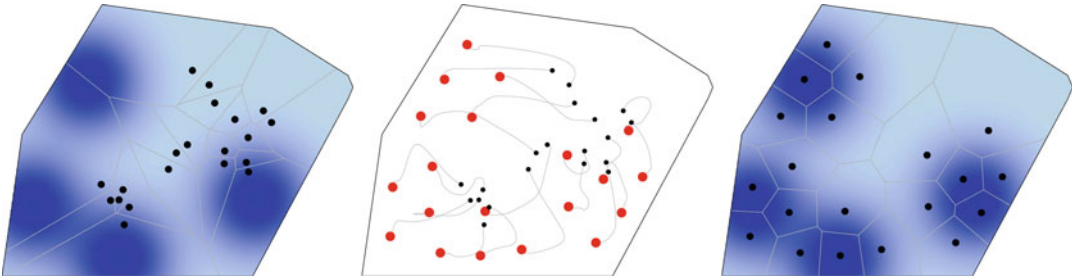
Simulation Results

We show evolutions of the Voronoi-centroid deployment algorithm in Fig. 1. One can verify that the final network configuration is a centroidal Voronoi configuration. For each evolution we depict the initial positions, the trajectories, and the final positions of all robots.

Finally, we show an evolution of limited-Voronoi-normal deployment algorithm in Fig. 2. One can verify that the final network configuration is an $\frac{r}{2}$ -limited area-centered Voronoi configuration. In other words, the deployment task is achieved.

Future Directions for Research

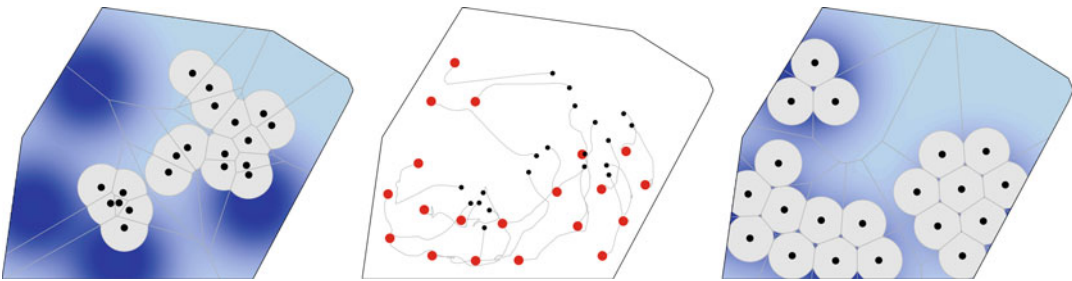
The algorithms described above achieve locally optimal deployment configurations with respect to expected-value multicenter functions. However, this simplified setting does not account for many important constraints, such as obstacles and deployment in non-convex environments (Pimenta et al. 2008; Caicedo-Núñez and Žefran 2008), deployment with visibility sensors, range-limited and wedge-shaped footprints (Ganguli et al. 2006; Laventall



Optimal Deployment and Spatial Coverage, Fig. 1

The evolution of the Voronoi-centroid deployment algorithm with $n = 20$ robots. The *left-hand* (resp., *right-hand*) figure illustrates the initial (resp., final) locations

and Voronoi partition. The central figure illustrates the evolution of the robots. After 13 s, the value of $\mathcal{H}_{\text{distor}}$ has monotonically increased to approximately -0.515



Optimal Deployment and Spatial Coverage, Fig. 2

The evolution of the limited-Voronoi-normal deployment algorithm with $n = 20$ robots and $r = 0.4$. The *left-hand* (resp., *right-hand*) figure illustrates the initial (respectively, final) locations and Voronoi partition. The

central figure illustrates the evolution of the robots. The $\frac{r}{2}$ -limited Voronoi cell of each robot is plotted in *light gray*. After 36 s, the value of $\mathcal{H}_{\text{area}}$, with $a = \frac{r}{2}$, has monotonically increased to approximately 14.141

and Cortés 2009), and energy and vehicle dynamical restrictions (Kwok and Martínez 2010a, b). Deployment strategies find application in exploration and data gathering tasks, and so these algorithms have been expanded to account for uncertainty and learning of unknown density functions (Schwager et al. 2009; Graham and Cortés 2012; Zhong and Cassandras 2011; Martínez 2010). Gossip and self-triggered communications (Bullo et al. 2012; Nowzari and Cortés 2012), self-triggered computations for region approximation (Ru and Martínez 2013), and area equitable partitions (Cortés 2010) have also been investigated. Much work is currently being devoted to solve on the current limitations of these nontrivial extensions, which make the problem settings significantly harder to solve.

Cross-References

- [Graphs for Modeling Networked Interactions](#)
- [Multi-vehicle Routing](#)
- [Networked Systems](#)

Bibliography

- Bullo F, Cortés J, Martínez S (2009) Distributed control of robotic networks. Applied mathematics series. Princeton University Press. Available at <http://www.coordinationbook.info>
- Bullo F, Carli R, Frasca P (2012) Gossip coverage control for robotic networks: dynamical systems on the space of partitions. SIAM J Control Optim 50(1): 419–447
- Caicedo-Núñez CH, Žefran M (2008) Performing coverage on nonconvex domains. In: IEEE conference on control applications, San Antonio, pp 1019–1024

- Cortés J (2010) Coverage optimization and spatial load balancing by robotic sensor networks. *IEEE Trans Autom Control* 55(3):749–754
- Cortés J, Martínez S, Bullo F (2005) Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM Control Optim Calc Var* 11: 691–719
- Drezner Z (ed) (1995) Facility location: a survey of applications and methods. Springer series in operations research. Springer, New York
- Du Q, Faber V, Gunzburger M (1999) Centroidal Voronoi tessellations: applications and algorithms. *SIAM Rev* 41(4):637–676
- Ganguli A, Cortés J, Bullo F (2006) Distributed deployment of asynchronous guards in art galleries. In: American control conference, Minneapolis, pp 1416–1421
- Graham R, Cortés J (2012) Cooperative adaptive sampling of random fields with partially known covariance. *Int J Robust Nonlinear Control* 22(5): 504–534
- Gray RM, Neuhoff DL (1998) Quantization. *IEEE Trans Inf Theory* 44(6):2325–2383. Commemorative Issue 1948–1998
- Hibbeler R (2006) Engineering mechanics: statics & dynamics, 11th edn. Prentice Hall, Upper Saddle River
- Kwok A, Martínez S (2010a) Deployment algorithms for a power-constrained mobile sensor network. *Int J Robust Nonlinear Control* 20(7): 725–842
- Kwok A, Martínez S (2010b) Unicycle coverage control via hybrid modeling. *IEEE Trans Autom Control* 55(2):528–532
- Laventall K, Cortés J (2009) Coverage control by multi-robot networks with limited-range anisotropic sensory. *Int J Control* 82(6):1113–1121
- Martínez S (2010) Distributed interpolation schemes for field estimation by mobile sensor networks. *IEEE Trans Control Syst Technol* 18(2): 491–500
- Nowzari C, Cortés J (2012) Self-triggered coordination of robotic networks for optimal deployment. *Automatica* 48(6):1077–1087
- Pimenta L, Kumar V, Mesquita R, Pereira G (2008) Sensing and coverage for a network of heterogeneous robots. In: IEEE international conference on decision and control, Cancun, pp 3947–3952
- Ru Y, Martínez S (2013) Coverage control in constant flow environments based on a mixed energy-time metric. *Automatica* 49:2632–2640
- Schwager M, Rus D, Slotine J (2009) Decentralized, adaptive coverage control for networked robots. *Int J Robot Res* 28(3):357–375
- Zhong M, Cassandras C (2011) Distributed coverage control and data collection with mobile sensor networks. *IEEE Trans Autom Control* 56(10): 2445–2455

Optimal Sampled-Data Control

Yutaka Yamamoto

Graduate School of Informatics, Kyoto University, Kyoto, Japan

Abstract

This entry gives a brief overview on the modern development of sampled-data control. Sampled-data systems intrinsically involve a mixture of two different time sets, one continuous and the other discrete. Due to this, sampled-data systems cannot be characterized in terms of the standard notions of transfer functions, steady-state response, or frequency response. The technique of lifting resolves this difficulty and enables the recovery of such concepts and simplified solutions to sampled-data H^∞ and H^2 optimization problems. We review the lifting point of view and its application to such optimization problems and finally present an instructive numerical example.

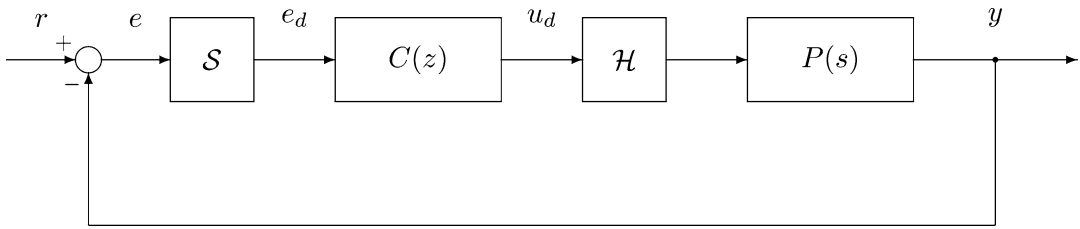
Keywords

Computer control · Frequency response · H^∞ and H^2 optimization · Lifting · Transfer operator

Introduction

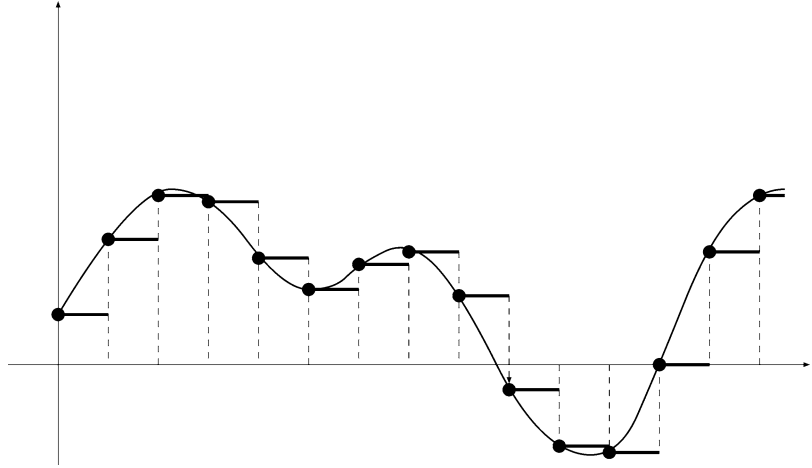
A sampled-data control system consists of a continuous-time plant and a discrete-time controller, with sample and hold devices that serve as an interface between these two components. As can be seen from this fact, sampled-data systems are *not* time-invariant, and various problems arise from this property.

To be more specific, consider the unity-feedback control system shown in Fig. 1; r is the reference signal, y the system output, and e the error signal. These are continuous-time signals. The error $e(t)$ goes through the *sampler*



Optimal Sampled-Data Control, Fig. 1 A unity-feedback system

Optimal Sampled-Data Control, Fig. 2 Sampling with 0-order hold



(or an *A/D converter*) $\mathcal{S}$. This sampler reads out the values of $e(t)$ at every time step h called the *sampling period* and produces a discrete-time signal $e_d[k]$, $k = 0, 1, 2, \dots$ (see Fig. 2). In particular, the sampling operator $\mathcal{S}$ acts on a continuous-time signal $w(t)$, $t \geq 0$, as

$$\mathcal{S}(w)[k] := w(kh), \quad k = 0, 1, 2, \dots$$

The discretized signal is then processed by the discrete-time controller $C(z)$ and becomes a control input u_d . There can also be a quantization effect, although for the sake of simplicity, it is neglected here. The obtained signal u_d then goes through another interface $\mathcal{H}$ called a *hold device* or a *D/A converter* to become a continuous-time signal. A typical example is the *0-order hold* where $\mathcal{H}$ simply maintains the value of a discrete-time signal $w[k]$ constant as its output until the next sampling time:

$$(\mathcal{H}(w[k]))(t) := w[k], \quad \text{for } kh \leq t < (k+1)h.$$

A typical sample-and-hold action is shown in Fig. 2.

While one can consider a nonlinear plant P or controller C , or infinite-dimensional P and C , we confine ourselves to linear and finite-dimensional P and C and also suppose that P and C are time-invariant in continuous-time and in discrete-time, respectively.

The Main Difficulty

As stated above, the unity-feedback system Fig. 1 is not time-invariant either in continuous-time or in discrete-time, even when the plant and controller are both time-invariant in their respective domains of operators. The mixture of the two time sets prohibits the total closed-loop system from being time-invariant.

The lack of time-invariance implies that we cannot naturally associate to sampled-data systems such classical concepts of transfer functions, steady-state response and frequency response.

One can regard Fig. 1 as a time-invariant discrete-time system by ignoring the intersample behavior and focusing attention on the sample-point behavior only. But the obtained model does not then reflect what happens between sampling times. This approach can lead to the neglect of undesirable intersample oscillations, called *ripples*. To monitor the intersample behavior, the notion of the *modified z-transform* was introduced; see, e.g., Jury (1958) and Ragazzini and Franklin (1958); however, this transform is usable only *after the controller has been designed* and hence not for the design problems considered in this entry.

Lifting: A Modern Approach

A new approach was introduced around 1990–1991 (Bamieh et al. 1991; Tadmor 1991; Toivonen 1992; Yamamoto 1990, 1994). The new idea, now called *lifting*, makes it possible to describe sampled-data systems via a *time-invariant model while maintaining the intersample behavior*.

Let $f(t)$ be a continuous-time signal. Instead of sampling $f(t)$, we will represent it as a *sequence of functions*. Namely, we set up the correspondence (see Fig. 3):

$$\begin{aligned} \mathcal{L} : f &\mapsto \{f[k](\theta)\}_{k=0}^{\infty}, \\ f[k](\theta) &= f(kh + \theta), \quad 0 \leq \theta < h. \end{aligned} \quad (1)$$

This idea makes it possible to view a (time-invariant or even periodically time-varying) *continuous-time* system as a linear, *time-invariant discrete-time* system. Let

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t). \end{aligned} \quad (2)$$

be a given continuous-time plant and lift the input $u(t)$ to obtain $u[k](\cdot)$. We apply this lifted input with the timing $t = kh$ (h is the prespecified sampling rate as above) and observe how it affects the system. Let $x[k]$ be the state at time $t = kh$. The state $x[k+1]$ at time $(k+1)h$ is given by

$$x[k+1] = e^{Ah}x[k] + \int_0^h e^{A(h-\tau)}Bu[k](\tau)d\tau. \quad (3)$$

The right-hand side integral defines an operator $L^2[0, h] \rightarrow \mathbb{R}^n : u(\cdot) \mapsto \int_0^h e^{A(h-\tau)}Bu(\tau)d\tau$. While the state-transition (3) only describes a discrete-time update, the system keeps producing an output during the intersample period. If we consider the lifting of $x(t)$, it is seen to be described by

$$x[k](\theta) = e^{A\theta}x[k] + \int_0^\theta e^{A(\theta-\tau)}Bu[k](\tau)d\tau.$$

As such, the lifted output $y[k](\cdot)$ is given by

$$\begin{aligned} y[k](\theta) &= Ce^{A\theta}x[k] \\ &\quad + \int_0^\theta Ce^{A(\theta-\tau)}Bu[k](\tau)d\tau. \end{aligned} \quad (4)$$

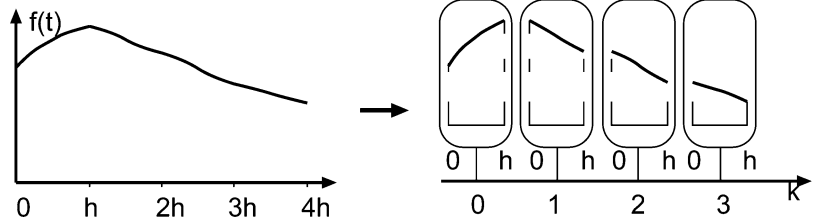
Observe that formulas (3) and (4) take the form

$$\begin{aligned} x[k+1] &= \mathcal{A}x[k] + \mathcal{B}u[k] \\ y[k] &= \mathcal{C}x[k] + \mathcal{D}u[k], \end{aligned}$$

and the operators $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ do not depend on the time variable k . In other words, it is possible to describe this continuous-time system with discrete timing, once we adopt the lifting point of view. To be more precise, the operators $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ are defined as follows:

$$\begin{aligned} \mathcal{A} : \mathbb{R}^n &\rightarrow \mathbb{R}^n : x \mapsto e^{Ah}x \\ \mathcal{B} : L^2[0, h] &\rightarrow \mathbb{R}^n : u \mapsto \int_0^h e^{A(h-\tau)}Bu(\tau)d\tau \\ \mathcal{C} : \mathbb{R}^n &\rightarrow L^2[0, h] : x \mapsto Ce^{A(\cdot)}x \\ \mathcal{D} : L^2[0, h] &\rightarrow L^2[0, h] : u \\ &\mapsto \int_0^\cdot Ce^{A(\cdot-\tau)}Bu(\tau)d\tau \end{aligned} \quad (5)$$

Thus the continuous-time plant (2) can be described by a *time-invariant* discrete-time model. Once this is done, it is straightforward to connect this expression with a discrete-time controller, and hence, sampled-data systems (for example, Fig. 1) can be fully described by time-invariant discrete-time equations, without

Optimal Sampled-Data Control, Fig. 3 Lifting


discarding the intersampling information. We will also denote the overall equation (with discrete-time controller included) abstractly in the form

$$\begin{aligned} x[k+1] &= \mathcal{A}x[k] + \mathcal{B}u[k] \\ y[k] &= \mathcal{C}x[k] + \mathcal{D}u[k]. \end{aligned} \quad (6)$$

While the obtained discrete-time model is a time-invariant, the input and output spaces are now infinite dimensional. Its *transfer function (operator)* is defined as

$$G(z) := \mathcal{D} + \mathcal{C}(zI - \mathcal{A})^{-1}\mathcal{B}. \quad (7)$$

Note that $\mathcal{A}$ in (6) is a matrix because it is so for $\mathcal{A}$ in (5). Hence, (6) is stable if $G(z)$ is analytic for $\{z : |z| \geq 1\}$, provided that there is no unstable pole-zero cancellation.

Definition 1 Let $G(z)$ be the transfer operator of the lifted system given by (7), which is stable in the sense above. The *frequency response operator* is the operator

$$G(e^{j\omega h}) : L^2[0, h] \rightarrow L^2[0, h] \quad (8)$$

regarded as a function of $\omega \in [0, \omega_s)$ ($\omega_s := 2\pi/h$). Its *gain* at ω is defined to be

$$\|G(e^{j\omega h})\| = \sup_{v \in L^2[0, h]} \frac{\|G(e^{j\omega h})v\|}{\|v\|}. \quad (9)$$

The maximum $\|G(e^{j\omega h})\|$ over $[0, \omega_s)$ is the H^∞ norm of $G(z)$. The H^2 -norm of G is defined by

$$\begin{aligned} \|G\|_2 : \\ = \left(\frac{h}{2\pi} \int_0^{2\pi/h} \text{trace} \{G^*(e^{j\omega h})G(e^{j\omega h})\} d\omega \right)^{1/2}, \end{aligned} \quad (10)$$

where the *trace* here is taken in the sense of the Hilbert-Schmidt norm; see Chen and Francis (1995) for details.

H^∞ and H^2 Control Problems

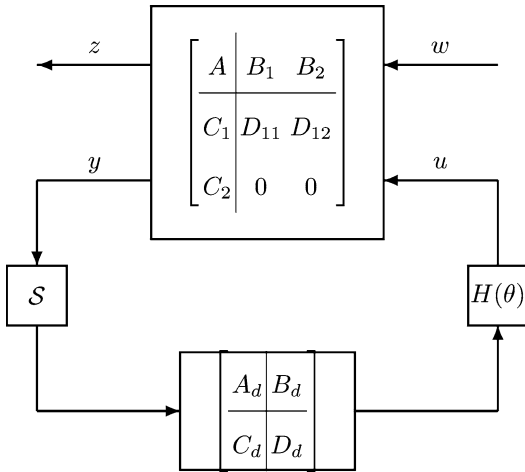
A significant consequence of the lifting approach described above is that various robust control problems such as H^∞ and H^2 control problems for sampled-data control systems can be converted to corresponding discrete-time (finite-dimensional) problems. The approach was initiated by Chen and Francis (1990) and later solved by Bamieh and Pearson (1992), Kabamba and Hara (1993), Sivashankar and Khargonekar (1994), Tadmor (1991), and Toivonen (1992) in more complete forms; see Chen and Francis (1995) for the pertinent historical accounts.

Let us introduce the notion of generalized plants. Suppose that a continuous-time plant is given in the following model:

$$\begin{aligned} \dot{x}_c(t) &= Ax_c(t) + B_1w(t) + B_2u(t) \\ z(t) &= C_1x_c(t) + D_{11}w(t) + D_{12}u(t) \\ y(t) &= C_2x_c(t) \end{aligned} \quad (11)$$

Here w is the exogenous input, $u(t)$ control input, $y(t)$ measured output, and $z(t)$ is the controlled output. The objective is to design a controller that takes the sampled measurements of y and returns a control variable u according to the following formulas:

$$\begin{aligned} x_d[k+1] &= A_dx_d[k] + B_dSy[k] \\ v[k] &= C_dx_d[k] + D_dSy[k] \\ u[k](\theta) &= H(\theta)v[k] \end{aligned} \quad (12)$$



Optimal Sampled-Data Control, Fig. 4 Sampled feedback system

where $H(\theta)$ is a suitable hold function. This is depicted in Fig. 4. The objective here is to design or characterize a controller that achieves a prescribed performance level $\gamma > 0$ in such a way that

$$\|T_{zw}\|_\infty < \gamma \quad (13)$$

where T_{zw} denotes the closed-loop transfer operator from w to z . This is the H^∞ control problem for sampled-data systems. If we take the H^2 -norm (10) instead, then the problem becomes that of the H^2 (sub)optimal control problem.

The difficulty here is that both w and z are continuous-time variables, and hence their lifted variables are infinite dimensional. A remarkable fact here is that the H^∞ problem (and the H^2 problem as well) (13) can be equivalently transformed to an H^∞ problem for a *finite-dimensional* discrete-time system. We will indicate in the next section how this can be done.

H^∞ Norm Computation and Reduction to Finite Dimension

Let us write the system (11) and (12) in the form

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] \\ y[k] &= Cx[k] + Du[k]. \end{aligned} \quad (14)$$

as in (6). For simplicity of treatments, assume D_{11} in (11) to be zero; for the general case, see Yamamoto and Khargonekar (1996).

Let $G(z)$ be the transfer operator $G(z) := \mathcal{D} + \mathcal{C}(zI - \mathcal{A})^{-1}\mathcal{B}$. The H^∞ norm of G is given as the maximum of the singular values of the gain $G(e^{j\omega h})$ for $\omega \in [0, 2\pi/h]$.

Now consider the singular value equation

$$(\gamma^2 I - G^* G(e^{j\omega h}))w = 0. \quad (15)$$

and suppose that $\gamma > \|\mathcal{D}\|$. A crux here is that $\mathcal{A}, \mathcal{B}, \mathcal{C}$ are finite-rank operators, and we can reduce this to a finite-dimensional rank condition. Taking the adjoint of (14), we obtain

$$p[k] = \mathcal{A}^* p_{k+1} + \mathcal{C}^* v[k]$$

$$e[k] = \mathcal{B}^* p_{k+1} + \mathcal{D}^* v[k].$$

Taking the z -transforms of both sides, setting $z = e^{j\omega h}$, and substituting $v = y$ and $e = \gamma^2 w$, we obtain

$$e^{j\omega h} x = \mathcal{A}x + \mathcal{B}w$$

$$p = e^{j\omega h} \mathcal{A}^* p + \mathcal{C}^* (\mathcal{C}x + \mathcal{D}w)$$

$$(\gamma^2 - \mathcal{D}^* \mathcal{D})w = e^{j\omega h} \mathcal{B}^* p + \mathcal{D}^* \mathcal{C}x.$$

Eliminating the variable w then yields

$$\begin{aligned} & \left(e^{j\omega h} \begin{bmatrix} I & \mathcal{B}R_\gamma^{-1}\mathcal{B}^* \\ 0 & \mathcal{A}^* + \mathcal{C}^* \mathcal{D}R_\gamma^{-1}\mathcal{B}^* \end{bmatrix} \right. \\ & \quad \left. - \begin{bmatrix} \mathcal{A} + \mathcal{B}R_\gamma^{-1}\mathcal{D}^* \mathcal{C} & 0 \\ \mathcal{C}^* (I + \mathcal{D}R_\gamma^{-1}\mathcal{D}^*) \mathcal{C} & I \end{bmatrix} \right) \begin{bmatrix} x \\ p \end{bmatrix} = 0 \end{aligned} \quad (16)$$

where $R_\gamma = (\gamma I - \mathcal{D}^* \mathcal{D})$. The important point to be noted here is that all the operators appearing here are actually matrices. For example, $\mathcal{B}$ is an operator from $L^2[0, h]$ to $\mathbb{R}^n$, and its adjoint $\mathcal{B}^*$ is an operator from $\mathbb{R}^n$ to $L^2[0, h]$. Hence, the composition $\mathcal{B}R_\gamma^{-1}\mathcal{B}^*$ is a linear operator from $\mathbb{R}^n$ into itself, i.e., a matrix. Thus, for a given γ , the singular value equation admits a nontrivial

solution w for (15) if and only if the *finite-dimensional equation* (16) admits a nontrivial solution $[x \ p]^T$ (Yamamoto 1993; Yamamoto and Khargonekar 1996). (Note that R_γ is invertible since $\gamma > \|\mathcal{D}\|$.)

It is possible to find matrices $\bar{A}$, $\bar{B}$, $\bar{C}$ such that $\bar{A} = \mathcal{A} + \mathcal{B}R_\gamma^{-1}\mathcal{D}^*\mathcal{C}$, $\bar{B}\bar{B}^*/\gamma^2 = \mathcal{B}R_\gamma^{-1}\mathcal{B}^*$, and $\bar{C}^*\bar{C} = \mathcal{C}^*(I + \mathcal{D}R_\gamma^{-1}\mathcal{D}^*)\mathcal{C}$, and hence (16) is equivalent to

$$\left(\lambda \begin{bmatrix} I & -\bar{B}\bar{B}^*/\gamma^2 \\ 0 & \bar{A}^* \end{bmatrix} - \begin{bmatrix} \bar{A} & 0 \\ -\bar{C}^*\bar{C} & I \end{bmatrix} \right) \begin{bmatrix} x \\ p \end{bmatrix} = 0 \quad (17)$$

for $\lambda = e^{j\omega h}$. In other words, we have that $\|G\|_\infty < \gamma$ if and only if there exists no λ of modulus 1 such that (17) holds.

It can be proven that by substituting the expressions of (11) and (12) for $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$, one obtains a finite-dimensional discrete-time generalized plant G_d with digital controller (12) such that $\|G\|_\infty < \gamma$ if and only if $\|G_d\|_\infty < \gamma$. The precise formulas for the discrete-time plant can be found, e.g., Bamieh and Pearson (1992), Chen and Francis (1995), Kabamba and Hara (1993), Yamamoto and Khargonekar (1996), and Cantoni and Glover (1997).

An H^∞ Design Example

For sampled-data control systems, there used to be, and still is, a rather common myth that if one takes a sufficiently fast sampling rate, it will not cause a major problem. This can be true for continuous-time design, but we here show that if we employ a sample-point discretization without a performance consideration for intersampling behavior, fast sampling rates can cause a serious problem.

Take a simple second-order plant $P(s) = 1/(s^2 + 0.1s + 1)$ and consider the disturbance rejection problem minimizing the H^∞ -norm from w to z as given in Fig. 5. Set the sampling time $h = 0.5$. We execute the following:

- Sampled-data H^∞ design with the generalized plant

$$G(s) = \begin{bmatrix} P(s) & P(s) \\ P(s) & P(s) \end{bmatrix},$$

- Discrete-time H^∞ design with the discrete-time generalized plant $G_d(z)$ given by the step-invariant transformation (see, e.g., Chen and Francis 1995) of $G(s)$.

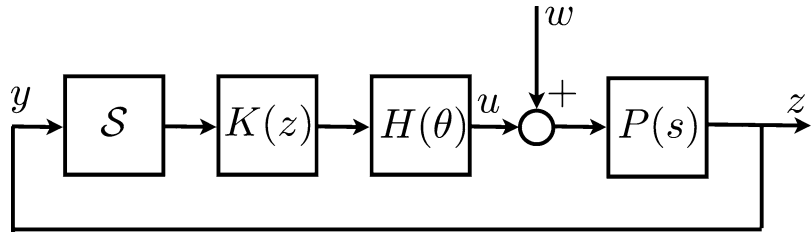
Figures 6 and 7 show the frequency and time responses of the two resulting closed-loop systems, respectively. In Fig. 6, the solid curve shows the response of the sampled design, while the dash-dotted curve shows the discrete-time frequency response, but purely reflecting its sample-point behavior only. At first glance, it may appear that the discrete-time design performs better. But when we actually compute the lifted sampled-data frequency response in the sense defined in Definition 1, it becomes obvious that the sampled-data design is far superior. The dashed curve shows the frequency response of the closed-loop, i.e., that of $G(s)$ connected with the discrete-time designed K_d . The response is similar to the discrete-time frequency response in low frequency but exhibits a very sharp peak around the Nyquist frequency (i.e., half the sampling frequency; in the present case, $\pi/h \sim 6.28$ rad/s, i.e., $1/2h = 1$ Hz).

This can also be verified from the initial-state responses Fig. 7 with $x(0) = (1, 1)$. The solid curve shows the sampled-data design and the dashed curve the discrete-time one. Both responses decay to zero rapidly *at sampled instants* as shown by the circles for the discrete-time design. But the discrete-time design exhibits very large ripples, with period approximately 1 s. This corresponds to 1 Hz, which is the same as $2\pi = \pi/h$ [rad/s], i.e., the Nyquist frequency. This is precisely captured in the lifted frequency response in Fig. 6.

It is worth noting that when we take the sampling period h smaller, the response for the discrete-time design becomes even more oscillatory and shows a very high peak in the frequency response.

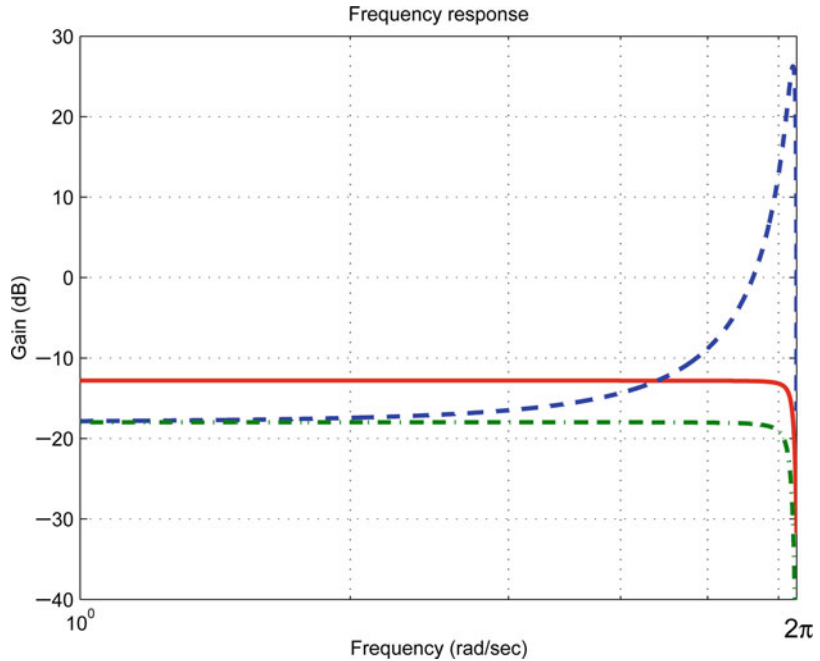
Optimal Sampled-Data Control, Fig. 5

Disturbance rejection



Optimal Sampled-Data Control, Fig. 6

Frequency responses $h = 0.5$



Summary, Bibliographical Notes, and Future Directions

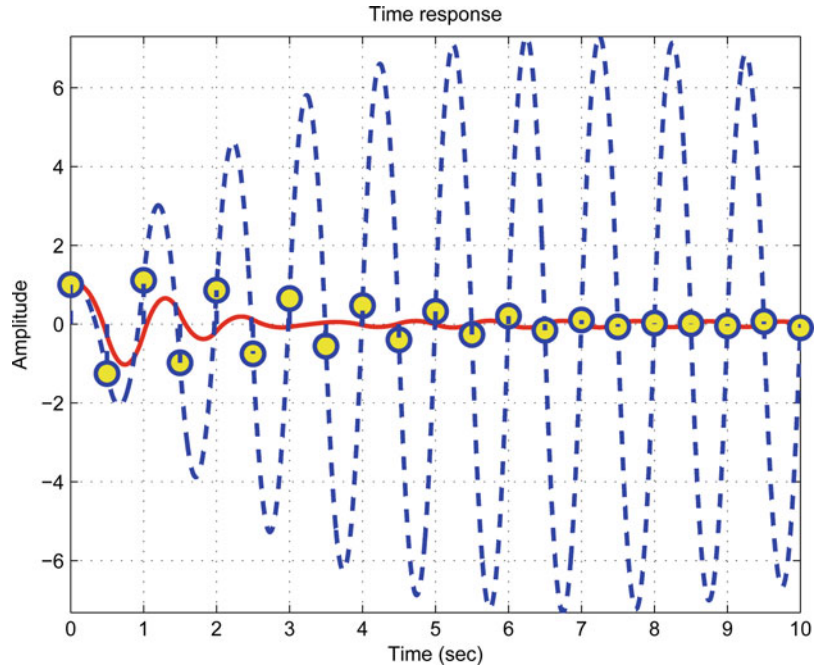
We have given a short summary of the main achievements of modern sampled-data control theory. Particularly, we have reviewed how the technique of lifting resolved the intrinsic difficulty arising from the mixture of two distinct time sets: continuous and discrete. This idea further led to the new notions of transfer operators and frequency response. These notions together enabled us to treat optimal sampled-data control problems in a unified and transparent way. We have outlined how the sampled-data H^∞ control problem can equivalently be reduced to a corresponding discrete-time H^∞ problem, without sacrificing the performance in the intersample behavior. This has been exemplified by a numerical example.

For classical treatments of sampled-data control, it is instructive to consult Jury (1958) and Ragazzini and Franklin (1958). The textbook Åström and Wittenmark (1996) covers both classical and modern aspects of digital control. For a mathematical background of the computation of adjoints treated in section “ H^∞ Norm Computation and Reduction to Finite Dimension,” consult Yamamoto (2012) as well as Yamamoto (1993).

There are other performance indices for optimality, typically those arising from H^2 and L^1 norms. These problems have also been studied extensively, and fairly complete solutions are available. For the lack of space, we cannot list all references, and the reader is referred to Chen and Francis (1995) and Yamamoto (1999), Yamamoto and Nagahara (2017) for a more concrete survey and references therein.

Optimal Sampled-Data Control, Fig. 7

Initial-state responses
 $h = 0.5$



For actual computation of these optimal control problems, we often resort to the so-called fast-sample/fast-hold approximation. For this technique, consult Anderson and Keller (1998), and also Yamamoto et al. (1999, 2002). The latter two discuss convergence properties of such approximations. Recently Hagiwara and coworkers (Hagiwara and Umeda 2008; Kim and Hagiwara 2016) introduced a modified version of fast-sample/fast-hold approximation by combining them with lifting.

It is also known that sampled-data systems exhibit some odd unstable zeros when sampled and combined with a (zero-order) hold (Åström et al. 1984). Much research has been conducted to circumvent this effect via multirate techniques, e.g., Mita et al. (1990). It has been also shown that such zeros disappear if one considers a lifted sampled-data model of the continuous-time system (Yamamoto et al. 2017a).

Until quite recently, the Nyquist frequency was regarded to be the upper bound for control, based on the sampling theorem. On the other hand, there are varied situations in which one needs to track or reject those signals beyond the Nyquist frequency. Control of hard-disc drives

where one needs to reject disturbance beyond the Nyquist frequency is such an example (Atsumi 2010). Some efforts have been made recently by utilizing a proper signal model and a multirate techniques (Yamamoto et al. 2016, 2017b; Zheng et al. 2016). See also Nagahara and Yamamoto (2016) for an application to repetitive control systems.

For nonlinear systems, the situation is quite different. At the moment, there does not seem to be a model similar to the lifted model discussed here. See, e.g., Nesić and Teel (2007), Monaco and Normand-Cyrot (1995), and Yuz and Goodwin (2005).

Sampled-data control has much to do with signal processing. Indeed, since it can optimize continuous-time performance, it can shed new light on digital signal processing. Traditionally, Shannon's paradigm is based on the perfect band-limiting hypothesis, and the sampling theorem has been prevalent in the signal processing community. Since the sampling theorem opts for perfect reconstruction, the resulting theory reduces mostly to discrete-time problems. In other words, the intersample information is buried in the sampling theorem. It should,

however, be noted that the very stringent band-limiting hypothesis is almost never satisfied in reality, and various approximations are necessitated. In contrast, sampled-data control can provide an optimal platform for dealing with and optimizing the response between sampling points when the band-limiting hypothesis does not hold. See, for example, Yamamoto et al. (2012) and Nagahara and Yamamoto (2012) for the idea and some efforts in this direction.

Cross-References

- [Control Applications in Audio Reproduction](#)
- [H₂ Optimal Control](#)
- [H-Infinity Control](#)
- [Optimal Control via Factorization and Model Matching](#)

Acknowledgments The author would like to thank Masaaki Nagahara and Masashi Wakaiki for their help in the numerical example references.

Bibliography

- Anderson BDO, Keller JP (1998) Discretization techniques in control systems, control and dynamic systems. Academic, New York
- Åström KJ, Wittenmark B (1996) Computer controlled systems—theory and design, 3rd edn. Prentice Hall, Upper Saddle River
- Åström KJ, Hagander P, Sternby J (1984) Zeros of sampled systems. *Automatica* 20:31–38
- Atsumi T (2010) Disturbance suppression beyond Nyquist frequency in hard disk drives. *Mechatronics* 20:67–73
- Bamieh B, Pearson JB (1992) A general framework for linear periodic systems with applications to H_∞ sampled-data control. *IEEE Trans Autom Control* 37:418–435
- Bamieh B, Pearson JB, Francis BA, Tannenbaum A (1991) A lifting technique for linear periodic systems with applications to sampled-data control systems. *Syst Control Lett* 17:79–88
- Cantoni M, Glover K (1997) H_∞ sampled-data synthesis and related numerical issues. *Automatica* 33:2233–2241
- Chen T, Francis BA (1990) On the $\mathcal{L}_2$ -induced norm of a sampled-data system. *Syst Control Lett* 15:211–219
- Chen T, Francis BA (1995) Optimal sampled-data control systems. Springer, New York
- Hagiwara T, Umeda H (2008) Modified fast-sample/fast-hold approximation for sampled-data system analysis. *Eur J Control* 14:286–296
- Jury EI (1958) Sampled-data control systems. Wiley, New York
- Kabamba PT, Hara S (1993) Worst case analysis and design of sampled data control systems. *IEEE Trans Autom Control* 38:1337–1357
- Kim JH, Hagiwara T (2016) L_1 discretization for sampled-data controller synthesis via piecewise linear approximation. *IEEE Trans Autom Control* 61:1143–1157
- Mita T, Chida Y, Kaku Y, Numasato H (1990) Two-delay robust digital control and its applications-avoiding the problem on unstable limiting zeros. *IEEE Trans Autom Control* 35:962–970
- Monaco S, Normand-Cyrot D (1995) A unified representation for nonlinear discrete-time and sampled dynamics. *J Math Syst Estimation Control* 5:1–27
- Nagahara M, Yamamoto Y (2012) Frequency domain min-max optimization of noise-shaping Delta-Sigma modulators. *IEEE Trans Signal Process* 60:2828–2839
- Nagahara M, Yamamoto Y (2016) Digital repetitive controller design via sampled-data delayed signal reconstruction *Automatica* 65:203–209
- Nesić D, Teel AR (2007) Sampled-data control of nonlinear systems: an overview of recent results. In: Lecture notes in control and information sciences, vol 268. Springer, New York, pp 221–239
- Ragazzini JR, Franklin GF (1958) Sampled-data control systems. McGraw-Hill, New York
- Sivashankar N, Khargonekar PP (1994) Characterization and computation of the $\mathcal{L}_2$ -induced norm of sampled-data systems. *SIAM J Control Optim* 32:1128–1150
- Tadmor G (1991) Optimal $\mathcal{H}_\infty$ sampled-data control in continuous-time systems. In: Proceedings of ACC'91, Boston, pp 1658–1663
- Toivonen HT (1992) Sampled-data control of continuous-time systems with an $\mathcal{H}_\infty$ optimality criterion. *Automatica* 28:45–54
- Yamamoto Y (1990) New approach to sampled-data systems: a function space method. In: Proceedings of 29th CDC, Honolulu, pp 1882–1887
- Yamamoto Y (1993) On the state space and frequency domain characterization of H^∞ -norm of sampled-data systems. *Syst Control Lett* 21:163–172
- Yamamoto Y (1994) A function space approach to sampled-data control systems and tracking problems. *IEEE Trans Autom Control* 39:703–712
- Yamamoto Y (1999) Digital control. In: Webster JG (ed) Wiley encyclopedia of electrical and electronics engineering, vol 5. Wiley, New York, pp 445–457
- Yamamoto Y (2012) From vector spaces to function spaces—introduction to functional analysis with applications. SIAM, Philadelphia
- Yamamoto Y, Khargonekar PP (1996) Frequency response of sampled-data systems. *IEEE Trans Autom Control* 41:166–176
- Yamamoto Y, Nagahara M (2017) Digital control In: Webster JG (ed) Wiley encyclopedia of electrical and electronics engineering online. <https://doi.org/10.1002/047134608X.W1010.pub2>

- Yamamoto Y, Madievski AG, Anderson BDO (1999) Approximation of frequency response for sampled-data control systems. *Automatica* 35:729–734
- Yamamoto Y, Anderson BDO, Nagahara M (2002) Approximating sampled-data systems with applications to digital redesign. In: *Proceedings of 41st CDC*, pp 3724–3729
- Yamamoto Y, Nagahara M, Khargonekar PP (2012) Signal reconstruction via H^∞ sampled-data control theory—beyond the Shannon paradigm. *IEEE Trans Signal Process* 60:613–625
- Yamamoto Y, Yamamoto K, Nagahara M (2016) Tracking of signals beyond the Nyquist frequency. In: *Proceedings of 55th IEEE CDC*, pp 4003–4008
- Yamamoto K, Yamamoto Y, Niculescu SI (2017a) A renewed look at zeros of sampled-data systems—from the lifting viewpoint In: *Proceedings of 20th IFAC World Congress*, pp 3731–3736
- Yamamoto K, Yamamoto Y, Nagahara M (2017b) Simultaneous rejection of signals below and above the Nyquist frequency. In: *Proceedings of 1st IEEE CCTA*, pp 1135–1139
- Yuz JI, Goodwin GC (2005) On sampled-data models for nonlinear systems. *IEEE Trans Autom Control* 50:1477–1489
- Zheng M, Sun L, Tomizuka M (2016) Multi-rate observer based sliding mode control with frequency shaping for vibration suppression beyond Nyquist frequency. *IFAC-PapersOnLine* 49:13–18

Optimization Algorithms for Model Predictive Control

Moritz Diehl
Department of Microsystems Engineering
(IMTEK), University of Freiburg, Freiburg,
Germany
ESAT-STADIUS/OPTEC, KU Leuven,
Leuven-Heverlee, Belgium

Abstract

This entry reviews optimization algorithms for both linear and nonlinear model predictive control (MPC). Linear MPC typically leads to specially structured convex quadratic programs (QP) that can be solved by structure exploiting active set, interior point, or gradient methods. Nonlinear MPC leads to specially structured nonlinear programs (NLP)

that can be solved by sequential quadratic programming (SQP) or nonlinear interior point methods.

Keywords

Banded matrix factorization · Convex optimization · Karush-Kuhn-Tucker (KKT) conditions · Sparsity exploitation

Introduction

Model predictive control (MPC) needs to solve at each sampling instant an optimal control problem with the current system state $\bar{x}_0$ as initial value. MPC optimization is almost exclusively based on the so-called *direct approach* which first discretizes the continuous time system to obtain a discrete time optimal control problem (OCP). This OCP has as optimization variables a state trajectory $X = [x_0^\top, \dots, x_N^\top]^\top$ with $x_i \in \mathbb{R}^{n_x}$ for $i = 0, \dots, N$ and a control trajectory $U = [u_0^\top, \dots, u_{N-1}^\top]^\top$ with $u_i \in \mathbb{R}^{n_u}$ for $i = 0, \dots, N - 1$. For simplicity of presentation, we restrict ourselves to the time-independent case, and the OCP we treat in this article is stated as follows:

$$\underset{X, U}{\text{minimize}} \quad \sum_{i=0}^{N-1} L(x_i, u_i) + E(x_N) \quad (1a)$$

$$\text{subject to} \quad x_0 - \bar{x}_0 = 0, \quad (1b)$$

$$x_{i+1} - f(x_i, u_i) = 0, \quad i = 0, \dots, N - 1, \quad (1c)$$

$$h(x_i, u_i) \leq 0, \quad i = 0, \dots, N - 1, \quad (1d)$$

$$r(x_N) \leq 0. \quad (1e)$$

The MPC objective is stated in Eq. (1a), the system dynamics enter via Eq. (1c), while path and terminal constraints enter via Eqs. (1d) and (1e). All functions are assumed to be differentiable and to have appropriate dimensions ($h(x, u) \in \mathbb{R}^{n_h}$ and $r(x) \in \mathbb{R}^{n_r}$). Note that $\bar{x}_0 \in \mathbb{R}^{n_x}$ is not an optimization variable, but a parameter upon

which the OCP depends via the initial value constraint in Eq. (1b). The optimal solution trajectories depend only on this value and can thus be denoted by $X^*(\bar{x}_0)$ and $U^*(\bar{x}_0)$. Obtaining them, in particular the first control value $u_0^*(\bar{x}_0)$, as fast and reliably as possible for each new value of $\bar{x}_0$ is the aim of all MPC optimization algorithms. The most important dividing line is between convex and non-convex optimal control problems (OCP). If the OCP is convex, algorithms exist that find a global solution reliably and in computable time. If the OCP is not convex, one usually needs to be satisfied with approximations of locally optimal solutions. The OCP (1) is convex if the objective (1a) and all components of the inequality constraint functions (1d) and (1e) are convex and if the equality constraints (1c) are linear.

We typically speak of *linear MPC* when the OCP to be solved is convex, and otherwise of *nonlinear MPC*.

General Algorithmic Features for MPC Optimization

In MPC we would dream to have the solution to a new optimal control problem instantly, which is impossible due to computational delays. Several ideas can help us to deal with this issue.

Off-line Precomputations and Code Generation

As consecutive MPC problems are similar and differ only in the value $\bar{x}_0$, many computations can be done once and for all before the MPC controller execution starts. Careful preprocessing and code optimization for the model routines is essential, and many tools automatically generate custom solvers in low-level languages. The generated code has fixed matrix and vector dimensions, has no online memory allocations, and contains a minimal number of if-then-else statements to ensure a smooth computational flow.

Delay Compensation by Prediction

When we know how long our computations for solving an MPC problem will take, it is a good idea *not* to address a problem starting at the current state but to simulate at which state

the system will be when we will have solved the problem. This can be done using the MPC system model and the open-loop control inputs that we will apply in the meantime. This feature is used in many practical MPC schemes with non-negligible computation time.

Division into Preparation and Feedback Phase

A third ingredient of several MPC algorithms is to divide the computations in each sampling time into a preparation phase and a feedback phase. The more CPU intensive *preparation phase* is performed with a predicted state $\bar{x}_0$, before the most current state estimate, say $\bar{x}'_0$, is available. Once $\bar{x}'_0$ is available, the *feedback phase* delivers quickly an *approximate* solution to the optimization problem for $\bar{x}'_0$.

Warmstarting and Shift

An obvious way to transfer solution information from one solved MPC problem to the next one uses the existing optimal solution as an initial guess to start the iterative solution procedure of the next problem. We can either directly use the existing solution without modification for warmstarting or we can first shift it in order to account for the advancement of time, which is particularly advantageous for systems with time-varying dynamics or objectives.

Iterating While the Problem Changes

A last important ingredient of some MPC algorithms is the idea to work on the optimization problem while it changes, i.e., to never iterate the optimization procedure to convergence for an MPC problem getting older and older during the iterations but to rather work with the most current information in each new iteration.

Convex Optimization for Linear MPC

Linear MPC is based on a linear system model of the form $x_{i+1} = Ax_i + Bu_i$ and convex objective and constraint functions in (1a), (1d), and (1e). The most widespread linear MPC setting uses a convex quadratic objective function and affine constraints and solves the following quadratic program (QP):

$$\begin{aligned} \underset{X, U}{\text{minimize}} \quad & \frac{1}{2} \sum_{i=0}^{N-1} \begin{bmatrix} x_i \\ u_i \end{bmatrix}^\top \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} x_i \\ u_i \end{bmatrix} + \frac{1}{2} x_N^\top P x_N \\ & (2a) \end{aligned}$$

$$\text{subject to} \quad x_0 - \bar{x}_0 = 0, \quad (2b)$$

$$x_{i+1} - A x_i - B u_i = 0, i = 0, \dots, N-1, \quad (2c)$$

$$b + C x_i + D u_i \leq 0, i = 0, \dots, N-1, \quad (2d)$$

$$c + F x_N \leq 0. \quad (2e)$$

Here, b, c are vectors and Q, S, R, P, C, D, F matrices, and matrices $\begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix}$ and P are symmetric and positive semi-definite to ensure the QP is convex.

Sparsity Exploitation

The QP (2) has a specific sparsity structure that can be exploited in different ways. One way is to reduce the variable space by a procedure called *condensing* and then to solve a smaller-scale QP instead of (2). Another way is to use a *banded matrix factorization*.

Condensing

The constraints (2b) and (2c) can be used to eliminate the state trajectory X . This yields an equivalent but smaller-scale QP of the following form:

$$\underset{U \in \mathbb{R}^{N n_u}}{\text{minimize}} \quad \frac{1}{2} \begin{bmatrix} U \\ \bar{x}_0 \end{bmatrix}^\top \begin{bmatrix} H & G \\ G^\top & J \end{bmatrix} \begin{bmatrix} U \\ \bar{x}_0 \end{bmatrix} \quad (3a)$$

$$\text{subject to} \quad d + K \bar{x}_0 + M U \leq 0. \quad (3b)$$

The number of inequality constraints is the same as in the original QP (2) and given by $m = N n_h + n_r$. Note that in the simplest case without inequalities ($m = 0$), the solution $U^*(\bar{x}_0)$ of the condensed QP can be obtained by setting the gradient of the objective to zero, i.e., by solving $H U^*(\bar{x}_0) + G \bar{x}_0 = 0$. The factorization of a dense matrix H with dimension $N n_u \times N n_u$ needs $O(N^3 n_u^3)$ arithmetic operations, i.e., the computational cost of condensing-based

algorithms typically grows cubically with the horizon length N .

Banded Matrix Factorization

An alternative way to deal with the sparsity is best sketched at hand of a sparse convex QP (2) without inequality constraints (2d) and (2e). We define the vector of Lagrange multipliers $Y = [y_0^\top, \dots, y_N^\top]^\top$ and the Lagrangian function by

$$\begin{aligned} L(X, U, Y) = & y_0^\top (x_0 - \bar{x}_0) + \frac{1}{2} x_N^\top P x_N \\ & + \frac{1}{2} \sum_{i=0}^{N-1} \begin{bmatrix} x_i \\ u_i \end{bmatrix}^\top \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} x_i \\ u_i \end{bmatrix} + y_{i+1}^\top \\ & (x_{i+1} - A x_i + B u_i). \end{aligned} \quad (4)$$

If we reorder all unknowns that enter the Lagrangian and summarize them in the vector

$$W = [y_0^\top, x_0^\top, u_0^\top, y_1^\top, x_1^\top, u_1^\top, \dots, y_N^\top, x_N^\top]^\top$$

the optimal solution $W^*(\bar{x}_0)$ is uniquely characterized by the first-order optimality condition

$$\nabla_W L(W^*) = 0.$$

Due to the linear quadratic dependence of L on W , this is a block-banded linear equation in the unknown W^* :

$$\begin{bmatrix} 0 & I & & & & & & & \\ I & Q & S & -A^\top & & & & & \\ & S^\top & R & -B^\top & & & & & \\ & -A & -B & 0 & I & & & & \\ & & & I & Q & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \cdot & & \\ & & & & \cdot & \cdot & 0 & I & \\ & & & & & & I & P & \end{bmatrix} W^* = \begin{bmatrix} \bar{x}_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}.$$

Because the above matrix has nonzero elements only on a band of width $2n_x + n_u$ around the diagonal, it can be factorized with a computational cost of order $O(N(2n_x + n_u)^3)$.

Treatment of Inequalities

An important observation is that both the unconstrained QP (2) and the equivalent condensed

QP (3) typically fall into the class of strictly convex parametric quadratic programs: the solution $U^*(\bar{x}_0)$, $X^*(\bar{x}_0)$ is unique and depends piecewise affinely and continuously on the parameter $\bar{x}_0$. Each affine piece of the solution map corresponds to one *active set* and is valid on one polyhedral *critical region* in parameter space. This observation forms the basis of *explicit MPC algorithms* which precompute the map $u_0^*(\bar{x}_0)$, but it can also be exploited in online algorithms for quadratic programming, which are the focus of this section.

We sketch the different ways of how to treat inequalities only for the condensed QP (3), but they can equally be applied in sparse QP algorithms that directly address (2). The optimal solution $U^*(\bar{x}_0)$ for a strictly convex QP (3) is – together with the corresponding vector of Lagrange multipliers, or dual variables $\lambda^*(\bar{x}_0) \in \mathbb{R}^m$ – uniquely characterized by the so-called Karush-Kuhn-Tucker (KKT) conditions, which we omit for brevity. There are three big families of solution algorithms for inequality-constrained QPs that differ in the way they treat inequalities: *active set methods*, *interior point methods*, and *gradient projection methods*.

Active Set Methods

The optimal solution of the QP is characterized by its *active set*, i.e., the set of inequality constraints (3b) that are satisfied with equality at this point. If one would know the active set for a given problem instance $\bar{x}_0$, it would be easy to find the solution. Active set methods work with guesses of the active set which they iteratively refine. In each iteration, they solve one linear system corresponding to a given guess of the active set. If the KKT conditions are satisfied, the optimal solution is found; if they are not, another division into active and inactive constraints needs to be tried. A crucial observation is that an existing factorization of the linear system can be reused to a large extent when only one constraint is added or removed from the guess of the active set. Many different active set strategies exist, three of which we mention: *Primal active set methods* first find a feasible point and then add or remove active constraints, always keeping the primal variables U feasible. Due to the difficulty of finding a

feasible point first, they are difficult to warmstart in the context of MPC optimization. *Dual active set strategies* always keep the dual variables λ positive. They can easily be warmstarted in the context of MPC. *Parametric* or *online active set strategies* ensure that all iterates stay primal and dual feasible and go on a straight line in parameter space from a solved QP problem to the current one, only updating the active set when crossing the boundary between critical regions, as implemented in the online QP solver qpOASES.

Active set methods are very competitive in practice, but their worst case complexity is not polynomial. They are often used together with the condensed QP formulation, for which each active set change is relatively cheap. This is particularly advantageous if an existing factorization can be kept between subsequent MPC optimization problems.

Interior Point Methods

Another approach is to replace the KKT conditions by a smooth approximation that uses a small positive parameter $\tau > 0$:

$$HU^* + G\bar{x}_0 + M^\top \lambda^* = 0, \quad (5a)$$

$$(d + K\bar{x}_0 + MU^*)_i \lambda_i^* + \tau = 0, \quad i = 1, \dots, m. \quad (5b)$$

These conditions form a smooth nonlinear system of equations that uniquely determines a primal dual solution $U^*(\bar{x}_0, \tau)$ and $\lambda^*(\bar{x}_0, \tau)$ in the interior of the feasible set. They are not equivalent to the KKT conditions, but for $\tau \rightarrow 0$, their solution tends to the exact QP solution. An interior point algorithm solves the system (5a) and (5b) by Newton's method. Simultaneously, the path parameter τ , that was initially set to a large value, is iteratively reduced, making the nonlinear set of equations a closer approximation of the original KKT system. In each Newton iteration, a linear system needs to be factored and solved, which constitutes the major computational cost of an interior point algorithm. For the condensed QP (3) with dense matrices H, M , the cost per Newton iteration is of order $O(N^3)$. But the interior point algorithm can also

be applied to the uncondensed sparse QP (2), in which case each iteration has a runtime of order $O(N)$. In practice, for both cases, 10–30 Newton iterations usually suffice to obtain very accurate solutions. As an interior point method needs always to start with a high value of τ and then reduces it during the iterations, warmstarting is of minor benefit. There exist efficient code generation tools that export convex interior point solvers as plain C-code such as CVXGEN and FORCES.

Gradient Projection Methods

Gradient projection methods do not need to factorize any matrix but only evaluate the gradient of the objective function $HU^{[k]} + G\bar{x}_0$ in each iteration. They can only be implemented efficiently if the feasible set is a simple set in the sense that a projection $P(U)$ on this set is very cheap to compute, as, e.g., for upper and lower bounds on the variables U , and if we know an upper bound $L_H > 0$ on the eigenvalues of the Hessian H . The simple gradient projection algorithm starts with an initialization $U^{[0]}$ and proceeds as follows:

$$U^{[k+1]} = P \left(U^{[k]} - \frac{1}{L_H} (HU^{[k]} + G\bar{x}_0) \right).$$

An improved version of the gradient projection algorithm is called the *optimal* or *fast gradient method* and has probably the best possible iteration complexity of all gradient type methods. All variants of gradient projection algorithms are easy to warmstart. Though they are not as versatile as active set or interior point methods, they have short code sizes and can offer advantages on embedded computational hardware, such as the code generated by the tool FIORDOS.

Optimization Algorithms for Nonlinear MPC

When the dynamic system $x_{i+1} = f(x_i, u_i)$ is not affine, the optimal control problem (1) is non-convex, and we speak of a nonlinear

MPC (NMPC) problem. NMPC optimization algorithms only aim at finding a locally optimal solution of this problem, and they usually do it in a Newton-type framework. For ease of notation, we summarize problem (1) in the form of a general nonlinear programming problem (NLP):

$$\begin{aligned} &\text{minimize} && \Phi(X, U) && (6a) \\ &&& X, U \end{aligned}$$

$$\text{subject to} \quad G_{\text{eq}}(X, U, \bar{x}_0) = 0, \quad (6b)$$

$$G_{\text{ineq}}(X, U) \leq 0. \quad (6c)$$

Let us first discuss a fundamental choice that regards the problem formulation and number of optimization variables.

Simultaneous vs. Sequential Formulation

When an optimization algorithm addresses problem (6) iteratively, it works intermediately with nonphysical, infeasible trajectories that violate the system constraints (6b). Only at the optimal solution the constraint residual is brought to zero and a physical simulation is achieved. We speak of a *simultaneous approach* to optimal control because the algorithm solves the simulation and the optimization problems simultaneously. Variants of this approach are *direct discretization* or *direct multiple shooting*.

On the other hand, the equality constraints (6b) could easily be eliminated by a nonlinear forward simulation for given initial value $\bar{x}_0$ and control trajectory U , similar to condensing in linear MPC. Such a forward simulation generates a state trajectory $X_{\text{sim}}(\bar{x}_0, U)$ such that for the given value of $\bar{x}_0$, U the equality constraints (6b) are automatically satisfied: $G_{\text{eq}}(X_{\text{sim}}(\bar{x}_0, U), U, \bar{x}_0) = 0$. Inserting this map into the NLP (6) allows us to formulate an equivalent optimization problem with a reduced variable space:

$$\begin{aligned} &\text{minimize} && \Phi(X_{\text{sim}}(\bar{x}_0, U), U) && (7a) \\ &&& U \end{aligned}$$

$$\text{subject to} \quad G_{\text{ineq}}(X_{\text{sim}}(\bar{x}_0, U), U) \leq 0. \quad (7b)$$

When solving this reduced problem with an iterative optimization algorithm, we sequentially simulate and optimize the system, and we speak of the *sequential approach* to optimal control.

The sequential approach has a lower dimensional variable space and is thus easier to use with a black-box NLP solver. On the other hand, the simultaneous approach leads to a sparse NLP and is better able to deal with unstable nonlinear systems. In the remainder of this section, we thus only discuss the specific structure of the simultaneous approach.

Newton-Type Optimization

In order to simplify notation further, we summarize and reorder all optimization variables U and X in a vector $V = [x_0^\top, u_0^\top, \dots, x_{N-1}^\top, u_{N-1}^\top, x_N^\top]^\top$ and use the same problem function names as in (6) also with the new argument V .

As in the section on linear MPC, we can introduce multipliers Y for the equalities and λ for the inequalities and define the Lagrangian

$$L(V, Y, \lambda) = \Phi(V) + Y^\top G_{\text{eq}}(V, \bar{x}_0) + \lambda^\top G_{\text{ineq}}(V). \quad (8)$$

All Newton-type optimization methods try to find a point satisfying the KKT conditions by using successive linearizations of the problem functions and Lagrangian. For this aim, starting with an initial guess $(V^{[0]}, Y^{[0]}, \lambda^{[0]})$, they generate sequences of primal-dual iterates $(V^{[k]}, Y^{[k]}, \lambda^{[k]})$.

An observation that is crucial for the efficiency of all NMPC algorithms is that the Hessian of the Lagrangian is at a current iterate given by a matrix of the form

$$\nabla_V^2 L(\cdot) = \begin{bmatrix} Q_0^{[k]} & S_0^{[k]} & & & \\ S_0^{[k],\top} & R_0^{[k]} & & & \\ & & Q_1^{[k]} & S_1^{[k]} & \\ & & S_1^{[k],\top} & R_1^{[k]} & \\ & & & & \ddots \\ & & & & & P^{[k]} \end{bmatrix}.$$

This block sparse matrix structure makes it possible to use in each Newton-type iteration the same sparsity-exploiting linear algebra techniques as outlined in section “[Convex Optimization for Linear MPC](#)” for the linear MPC problem.

Major differences exist on how to treat the inequality constraints, and the two big families of Newton-type optimization methods are *sequential quadratic programming (SQP)* methods and *nonlinear interior point (NIP)* methods.

Sequential Quadratic Programming (SQP)

A first variant to iteratively solve the KKT system is to linearize all nonlinear functions at the current iterate $(V^{[k]}, Y^{[k]}, \lambda^{[k]})$ and to find a new solution guess from the solution of a quadratic program (QP):

$$\underset{V}{\text{minimize}} \quad \Phi_{\text{quad}}(V; V^{[k]}, Y^{[k]}, \lambda^{[k]}) \quad (9a)$$

$$\text{subject to} \quad G_{\text{eq,lin}}(V, \bar{x}_0; V^{[k]}) = 0, \quad (9b)$$

$$G_{\text{ineq,lin}}(V; V^{[k]}) \leq 0. \quad (9c)$$

Here, the subindex “lin” in the constraints $G_{\cdot,\text{lin}}$ expresses that a first-order Taylor expansion at $V^{[k]}$ is used, while the QP objective is given by $\Phi_{\text{quad}}(V; V^{[k]}, Y^{[k]}, \lambda^{[k]}) = \Phi_{\text{lin}}(V; V^{[k]}) + \frac{1}{2}(V - V^{[k]})^\top \nabla_V^2 L(\cdot)(V - V^{[k]})$. Note that the QP has the same sparsity structure as the QP (2) resulting from linear MPC, with the only difference that all matrices are now time varying over the MPC horizon. In the case that the Hessian matrix is positive semi-definite, this QP is convex so that global solutions can be found reliably with any of the methods from section “[Convex Optimization for Linear MPC](#).” The solution of the QP along with the corresponding constraint multipliers gives the next SQP iterate $(V^{[k+1]}, Y^{[k+1]}, \lambda^{[k+1]})$. Apart from the presented “exact Hessian” SQP variant, which has quadratic convergence speed, several other SQP variants exist, which make use of other Hessian approximations. A particularly useful Hessian approximation for NMPC is possible if the original objective function $\Phi(V)$ is convex quadratic, and the resulting SQP variant is called

the *generalized Gauss-Newton* method. In this case, one can just use the original objective as cost function in the QP (9a), resulting in convex QP subproblems and (often fast) linear convergence speed.

Nonlinear Interior Point (NIP) Method

In contrast to SQP methods, an alternative way to address the solution of the KKT system is to replace the last nonsmooth KKT conditions by a smooth nonlinear approximation, with $\tau > 0$:

$$\nabla_V L(V^*, Y^*, \lambda^*) = 0 \quad (10a)$$

$$G_{\text{eq}}(V^*, \bar{x}_0) = 0 \quad (10b)$$

$$G_{\text{ineq},i}(V^*) \lambda_i^* + \tau = 0, \quad i = 1, \dots, m. \quad (10c)$$

We summarize all variables in a vector $W = [V^\top, Y^\top, \lambda^\top]^\top$ and summarize the above set of equations as

$$G_{\text{NIP}}(W, \bar{x}_0, \tau) = 0. \quad (11)$$

The resulting root finding problem is then solved with Newton's method, for a descending sequence of path parameters $\tau^{[k]}$. The NIP method proceeds thus exactly as in an interior point method for convex problems, with the only difference that it has to re-linearize all problem functions in each iteration. An excellent software implementation of the NIP method is given in the form of the code IPOPT.

Continuation Methods and Tangential Predictors

In nonlinear MPC, a sequence of OCPs with different initial values $\bar{x}_0^{[0]}, \bar{x}_0^{[1]}, \bar{x}_0^{[2]}, \dots$ is solved. For the transition from one problem to the next, it is beneficial to take into account the fact that the optimal solution $W^*(\bar{x}_0)$ depends almost everywhere differentiably on $\bar{x}_0$. The concept of a continuation method is most easily explained in the context of an NIP method with fixed path parameter $\bar{\tau} > 0$. In this case, the solution $W^*(\bar{x}_0, \bar{\tau})$ of the smooth root finding problem $G_{\text{NIP}}(W^*(\bar{x}_0, \bar{\tau}), \bar{x}_0, \bar{\tau}) = 0$ from Eq. (11) is

smooth with respect to $\bar{x}_0$. This smoothness can be exploited by making use of a *tangential predictor* in the transition from one value of $\bar{x}_0$ to another. Unfortunately, the interior point solution manifold is strongly nonlinear at points where the active set changes, and the tangential predictor is not a good approximation when we linearize at such points.

Generalized Tangential Predictor and Real-Time Iterations

In fact, the true NLP solution is not determined by a smooth root finding problem (10a)–(10c) but by the (nonsmooth) KKT conditions. The solution manifold has smooth parts when the active set does not change, but non-differentiable points occur whenever the active set changes. We can deal with this fact naturally in an SQP framework by solving one QP of form (9) in order to generate a tangential predictor that is also valid in the presence of active set changes. In the extreme case that only one such QP is solved per sampling time, we speak of a *real-time iteration (RTI)* algorithm. The computations in each iteration can be subdivided into two phases, the *preparation phase*, in which the derivatives are computed and the QP is condensed, and the *feedback phase*, which only starts once $\bar{x}_0^{[k+1]}$ becomes available and in which only a condensed QP of form (3) is solved, minimizing the feedback delay. This NMPC algorithm can be generated as plain C-code, e.g., by the tool ACADO. Another class of real-time NMPC algorithms based on a continuation method can be generated by the tool AutoGenU.

Cross-References

- [Explicit Model Predictive Control](#)
- [Model Predictive Control in Practice](#)
- [Numerical Methods for Nonlinear Optimal Control Problems](#)

Recommended Reading

Many of the algorithmic ideas presented in this article can be used in different combinations than those presented, and several

other ideas had to be omitted for the sake of brevity. Some more details can be found in the following two overview articles on MPC optimization: Binder et al. (2001) and Diehl et al. (2009). The general field of numerical optimal control is treated in Bryson and Ho (1975), Betts (2010), and the even broader field of numerical optimization is covered in the excellent textbooks (Fletcher 1987; Wright 1997; Nesterov 2004; Gill et al. 1999; Nocedal and Wright 2006; Biegler 2010). General purpose open-source software for MPC and NMPC is described in the following papers: FORCES (Domahidi et al. 2012), CVXGEN (Mattingley and Boyd 2009), qpOASES (Ferreau et al. 2008), FiOr-dOs (Richter et al. 2011), AutoGenU (Ohtsuka and Kodama 2002), ACADO (Houska et al. 2011), and IPOPT (Wächter and Biegler 2006).

Bibliography

- Betts JT (2010) Practical methods for optimal control and estimation using nonlinear programming, 2nd edn. SIAM, Philadelphia
- Biegler LT (2010) Nonlinear programming. SIAM, Philadelphia
- Binder T, Blank L, Bock HG, Bulirsch R, Dahmen W, Diehl M, Kronseder T, Marquardt W, Schlöder JP, Stryk OV (2001) Introduction to model based optimization of chemical processes on moving horizons. In: Grötschel M, Krumke SO, Rambau J (eds) Online optimization of large scale systems: state of the art. Springer, Berlin, pp 295–340
- Bryson AE, Ho Y-C (1975) Applied optimal control. Wiley, New York
- Diehl M, Ferreau HJ, Haverbeke N (2009) Efficient numerical methods for nonlinear MPC and moving horizon estimation. In: Nonlinear model predictive control. Lecture notes in control and information sciences, vol 384. Springer, Berlin, pp 391–417
- Domahidi A, Zraggen A, Zeilinger MN, Morari M, Jones CN (2012) Efficient interior point methods for multi-stage problems arising in receding horizon control. In: IEEE conference on decision and control (CDC), Maui, Dec 2012, pp 668–674
- Ferreau HJ, Bock HG, Diehl M (2008) An online active set strategy to overcome the limitations of explicit MPC. *Int J Robust Nonlinear Control* 18(8): 816–830
- Fletcher R (1987) Practical methods of optimization, 2nd edn. Wiley, Chichester
- Gill PE, Murray W, Wright MH (1999) Practical optimization. Academic, London
- Houska B, Ferreau HJ, Diehl M (2011) An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica* 47(10): 2279–2285
- Mattingley J, Boyd S (2009) Automatic code generation for real-time convex optimization. In: Convex optimization in signal processing and communications. Cambridge University Press, New York, pp 1–43
- Nesterov Y (2004) Introductory lectures on convex optimization: a basic course. Applied optimization, vol 87. Kluwer, Boston
- Nocedal J, Wright SJ (2006) Numerical optimization. Springer series in operations research and financial engineering, 2nd edn. Springer, New York
- Ohtsuka T, Kodama A (2002) Automatic code generation system for nonlinear receding horizon control. *Trans Soc Instrum Control Eng* 38(7): 617–623
- Richter S, Morari M, Jones CN (2011) Towards computational complexity certification for constrained MPC based on Lagrange relaxation and the fast gradient method. In: 50th IEEE conference on decision and control and European control conference (CDC-ECC), Orlando, Dec 2011, pp 5223–5229
- Wächter A, Biegler LT (2006) On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Math Program* 106(1):25–57
- Wright SJ (1997) Primal-dual interior-point methods. SIAM, Philadelphia

Optimization-Based Control Design Techniques and Tools

Pierre Apkarian¹ and Dominikus Noll²

¹Control Department, DCSD, ONERA – The French Aerospace Lab, Toulouse, France

²Institut de Mathématiques, Université de Toulouse, Toulouse, France

Abstract

Structured output feedback controller synthesis is an exciting recent concept in modern control design, which bridges between theory and practice in so far as it allows for the first time to apply sophisticated mathematical

design paradigms like H_∞ - or H_2 -control within control architectures preferred by practitioners. The new approach to structured H_∞ -control, developed by the authors during the past decade, is rooted in a change of paradigm in the synthesis algorithms. Structured design is no longer based on solving algebraic Riccati equations or matrix inequalities. Instead, optimization-based design techniques are required. In this essay we indicate why structured controller synthesis is central in modern control engineering. We explain why non-smooth optimization techniques are needed to compute structured control laws, and we point to software tools which enable practitioners to use these new tools in high-technology applications.

Keywords

Controller tuning · H_∞ synthesis ·
Multi-objective design · Nonsmooth
optimization · Structured controllers ·
Robust control

Introduction

In the modern high-technology field, control engineers usually face a large variety of concurring design specifications such as noise or gain attenuation in prescribed frequency bands, damping, decoupling, constraints on settling time or risetime, and much else. In addition, as plant models are generally only approximations of the true system dynamics, control laws have to be robust with respect to uncertainty in physical parameters or with regard to un-modeled high-frequency phenomena. Not surprisingly, such a plethora of constraints presents a major challenge for controller tuning, due not only to the ever growing number of such constraints but also because of their very different provenience.

The steady increase in plant complexity is exacerbated by the quest that regulators should be as simple as possible, easy to understand and to tune by practitioners, convenient to hardware

implement, and generally available at low cost. These practical constraints highlight the limited use of Riccati- or LMI-based controllers, and they are the driving force for the implementation of *structured* control architectures. On the other hand, this means that hand-tuning methods have to be replaced by rigorous algorithmic optimization tools.

Structured Controllers

Before addressing specific optimization techniques, we recall some basic terminology for control design problems with structured controllers. The plant model P is described as

$$P : \begin{cases} \dot{x}_P = A x_P + B_1 w + B_2 u \\ z = C_1 x_P + D_{11} w + D_{12} u \\ y = C_2 x_P + D_{21} w + D_{22} u \end{cases} \quad (1)$$

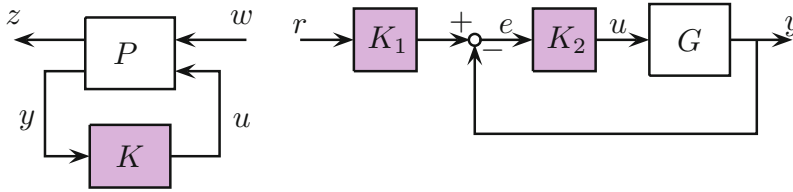
where $A, B_1, \dots$ are real matrices of appropriate dimensions, $x_P \in \mathbb{R}^{n_P}$ is the state, $u \in \mathbb{R}^{n_u}$ the control, $y \in \mathbb{R}^{n_y}$ the measured output, $w \in \mathbb{R}^{n_w}$ the exogenous input, and $z \in \mathbb{R}^{n_z}$ the regulated output. Similarly, the sought output feedback controller K is described as

$$K : \begin{cases} \dot{x}_K = A_K x_K + B_K y \\ u = C_K x_K + D_K y \end{cases} \quad (2)$$

with $x_K \in \mathbb{R}^{n_K}$ and is called *structured* if the (real) matrices A_K, B_K, C_K, D_K depend smoothly on a design parameter $\mathbf{x} \in \mathbb{R}^n$, referred to as the vector of tunable parameters. Formally, we have differentiable mappings

$$\begin{aligned} A_K &= A_K(\mathbf{x}), B_K = B_K(\mathbf{x}), C_K = C_K(\mathbf{x}), \\ D_K &= D_K(\mathbf{x}), \end{aligned}$$

and we abbreviate these by the notation $K(\mathbf{x})$ for short to emphasize that the controller is structured with $\mathbf{x}$ as tunable elements. A structured controller synthesis problem is then an optimization problem of the form



Optimization-Based Control Design Techniques and Tools, Fig. 1 Black-box full-order controller K on the left, structured 2-DOF control architecture with $K = \text{diag}(K_1, K_2)$ on the right

$$\begin{aligned} & \text{minimize } \|T_{wz}(P, K(\mathbf{x}))\| \\ & \text{subject to } K(\mathbf{x}) \text{ closed-loop stabilizing} \quad (3) \\ & K(\mathbf{x}) \text{ structured, } \mathbf{x} \in \mathbb{R}^n \end{aligned}$$

where $T_{wz}(P, K) = \mathcal{F}_\ell(P, K)$ is the lower feed-back connection of (1) with (2) as in Fig. 1 (left), also called the linear fractional transformation (Zhou et al. 1996). The norm $\|\cdot\|$ stands for the H_∞ -norm, the H_2 -norm, or any other system norm, while the optimization variable $\mathbf{x} \in \mathbb{R}^n$ regroups the tunable parameters in the design.

Standard examples of structured controllers $K(\mathbf{x})$ include realizable PIDs, observer-based, reduced-order, or decentralized controllers, which in state space are expressed as:

$$\begin{aligned} & \left[\begin{array}{cc|c} 0 & 0 & 1 \\ 0 & -1/\tau & -k_D/\tau \\ \hline k_I & 1/\tau & k_P + k_D/\tau \end{array} \right], \\ & \left[\begin{array}{c|c} A - B_2K_c - K_fC_2 & K_f \\ \hline -K_c & 0 \end{array} \right], \quad \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array} \right], \\ & \left[\begin{array}{c|c} \text{diag}(A_{K_1}, \dots, A_{K_q}) & \text{diag}(B_{K_1}, \dots, B_{K_q}) \\ \hline \text{diag}(C_{K_1}, \dots, C_{K_q}) & \text{diag}(D_{K_1}, \dots, D_{K_q}) \end{array} \right]. \end{aligned}$$

In the case of a PID, the tunable parameters are $\mathbf{x} = (\tau, k_P, k_I, k_D)$; for observer-based controllers, $\mathbf{x}$ regroups the estimator and state-feedback gains (K_f, K_c); for reduced order controllers $n_K < n_P$, the tunable parameters $\mathbf{x}$ are the $n_K^2 + n_K n_y + n_K n_u + n_y n_u$ unknown entries in (A_K, B_K, C_K, D_K) ; and in the decentralized form, $\mathbf{x}$ regroups the unknown entries in $A_{K_1}, \dots, D_{K_q}$. In contrast, full-order controllers have the maximum number $N = n_P^2 + n_P n_y + n_P n_u + n_y n_u$ of degrees

of freedom and are referred to as unstructured or as *black-box* controllers.

More sophisticated controller structures $K(\mathbf{x})$ arise from architectures like a 2-DOF control arrangement with feedback block K_2 and a set-point filter K_1 as in Fig. 1 (right). Suppose K_1 is the 1st-order filter $K_1(s) = a/(s + a)$ and K_2 the PI feedback $K_2(s) = k_P + k_I/s$. Then the transfer T_{ry} from r to y can be represented as the feedback connection of P and $K(\mathbf{x}, s)$ with

$$P = \left[\begin{array}{c|cc} A & 0 & B \\ \hline C & 0 & D \\ 0 & I & 0 \\ \hline -C & I & -D \end{array} \right],$$

$$K(\mathbf{x}, s) = \begin{bmatrix} K_1(a, s) & K_2(k_P, k_I, s) \end{bmatrix},$$

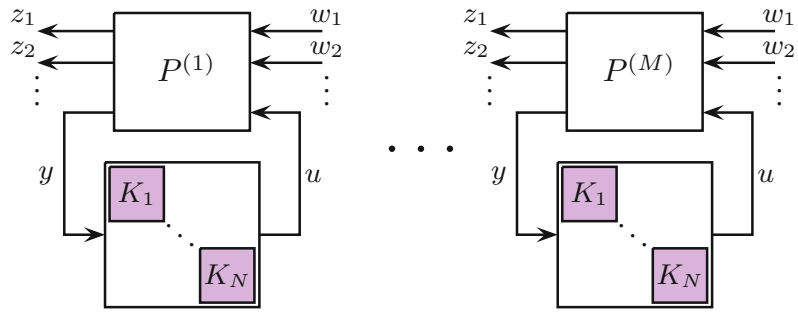
and gathering tunable elements in $\mathbf{x} = (a, k_P, k_I)$.

In much the same way, arbitrary multi-loop interconnections of fixed-model elements with tunable controller blocks $K_i(\mathbf{x})$ can be rearranged as in Fig. 2, so that $K(\mathbf{x})$ captures all tunable blocks in a decentralized structure general enough to cover most engineering applications.

The structure concept is equally useful to deal with the second central challenge in control design: *system uncertainty*. The latter may be handled with μ -synthesis techniques (Stein and Doyle 1991) if a parametric uncertain model is available. A less ambitious but often more practical alternative consists in optimizing the structured controller $K(\mathbf{x})$ against a finite set of plants $P^{(1)}, \dots, P^{(M)}$ representing model variations due to uncertainty, aging, sensor and actuator breakdown, un-modeled dynamics, in tandem

Optimization-Based Control Design Techniques and Tools,

Fig. 2 Synthesis of $K = \text{diag}(K_1, \dots, K_N)$ against multiple requirements or models $P^{(1)}, \dots, P^{(M)}$. Each $K_i(x)$ can be structured



with the robustness and performance specifications. This is again formally covered by Fig. 2 and leads to a multi-objective constrained optimization problem of the form

$$\begin{aligned} \text{minimize } f(\mathbf{x}) &= \max_{k \in \text{SOFT}, i \in I_k} \|T_{w_i z_i}^{(k)}(K(\mathbf{x}))\| \\ \text{subject to } g(\mathbf{x}) &= \max_{k \in \text{HARD}, j \in J_k} \|T_{w_j z_j}^{(k)}(K(\mathbf{x}))\| \leq 1 \\ &K(\mathbf{x}) \text{ structured and stabilizing} \\ &\mathbf{x} \in \mathbb{R}^n \end{aligned} \quad (4)$$

where $T_{w_i z_i}^{(k)}$ denotes the i th closed-loop robustness or performance channel $w_i \rightarrow z_i$ for the k -th plant model $P^{(k)}$. SOFT and HARD denote index sets taken over a finite set of specifications, say in $\{1, \dots, M\}$. The rationale of (4) is to minimize the worst-case cost of the soft constraints $\|T_{w_i z_i}^{(k)}\|$, $k \in \text{SOFT}$, while enforcing the hard constraints $\|T_{w_j z_j}^{(k)}\| \leq 1$, $k \in \text{HARD}$, which prevail over soft ones and are mandatory. In addition to local optimization (4), the problem can undergo a global optimization step in order to prove global stability and performance of the design, see Ravanbod et al. (2017) and Apkarian et al. (2015a, b).

Optimization Techniques Over the Years

During the late 1990s, the necessity to develop design techniques for structured regulators $K(\mathbf{x})$ was recognized (Fares et al. 2001), and the limitations of synthesis methods based on algebraic Riccati equations or linear matrix inequalities

(LMIs) became evident, as these techniques cannot provide structured controllers needed in practice. The lack of appropriate synthesis techniques for structured $K(\mathbf{x})$ led to the unfortunate situation, where sophisticated approaches like the H_∞ paradigm developed by academia since the 1980s could not be brought to work for the design of those controller structures $K(\mathbf{x})$ preferred by practitioners. Design engineers had to continue to rely on heuristic and ad hoc tuning techniques, with only limited scope and reliability. As an example, post-processing to reduce a black-box controller to a practical size is prone to failure. It may at best be considered a fill-in for a rigorous design method which directly computes a reduced-order controller. Similarly, hand-tuning of the parameters $\mathbf{x}$ remains a puzzling task because of the loop interactions and fails as soon as complexity increases.

In the late 1990s and early 2000s, a change of methods was observed. Structured H_2 - and H_∞ -synthesis problems (3) were addressed by bilinear matrix inequality (BMI) optimization, which used local optimization techniques based on the augmented Lagrangian method (Fares et al. 2001; Noll et al. 2004; Kocvara and Stingl 2003; Noll 2007), sequential semidefinite programming methods (Fares et al. 2002; Apkarian et al. 2003), and non-smooth methods for BMIs (Noll et al. 2009; Lemaréchal and Oustry 2000). However, these techniques were based on the bounded real lemma or similar matrix inequalities and were therefore of limited success due to the presence of Lyapunov variables, i.e., matrix-valued unknowns, whose dimension grows quadratically in $n_P + n_K$ and represents the bottleneck of that approach.

The epoch-making change occurs with the introduction of non-smooth optimization techniques (Noll and Apkarian 2005; Apkarian and Noll 2006b,c, 2007) to programs (3) and (4). Today non-smooth methods have superseded matrix inequality-based techniques and may be considered the state of art as far as realistic applications are concerned. The transition took almost a decade.

Alternative control-related local optimization techniques and heuristics include the gradient sampling technique of Burke et al. (2005), and other derivative-free optimization techniques as in Kolda et al. (2003) and Apkarian and Noll (2006a), particle swarm optimization, see Oi et al. (2008) and references therein, and also evolutionary computation techniques (Lieslehto 2001). All these classes do not exploit derivative information and rely on function evaluations only. They are therefore applicable to a broad variety of problems including those where function values arise from complex numerical simulations. The combinatorial nature of these techniques, however, limits their use to small problems with a few tens of variable. More significantly, these methods often lack a solid convergence theory. In contrast, as we have demonstrated over recent years (Apkarian and Noll 2006b; Noll et al. 2008; Apkarian et al. 2016, 2018), specialized non-smooth techniques are highly efficient in practice, are based on a sophisticated convergence theory, are capable of solving medium-size problems in a matter of seconds, and are still operational for large-size problems with several hundreds of states.

Non-smooth Optimization Techniques

The benefit of the non-smooth casts (3) and (4) lies in the possibility to avoid searching for Lyapunov variables, a major advantage as their number $(n_P + n_K)^2/2$ usually largely dominates n , the number of true decision parameters $\mathbf{x}$. Lyapunov variables do still occur implicitly in the function evaluation procedures, but this has no harmful effect for systems up to several hundred

states. In abstract terms, a non-smooth optimization program has the form

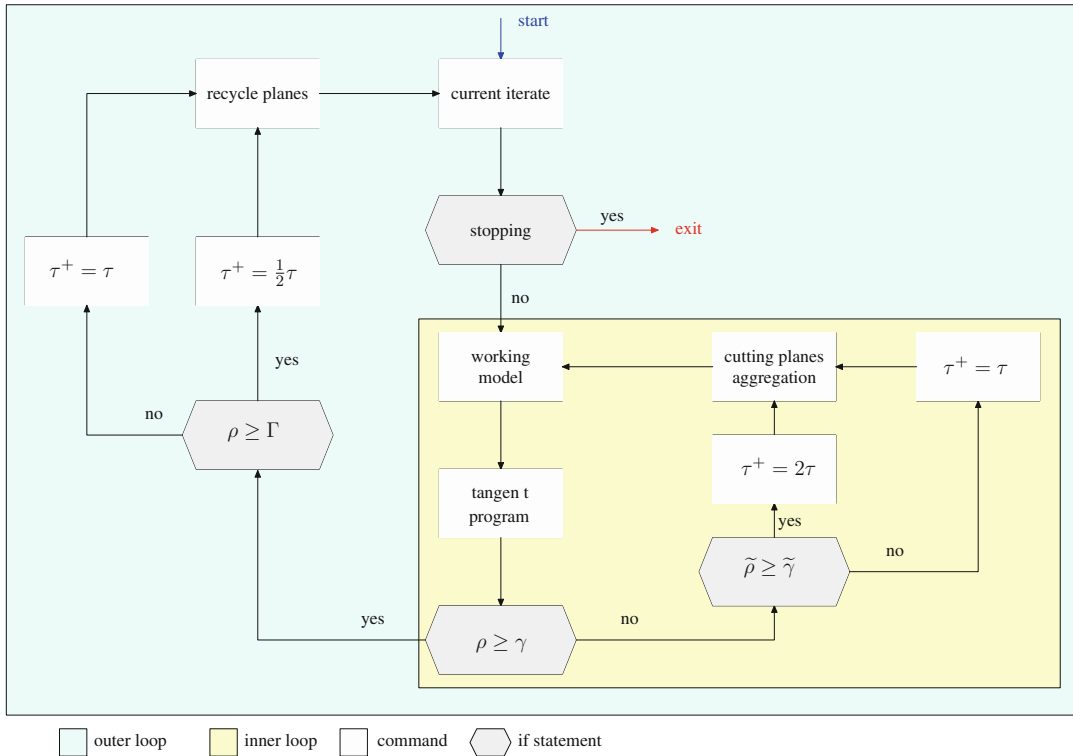
$$\begin{aligned} & \text{minimize } f(\mathbf{x}) \\ & \text{subject to } g(\mathbf{x}) \leq 0 \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \quad (5)$$

where $f, g : \mathbb{R}^n \rightarrow \mathbb{R}$ are locally Lipschitz functions and are easily identified from the cast in (4).

In the realm of convex optimization, non-smooth programs are conveniently addressed by so-called bundle methods, introduced in the late 1970s by Lemaréchal (1975). Bundle methods are used to solve difficult problems in integer programming or in stochastic optimization via Lagrangian relaxation. Extensions of the bundling technique to non-convex problems like (3) or (4) were first developed in Apkarian and Noll (2006b,c, 2007), Apkarian et al. (2008), and Noll et al. (2009) and, in more abstract form, in Noll et al. (2008). Recently, we also extended bundle techniques to the trust-region framework (Apkarian et al. 2016, 2018; Apkarian and Noll 2018), which leads to the first extension of the classical trust-region method to non-differential optimization supported by a valid convergence theory.

Figure 3 shows a schematic view of a non-convex bundle method consisting of a descent-step generating inner loop (yellow block) comparable to a line search in smooth optimization, embedded into the outer loop (blue box), where serious iterates are processed, stopping criteria are applied, and the acceptance rules of traditional trust-region techniques are assured. At the core of the interaction between inner and outer loop is the management of the proximity control parameter τ , which governs the stepsize $\|\mathbf{x} - \mathbf{y}^k\|$ between trial steps $\mathbf{y}^k$ at the current serious iterate $\mathbf{x}$. Similar to the management of a trust-region radius or of the step size in a line search, proximity control allows to force shorter trial steps if agreement of the local model with the true objective function is poor and allows larger steps if agreement is satisfactory.

Oracle-based bundle methods traditionally assure global convergence in the sense of



Optimization-Based Control Design Techniques and Tools, Fig. 3 Flow chart of proximity control bundle algorithm

subsequences under the sole hypothesis that for every trial point x , the function value $f(x)$ and one Clarke subgradient $\phi \in \partial f(x)$ are provided. In automatic control applications, it is as a rule possible to provide more specific information, which may be exploited to speed up convergence (Apkarian and Noll 2006b).

Computing function value and gradients of the H_2 -norm $f(x) = \|T_{wz}(P, K(x))\|_2$ requires essentially the solution of two Lyapunov equations of size $n_P + n_K$; see Apkarian et al. (2007). For the H_∞ -norm, $f(x) = \|T_{wz}(P, K(x))\|_\infty$, function evaluation is based on the Hamiltonian algorithm of Benner et al. (2012) and Boyd et al. (1989). The Hamiltonian matrix is of size $2(n_P + n_K)$, so that function evaluations may be costly for very large plant state dimension ($n_P > 500$), even though the number of outer loop iterations of the bundle algorithm is not affected by a large n_P and generally relates to n , the dimension of x .

The additional cost for subgradient computation for large n_P is relatively cheap as it relies on linear algebra (Apkarian and Noll 2006b). Function and subgradient evaluations for H_∞ and H_2 norms are typically obtained in $\mathcal{O}((n_P + n_K)^3)$ flops.

Computational Tools

Our non-smooth optimization methods became available to the engineering community since 2010 via the MATLAB Robust Control Toolbox (Robust Control Toolbox 4.2 2012; Gahinet and Apkarian 2011). Routines HINFSTRUCT, LOOPTUNE, and SYSTUNE are versatile enough to define and combine tunable blocks $K_i(x)$, to build and aggregate multiple models and design requirements on $T_{wz}^{(k)}$ of different nature, and to provide suitable validation tools. Their

implementation was carried out in cooperation with P. Gahinet (MathWorks). These routines further exploit the structure of problem (4) to enhance efficiency; see Apkarian and Noll (2007) and Apkarian and Noll (2006b).

It should be mentioned that design problems with multiple hard constraints are inherently complex and generally NP-hard, so that exhaustive methods fail even for small- to medium-size problems. The principled decision made in Apkarian and Noll (2006b), and reflected in the MATLAB tools, is to rely on local optimization techniques instead. This leads to weaker convergence certificates but has the advantage to work successfully in practice. In the same vein, in (4) it is preferable to rely on a mixture of soft and hard requirements, for instance, by the use of exact penalty functions (Noll and Apkarian 2005). Key features implemented in the mentioned MATLAB routines are discussed in Apkarian (2013), Gahinet and Apkarian (2011), and Apkarian and Noll (2007).

Applications

Design of a feedback regulator is an interactive process, in which tools like SYSTUNE, LOOPTUNE, or HINFSTRUCT support the designer in various ways. In this section we illustrate their enormous potential by showing that even infinite-dimensional systems may be successfully addressed by these tools. For a plethora of design examples for real-rational systems including parametric and complex dynamic uncertainty, we refer to Ravanbod et al. (2017), Apkarian et al. (2015a, 2016, 2018), and Apkarian and Noll (2018). For recent applications of our tools in real-world applications, see also Falcoz et al. (2015), where it is in particular explained how HINFSTRUCT helped in 2014 to save the Rosetta mission. Another important application of HINFSTRUCT is the design of the atmospheric flight pilot for the ARIANE VI launcher by the ArianeGroup (Ganet-Schoeller et al. 2017).

Illustrative Example

We discuss boundary control of a wave equation with anti-stable damping,

$$\begin{aligned}x_{tt}(\xi, t) &= x_{\xi\xi}(\xi, t), \quad t \geq 0, \xi \in [0, 1] \\x_{\xi}(0, t) &= -qx_t(0, t), \quad q > 0, q \neq 1 \quad (6) \\x_{\xi}(1, t) &= u(t).\end{aligned}$$

where notations x_y and x_{yy} stand for partial first- and second-order derivatives of x with respect to y , respectively. In (6), $x(\cdot, t)$, $x_t(\cdot, t)$ is the state; the control applied at the boundary $\xi = 1$ is $u(t)$, and we assume that the measured outputs are

$$\begin{aligned}y_1(t) &= x(0, t), \quad y_2(t) = x(1, t), \\y_3(t) &= x_t(1, t).\end{aligned} \quad (7)$$

The system has been discussed previously in Smyshlyaev and Krstic (2009), Fridman (2014), and Bresch-Pietri and Krstic (2014) and has been proposed for the control of slip-stick vibrations in drilling devices (Saldivar et al. 2013). Here measurements y_1, y_2 correspond to the angular positions of the drill string at the top and bottom level, and y_3 measures angular speed at the top level, while control corresponds to a reference velocity at the top. The friction characteristics at the bottom level are characterized by the parameter q , and the control objective is to maintain a constant angular velocity at the bottom.

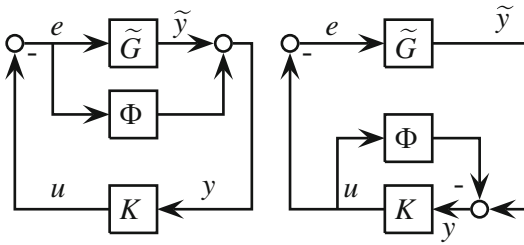
Similar models have been used to control pressure fields in duct combustion dynamics; see DeQueiroz and Rahn (2002). The challenge in (6) and (7) is to design implementable controllers despite the use of an infinite-dimensional system model.

The transfer function of (6) is obtained from

$$G(\xi, s) = \frac{x(\xi, s)}{u(s)} = \frac{1}{s} \cdot \frac{(1-q)e^{s\xi} + (1+q)e^{-s\xi}}{(1-q)e^s - (1+q)e^{-s}},$$

which in view of (7) leads to $G(s)^T = [G_1 \ G_2 \ G_3] = [G(0, s) \ G(1, s) \ sG(1, s)]$.

Putting G in feedback with the controller $K_0 = [0 \ 0 \ 1]$ leads to $\widehat{G} = G/(1 + G_3)$, where



Optimization-Based Control Design Techniques and Tools, Fig. 4 Stability of the closed-loop ($\tilde{G} + \Phi, K$) is equivalent to stability of the closed-loop ($\tilde{G}$, feedback(K, Φ)). See also Moelja and Meinsma (2003)

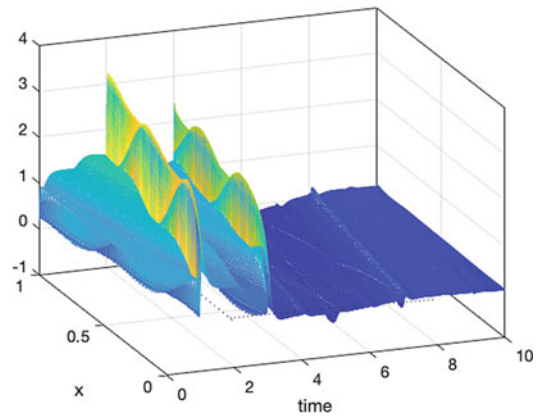
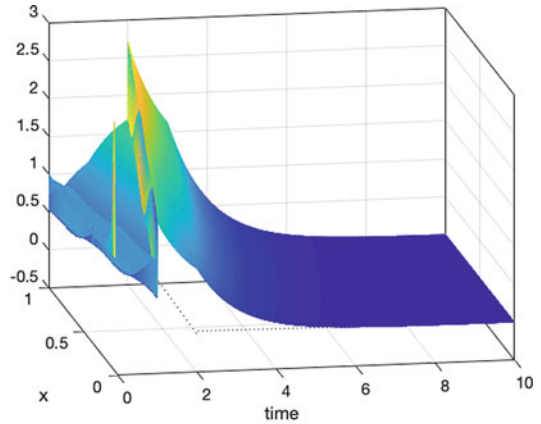
$$\hat{G}(s) = \begin{bmatrix} \frac{1}{s(1-q)} \\ \frac{1+Q}{2s} \\ \frac{1}{2} \end{bmatrix} + \begin{bmatrix} -\frac{1-e^{-s}}{s(1-q)} \\ -\frac{Q(1-e^{-2s})}{2s} \\ \frac{Q}{2}e^{-2s} \end{bmatrix} \\ =: \tilde{G}(s) + \Phi(s), \quad (8)$$

where $Q = (1+q)/(1-q)$.

Here $\tilde{G}$ is real-rational and unstable, while Φ is stable but infinite dimensional. The latter follows from the fact that $\frac{1-e^{-s}}{s}$ and $\frac{1-e^{-2s}}{2s}$ are stable transfert functions as clarified by series expansions. Now we use the fact that stability of the closed loop ($\tilde{G} + \Phi, K$) is equivalent to stability of the loop ($\tilde{G}$, feedback(K, Φ)) upon defining feedback(M, N) := $M(I + NM)^{-1}$. The loop transformation is explained in Fig. 4.

Using (8) we construct a finite-dimensional structured controller $\tilde{K} = \tilde{K}(x)$ which stabilizes $\tilde{G}$. The controller K stabilizing $\hat{G}$ in (8) is then recovered from $\tilde{K}$ through the equation $\tilde{K} = \text{feedback}(K, \Phi)$, which when inverted gives $K = \text{feedback}(\tilde{K}, -\Phi)$. The overall controller for (6) is $K^* = K_0 + K$, and since along with K only delays appear in Φ , the controller K^* is implementable.

Construction of $\tilde{K}$ uses SYSTUNE with pole placement via TuningGoal.Poles, imposing that closed-loop poles have a minimum decay of 0.9, minimum damping of 0.9, and a maximum frequency of 4.0. The controller structure is chosen as static, so that $x \in \mathbb{R}^3$. A simulation with K^* is shown in Fig. 5 (bottom), and some acceleration over the backstepping controller from Bresch-Pietri and Krstic (2014) (top) is observed.



Optimization-Based Control Design Techniques and Tools, Fig. 5 Wave equation. Simulations for K obtained by backstepping control (top) (Bresch-Pietri and Krstic 2014) and $K^* = K_0 + K$ obtained by optimizing feedback($\tilde{G}, \tilde{K}$) via SYSTUNE (bottom). Both controllers are ∞ -dimensional but implementable

Gain-Scheduling Control

Our last study is when the parameter $q \geq 0$ is uncertain or allowed to vary in time with sufficiently slow variations as in Shamma and Athans (1990). We assume that a nominal $q_0 > 0$ and an uncertain interval $[q, \bar{q}]$ with $q_0 \in (q, \bar{q})$ and $1 \notin [q, \bar{q}]$ are given.

The following scheduling scenarios, all leading to implementable controllers, are possible: (a) computing a nominal controller $\tilde{K}$ at q_0 as before, and scheduling through $\Phi(q)$, which depend explicitly on q , so that $K^{(1)}(q) = K_0 + \text{feedback}(\tilde{K}, -\Phi(q))$, and (b) computing $\tilde{K}(q)$ which depends on q , and using $K^{(2)}(q) = K_0 + \text{feedback}(\tilde{K}(q), -\Phi(q))$.

While (a) uses (3) based on Apkarian and Noll (2006b,c) and available in SYSTUNE, we show that one can also apply (3) to case (b). We use Fig. 4 to work in the finite-dimensional system $(\tilde{G}(q), \tilde{K}(q))$, where plant and controller now depend on q , which is a parameter-varying design.

For that we have to decide on a parametric form of the controller $\tilde{K}(q)$, which we choose as

$$\begin{aligned}\tilde{K}(q, \mathbf{x}) = & \tilde{K}(q_0) + (q - q_0)\tilde{K}_1(\mathbf{x}) \\ & + (q - q_0)^2\tilde{K}_2(\mathbf{x}),\end{aligned}$$

and where we adopted the simple static form $\tilde{K}_1(\mathbf{x}) = [\mathbf{x}_1 \ \mathbf{x}_2 \ \mathbf{x}_3]$, $\tilde{K}_2 = [\mathbf{x}_4 \ \mathbf{x}_5 \ \mathbf{x}_6]$, featuring a total of 6 tunable parameters. The nominal $\tilde{K}(q_0)$ is obtained via (3) as above. For $q_0 = 3$ this leads to $\tilde{K}(q_0) = [-1.049 \ -1.049 \ -0.05402]$, computed via SYSTUNE.

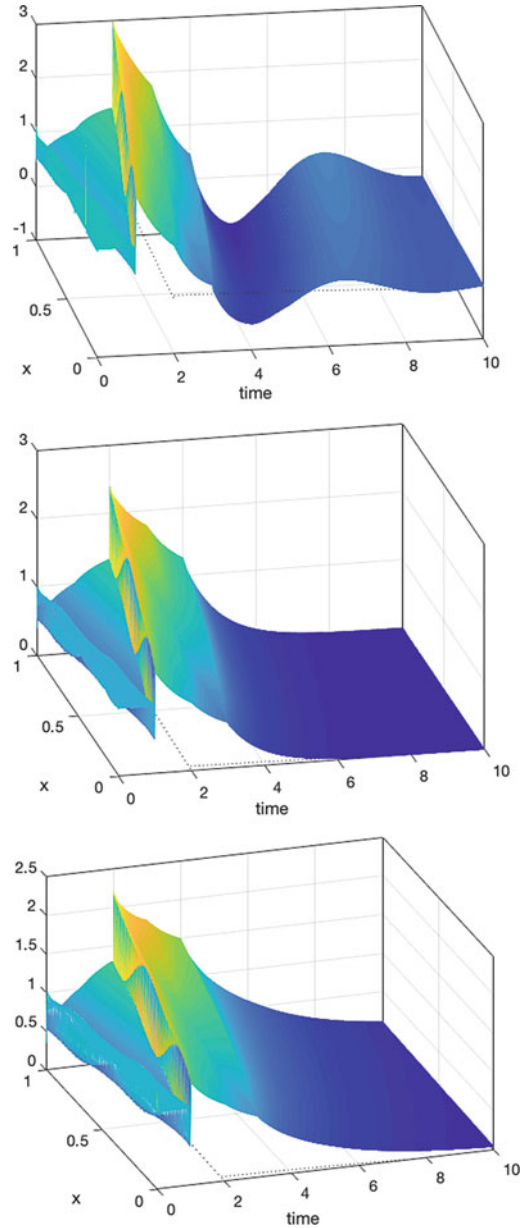
With the parametric form $\tilde{K}(q, \mathbf{x})$ fixed, we now use again the feedback system $(\tilde{G}(q), \tilde{K}(q))$ in Fig. 4 and design a parametric robust controller using the method of Apkarian et al. (2015a), which is included in the SYSTUNE package and used by default if an uncertain closed-loop is entered. The tuning goals are chosen as constraints on closed-loop poles including minimum decay of 0.7, minimum damping of 0.9, with maximum frequency 2. The controller obtained is (with $q_0 = 3$)

$$\begin{aligned}\tilde{K}(q, \mathbf{x}^*) = & \tilde{K}(q_0) + (q - q_0)\tilde{K}_1(\mathbf{x}^*) \\ & + (q - q_0)^2\tilde{K}_2(\mathbf{x}^*),\end{aligned}$$

with $\tilde{K}_1 = [-0.1102, -0.1102, -0.1053]$, $\tilde{K}_2 = [0.03901, 0.03901, 0.02855]$, and we retrieve the final parameter varying controller for $G(q)$ as

$$K^{(2)}(q) = K_0 + \text{feedback}(\tilde{K}(q, \mathbf{x}^*), -\Phi(q)).$$

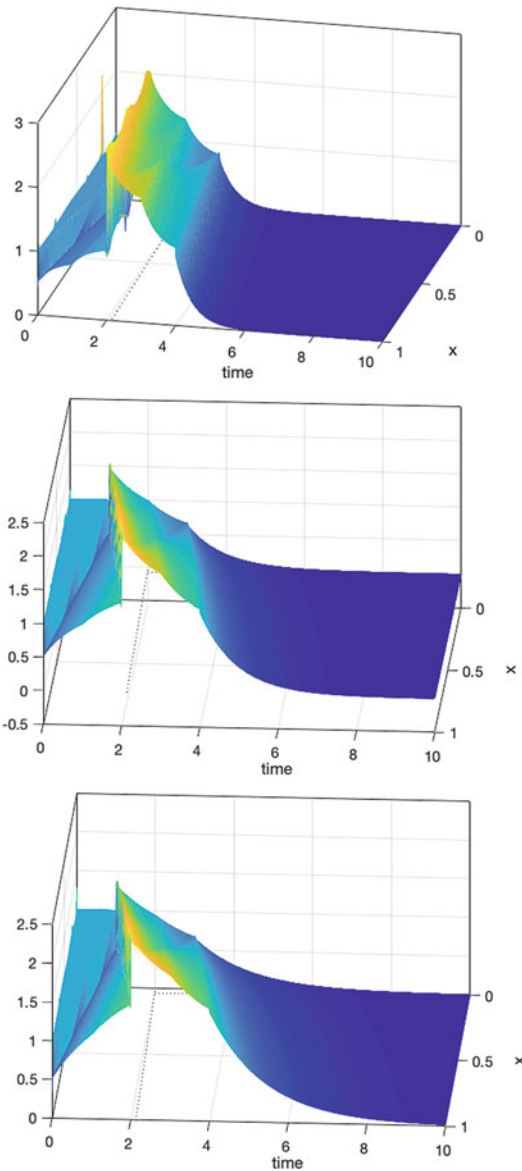
Nominal and scheduled controllers are compared in simulation in Figs. 6, 7, and 8, which indicate that $K^{(2)}(q)$ achieves the best performance for frozen-in-time values $q \in [2, 4]$. All controllers



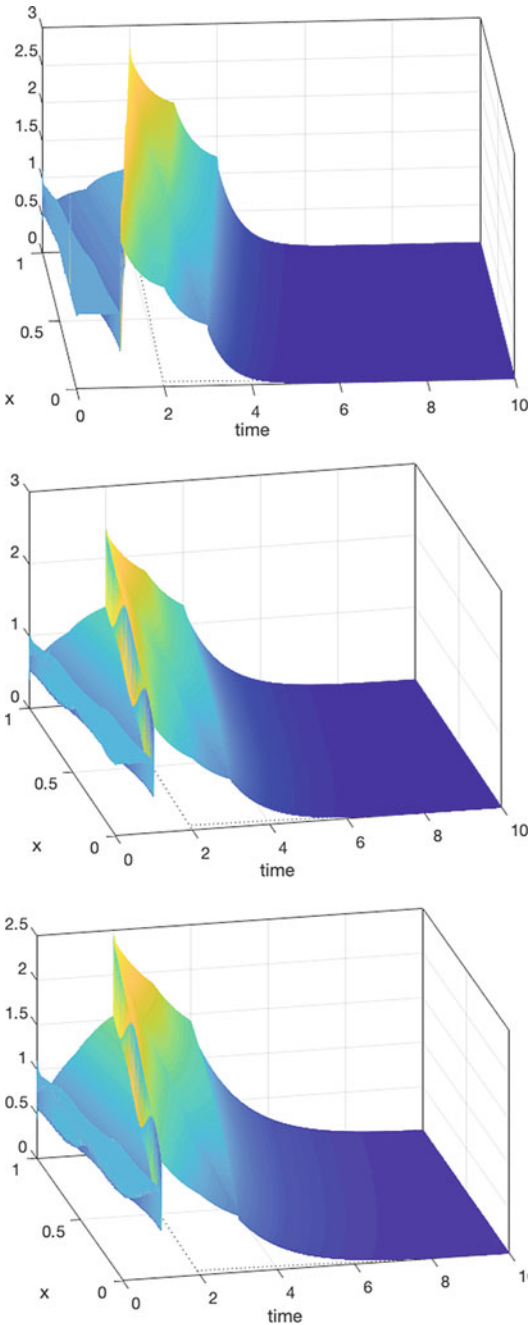
Optimization-Based Control Design Techniques and Tools, Fig. 6 Synthesis at nominal $q_0 = 3$. Simulations of nominal $K = K_0 + \text{feedback}(\tilde{K}, \Phi(3))$ for $q = 2, 3, 4$. Nominal controller is robustly stable over $[q, \bar{q}]$

are easily implementable, since only real-rational elements in combination with delays are used.

The non-smooth program (5) was solved with SYSTUNE in 30 s CPU on a Mac OS X with



Optimization-Based Control Design Techniques and Tools, Fig. 7 Method 1. $\tilde{K}$ obtained for nominal $q = 3$, but scheduled $K(q) = K_0 + \text{feedback}(\tilde{K}, \Phi(q))$. Simulations for $q = 2$ top, $q = 3$ middle, $q = 4$ bottom



Optimization-Based Control Design Techniques and Tools, Fig. 8 Method 2. $\tilde{K}(q) = \tilde{K}_{\text{nom}} + (q - 3)\tilde{K}_1 + (q - 3)^2\tilde{K}_2$ and $K(q) = K_0 + \text{feedback}(\tilde{K}(q), \Phi(q))$. Simulations for $q = 2, 3, 4$

2.66 GHz Intel Core i7 and 8 GB RAM. The reader is referred to the MATLAB Control Toolbox 2018b and higher versions for additional examples. More details on this study can be found in Apkarian and Noll (2019).

Cross-References

- [Multi-objective \$H_2/H_\infty\$ Control Designs](#)
- [Optimization-Based Robust Control](#)
- [Robust Synthesis and Robustness Analysis Techniques and Tools](#)

Bibliography

- Apkarian P (2013) Tuning controllers against multiple design requirements. In: American control conference (ACC), pp 3888–3893
- Apkarian P, Noll D (2006a) Controller design via nonsmooth multi-directional search. *SIAM J Control Optim* 44(6):1923–1949
- Apkarian P, Noll D (2006b) Nonsmooth H_∞ synthesis. *IEEE Trans Aut Control* 51(1):71–86
- Apkarian P, Noll D (2006c) Nonsmooth optimization for multidisk H_∞ synthesis. *Eur J Control* 12(3):229–244
- Apkarian P, Noll D (2007) Nonsmooth optimization for multiband frequency domain control design. *Automatica* 43(4):724–731
- Apkarian P, Noll D (2018) Structured H_∞ -control of infinite dimensional systems. *Int J Robust Nonlinear Control* 28(9):3212–3238
- Apkarian P, Noll D (2019) Boundary control of partial differential equations using frequency domain optimization techniques. [arXiv:1905.06786v1](https://arxiv.org/abs/1905.06786v1), pp 1–22
- Apkarian P, Noll D, Thevenet JB, Tuan HD (2003) A spectral quadratic-SDP method with applications to fixed-order H_2 and H_∞ synthesis. *Eur J Control* 10(6):527–538
- Apkarian P, Noll D, Rondepierre A (2007) Mixed H_2/H_∞ control via nonsmooth optimization. In: Proceedings of the 46th IEEE conference on decision and control, New Orleans, pp 4110–4115
- Apkarian P, Noll D, Prot O (2008) A trust region spectral bundle method for nonconvex eigenvalue optimization. *SIAM J Optim* 19(1):281–306
- Apkarian P, Dao MN, Noll D (2015a) Parametric robust structured control design. *IEEE Trans Auto Control* 60(7):1857–1869
- Apkarian P, Noll D, Ravanbod L (2015b) Computing the structured distance to instability. In: SIAM conference on control and its applications, pp 423–430. <https://doi.org/10.1137/1.9781611974072.58>
- Apkarian P, Noll D, Ravanbod L (2016) Nonsmooth bundle trust-region algorithm with applications to robust stability. *Set-Valued Var Anal* 24(1):115–148
- Apkarian P, Noll D, Ravanbod L (2018) Non-smooth optimization for robust control of infinite-dimensional systems. *Set-Valued Var Anal* 26(2):405–429
- Benner P, Sima V, Voigt M (2012) L_∞ -norm computation for continuous-time descriptor systems using structured matrix pencils. *IEEE Trans Aut Control* 57(1):233–238
- Boyd S, Balakrishnan V, Kabamba P (1989) A bisection method for computing the H_∞ norm of a transfer matrix and related problems. *Math Control Signals Syst* 2(3):207–219
- Bresch-Pietri D, Krstic M (2014) Output-feedback adaptive control of a wave PDE with boundary anti-damping. *Automatica* 50(5):1407–1415
- Burke J, Lewis A, Overton M (2005) A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM J Optim* 15:751–779
- DeQueiroz M, Rahn CD (2002) Boundary control of vibrations and noise in distributed parameter systems: an overview. *Mech Syst Signal Process* 16(1):19–38
- Falcoz A, Pittet C, Bennani S, Guignard A, Bayart C, Frapard B (2015) Systematic design methods of robust and structured controllers for satellites. Application to the refinement of Rosetta’s orbit controller. *CEAS Space J* 7:319–334. <https://doi.org/10.1007/s12567-015-0099-8>
- Fares B, Apkarian P, Noll D (2001) An augmented Lagrangian method for a class of LMI-constrained problems in robust control theory. *Int J Control* 74(4):348–360
- Fares B, Noll D, Apkarian P (2002) Robust control via sequential semidefinite programming. *SIAM J Control Optim* 40(6):1791–1820
- Fridman E (2014) Introduction to time-delay systems. Systems and control, foundations and applications. Birkhuser, Basel
- Gahinet P, Apkarian P (2011) Structured H_∞ synthesis in MATLAB. In: Proceedings of IFAC world congress, Milan, pp 1435–1440
- Ganet-Schoeller M, Desmarioux J, Combier C (2017) Structured control for future european launchers. *AeroSpaceLab* 13:2–10
- Kocvara M, Stingl M (2003) A code for convex nonlinear and semidefinite programming. *Optim Methods Softw* 18(3):317–333
- Kolda TG, Lewis RM, Torczon V (2003) Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Rev* 45(3):385–482
- Lemaréchal C (1975) An extension of Davidson’s methods to nondifferentiable problems. In: Balinski ML, Wolfe P (eds) Nondifferentiable optimization. Mathematical programming study, Springer, Berlin, Heidelberg, vol 3, pp 95–109
- Lemaréchal C, Oustry F (2000) Nonsmooth algorithms to solve semidefinite programs. In: El Ghaoui L, Niculescu S-I (ed) SIAM advances in linear matrix inequality methods in control Series, SIAM, pp 57–77
- Lieslehto J (2001) PID controller tuning using evolutionary programming. In: American control conference, vol 4, pp 2828–2833
- Moelja AA, Meinsma G (2003) Parametrization of stabilizing controllers for systems with multiple i/o delays. *IFAC Proc Vol* 36(19):351–356. 4th IFAC workshop on time delay systems (TDS 2003), Rocquencourt, 8–10 Sept 2003
- Noll D (2007) Local convergence of an augmented Lagrangian method for matrix inequality constrained programming. *Optim Methods Softw* 22(5):777–802

- Noll D, Apkarian P (2005) Spectral bundle methods for nonconvex maximum eigenvalue functions: first-order methods. *Math Prog Ser B* 104(2):701–727
- Noll D, Torki M, Apkarian P (2004) Partially augmented Lagrangian method for matrix inequality constraints. *SIAM J Optim* 15(1):161–184
- Noll D, Prot O, Rondepierre A (2008) A proximity control algorithm to minimize nonsmooth and nonconvex functions. *Pac J Optim* 4(3):571–604
- Noll D, Prot O, Apkarian P (2009) A proximity control algorithm to minimize nonsmooth and nonconvex semi-infinite maximum eigenvalue functions. *J Convex Anal* 16(3 & 4):641–666
- Oi A, Nakazawa C, Matsui T, Fujiwara H, Matsumoto K, Nishida H, Ando J, Kawaura M (2008) Development of PSO-based PID tuning method. In: International conference on control, automation and systems, pp 1917–1920
- Ravanbod L, Noll D, Apkarian P (2017) Branch and bound algorithm with applications to robust stability. *J Glob Optim* 67(3):553–579
- Robust Control Toolbox 42 (2012) The MathWorks Inc. Natick, MA
- Saldivar B, Mondié S, Loiseau J, Rasvan V (2013) Suppressing axial-torsional vibrations in drillstrings. *J Control Eng Appl Inf, SRAIT* 14:3–10
- Shamma JF, Athans M (1990) Analysis of gain scheduled control for nonlinear plants. *IEEE Trans Aut Control* 35(8):898–907
- Smyshlyaev A, Krstic M (2009) Boundary control of an anti-stable wave equation with anti-damping on the uncontrolled boundary. *Syst Control Lett* 58: 617–623
- Stein G, Doyle J (1991) Beyond singular values and loopshapes. *AIAA J Guid Control* 14:5–16
- Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River

Optimization-Based Robust Control

Didier Henrion

LAAS-CNRS, University of Toulouse, Toulouse, France

Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic

Abstract

This entry describes the basic setup of linear robust control and the difficulties typically encountered when designing optimization

algorithms to cope with robust stability and performance specifications.

Keywords

Linear systems · Optimization · Robust control

Linear Robust Control

Robust control allows dealing with uncertainty affecting a dynamical system and its environment. In this section, we assume that we have a mathematical model of the dynamical system without uncertainty (the so-called nominal system) jointly with a mathematical model of the uncertainty. We restrict ourselves to **linear systems**: if the dynamical system we want to control has some nonlinear components (e.g., input saturation), they must be embedded in the uncertainty model. Similarly, we assume that the control system is relatively small scale (low number of states): higher-order dynamics (e.g., highly oscillatory but low energy components) are embedded in the uncertainty model. Finally, for conciseness, we focus exclusively on continuous-time systems, even though most of the techniques described in this section can be transposed readily to discrete-time systems.

Our control system is described by the first-order ordinary differential equation

$$\begin{aligned}\dot{x} &= A(\delta)x + D(\delta)u \\ y &= C(\delta)x\end{aligned}$$

where as usual $x \in \mathbb{R}^n$ denotes the states, $u \in \mathbb{R}^m$ denotes the controlled inputs, and $y \in \mathbb{R}^p$ denotes the measured outputs, all depending on time t , with $\dot{x}$ denoting the time derivative of x . The system is subject to uncertainty, and this is reflected by the dependence of matrices A , B , and C on uncertain parameter δ which is typically time varying and restricted to some bounded set

$$\delta \in \Delta \subset \mathbb{R}^q.$$

A linear control law

$$u = Ky$$

modeled by a matrix $K \in \mathbb{R}^{m \times p}$ must be designed to overcome the effect of the uncertainty while optimizing some performance criterion (e.g., pole placement, disturbance rejection, H_2 or H_∞ norm). Sometimes, a relevant performance criterion is that the control should be stabilizing for the largest possible uncertainty (measured, e.g., by some norm on Δ). In this section, for conciseness, we restrict our attention to **static output feedback** control laws, but most of the results can be extended to dynamical output feedback control laws, where the control signal u is the output of a controller (a linear system to be designed) whose input is y .

Uncertainty Models

Among the simplest possible uncertainty models, we can find the following:

- **Unstructured uncertainty**, also called norm-bounded uncertainty, where

$$\Delta = \{\delta \in \mathbb{R}^q : \|\delta\| \leq 1\}$$

and the given norm can be a standard vector norm or a more complicated matrix norm if δ is interpreted as a vector obtained by stacking the column of a matrix

- **Structured uncertainty**, also called polytopic uncertainty, where

$$\Delta = \text{conv} \{\delta_i, i = 1, \dots, N\}$$

is a polytope modeled as the convex combination of a finite number of given vertices $\delta_i \in \mathbb{R}^q, i = 1, \dots, N$

We can find more complicated uncertainty models (e.g., combinations of the two above: see Zhou et al. 1996), but to keep the developments elementary, they are not discussed here.

Nonconvex Nonsmooth Robust Optimization

The main difficulties faced when seeking a feedback matrix K are as follows:

- **Nonconvexity**: The stability conditions are typically nonconvex in K .
- **Nondifferentiability**: The performance criterion to be optimized is typically a non-differentiable function of K .
- **Robustness**: Stability and performance should be ensured for every possible instance of the uncertainty.

So if we are to formulate the robust control problem as an optimization problem, we should be ready to develop and use techniques from nonconvex, nondifferentiable, robust optimization.

Let us first elaborate on the first difficulty faced by optimization-based robust control, namely, the nonconvexity of the stability conditions. In continuous time, stability of a linear system $\dot{x} = Ax$ is equivalent to negativity of the spectral abscissa, which is defined as the maximum real part of the eigenvalues of A :

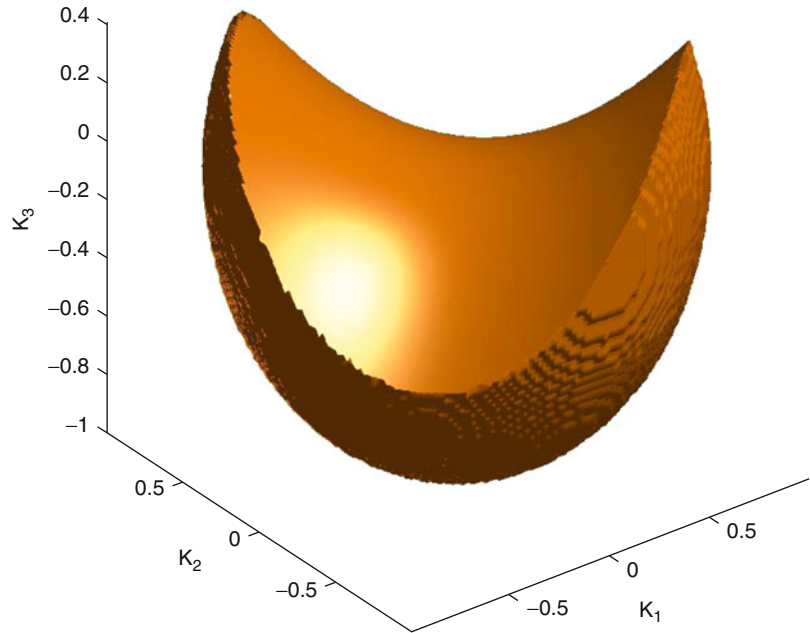
$$\alpha(A) = \max\{\text{Re } \lambda : \det(\lambda I_n - A) = 0, \lambda \in \mathbb{C}\}.$$

It turns out that the open cone of matrices $A \in \mathbb{R}^{n \times n}$ such that $\alpha(A) < 0$ is nonconvex (Ackermann 1993). This is illustrated in Fig. 1 where we represent the set of vectors $K = (k_1, k_2, k_3) \in \mathbb{R}^3$ such that $k_1^2 + k_2^2 + k_3^2 < 1$ and $\alpha(A(K)) < 0$ for

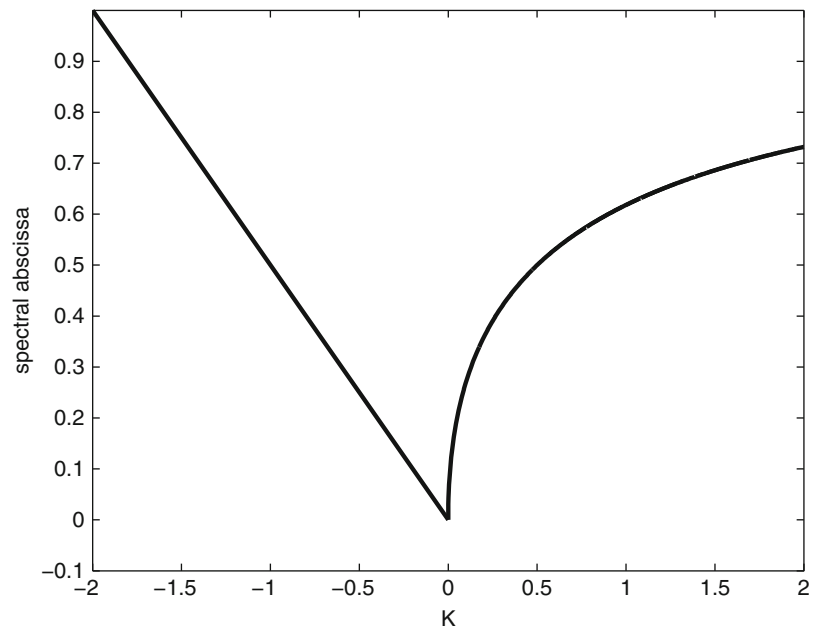
$$A(K) = \begin{pmatrix} -1 & k_1 \\ k_2 & k_3 \end{pmatrix}.$$

There exist various approaches to handling nonconvexity. One possibility consists of building convex inner approximations of the stability region in the parameter space. The approximations can be polytopes, balls, ellipsoids, or more complicated convex objects described by linear matrix inequalities (LMI). The resulting stability conditions are convex, but surely conservative, in the sense that the conditions are only sufficient for stability and not necessary. Another approach to handling nonconvexity consists of formulating the stability conditions algebraically (e.g., via the Routh-Hurwitz stability criterion or its symmetric version by Hermite) and using converging hierarchies of LMI relaxations to solve

Optimization-Based Robust Control, Fig. 1 A nonconvex ball of stable matrices



Optimization-Based Robust Control, Fig. 2 The spectral abscissa is typically nonconvex and nonsmooth



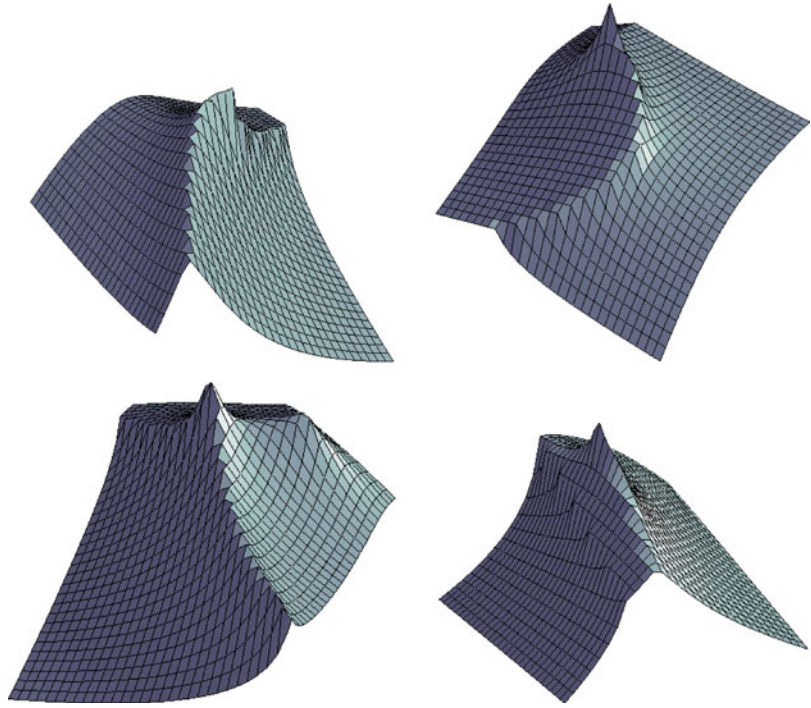
the resulting nonconvex polynomial optimization problem: see, e.g., Henrion and Lasserre (2004) and Chesi (2010).

The second difficulty characteristic of optimization-based robust control is the potential nondifferentiability of the objective function. Consider for illustration one of the simplest optimization problems which consists of minimizing

the spectral abscissa $\alpha(A(K))$ of a matrix $A(K)$ depending linearly on a matrix K . Such a minimization makes sense since negativity of the spectral abscissa is equivalent to system stability. Then typically, $\alpha(A(K))$ is a continuous but non-Lipschitz function of K , which means that its gradient can be unbounded locally. In Fig. 2, we plot the spectral abscissa $\alpha(A(K))$ for

Optimization-Based Robust Control, Fig. 3

The graph of the negative spectral abscissa for some randomly generated matrix parametrizations



$$A(K) = \begin{pmatrix} 0 & 1 \\ K & -K \end{pmatrix}$$

and $K \in \mathbb{R}$. The function is non-Lipschitz at $K = 0$, at which the global minimum $\alpha(A(0)) = 0$ is achieved. Nonconvexity of the function is also apparent in this example. The lack of convexity and smoothness of the spectral abscissa and other similar performance criteria renders optimization of such functions particularly difficult (Burke et al. 2001, 2006b). In Fig. 3, we represent graphs of the spectral abscissa (with flipped vertical axis for better visualization) of some small-size matrices depending on two real parameters, with randomly generated parametrization. We observe the typical nonconvexity and lack of smoothness around local and global optima.

The third difficulty for optimization-based robust control is the uncertainty. As explained above, optimization of a performance criterion with respect to controller parameters is already a potentially difficult problem for a nominal system (i.e., when the uncertainty parameter is equal to zero). This becomes even more difficult when this optimization must be carried out for

all possible instances of the uncertainty δ in Δ . This is where the above assumption that the uncertainty set Δ has a simple description proves useful. If the uncertainty δ is unstructured and not time varying, then it can be handled with the complex stability radius (Ackermann 1993), the pseudospectral abscissa (Trefethen and Embree 2005), or via an H_∞ norm constraint (Zhou et al. 1996). If the uncertainty δ is structured, then we can try to optimize a performance criterion at every vertex in the polytopic description (which is a relaxation of the problem of stabilizing the whole polytope). An example is the problem of simultaneous stabilization, where a controller K must be found such that the maximum spectral abscissa of several matrices $A_i(K)$, $i = 1, \dots, N$ is negative (Blondel 1994). Finally, if the uncertainty δ is time varying, then performance and stability guarantees can still be achieved with the help of Lyapunov certificates or potentially conservative convex LMI conditions: see, e.g., Boyd et al. (1994) and Scherer et al. (1997).

A unified approach to addressing conflicting performance criteria and uncertainty consists of searching for locally optimal solutions of a nonsmooth optimization problem that is built

to incorporate minimization objectives and constraints for multiple plants. This is called (linear robust) **multiobjective control**, and formally, it can be expressed as the following optimization problem

$$\begin{aligned} \min_K \max_{i=1,\dots,N} \{g_i(K) : \beta_i = \infty\} \\ \text{s.t. } g_i(K) \leq \beta_i, i = 1, \dots, N, \end{aligned}$$

where each $g_i(K)$ is a function of the closed-loop matrix $A_i(K)$ (e.g., a spectral abscissa or an H_∞ norm) and the scalars β_i are given and such that if $\beta_i = \infty$ for some i , then g_i appears in the objective function and not in a constraint: see Gumussoy et al. (2009) for details. In the above problem, the objective function, a maximum of nonsmooth and nonconvex functions, is typically also nonsmooth and nonconvex. Moreover, without loss of generality, we can easily impose a sparsity pattern on controller matrix K to account for structural constraints (e.g., a low-order decentralized controller).

Software Packages

Algorithms for **nonconvex nonsmooth optimization** have been developed and interfaced for linear robust multiobjective control in the public domain Matlab package HIFOO released in 2006 (Burke et al. 2006a) and updated in 2009 (Gumussoy et al. 2009) and based on the theory described in Burke et al. (2006b). In 2011, The MathWorks released HINFSTRUCT, a commercial implementation of these techniques based on the theory described in Apkarian and Noll (2006).

Cross-References

- [H-Infinity Control](#)
- [LMI Approach to Robust Control](#)

Bibliography

- Ackermann J (1993) Robust control – systems with uncertain physical parameters. Springer, Berlin
- Apkarian P, Noll D (2006) Nonsmooth H-infinity synthesis. IEEE Trans Autom Control 51(1):71–86

- Blondel VD (1994) Simultaneous stabilization of linear systems. Springer, Heidelberg
- Boyd S, El Ghaoui L, Feron E, Balakrishnan V (1994) Linear matrix inequalities in system and control theory. SIAM, Philadelphia
- Burke JV, Lewis AS, Overton ML (2001) Optimizing matrix stability. Proc AMS 129:1635–1642
- Burke JV, Henrion D, Lewis AS, Overton ML (2006a) HIFOO – a Matlab package for fixed-order controller design and H-infinity optimization. In: Proceedings of the IFAC symposium robust control design, Toulouse
- Burke JV, Henrion D, Lewis AS, Overton ML (2006b) Stabilization via nonsmooth, nonconvex optimization. IEEE Trans Autom Control 51(11):1760–1769
- Chesi G (2010) LMI techniques for optimization over polynomials in control: a survey. IEEE Trans Autom Control 55(11):2500–2510
- Gumussoy S, Henrion D, Millstone M, Overton ML (2009) Multiobjective robust control with HIFOO 2.0. In: Proceedings of the IFAC symposium on robust control design (ROCOND 2009), Haifa
- Henrion D, Lasserre JB (2004) Solving nonconvex optimization problems – how GloptiPoly is applied to problems in robust and nonlinear control. IEEE Control Syst Mag 24(3):72–83
- Scherer CW, Gahinet P, Chilali M (1997) Multi-objective output feedback control via LMI optimization. IEEE Trans Autom Control 42(7):896–911
- Trefethen LN, Embree M (2005) Spectra and pseudospectra: the behavior of nonnormal matrices and operators. Princeton University Press, Princeton
- Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River

Option Games: The Interface Between Optimal Stopping and Game Theory

Benoît Chevalier-Roignant¹ and Lenos Trigeorgis²

¹Emlyon Business School, Écully, France

²Bank of Cyprus Chair Professor of Finance, University of Cyprus, Nicosia, Cyprus

Abstract

Managers can stake a claim by committing to capital investments today that can influence their rivals' behavior or take a "wait-and-see" or step-by-step approach to avoid possible adverse market consequences tomorrow. At the core of this corporate dilemma lies the classic trade-off between commitment and

flexibility. This trade-off calls for a careful balancing of the merits of flexibility against those of commitment. This balancing is captured by option games.

Keywords

Real options · Game theory · Option games · Optimal stopping

Introduction

The global competitive environment has become increasingly challenging as modern economies undergo unprecedented changes in the midst of the global economic turmoil. Real-world dilemmas corporate managers face today are driven by the interplay among strategic and market uncertainty. The tech industry has evolved most rapidly, putting companies unable to respond to market developments and technological breakthroughs at severe disadvantage. Corporate management's plans and how they implement their strategy will likely determine whether the firm will survive and be successful in the marketplace or become extinct.

Formulating the right strategy in the right competitive environment at the right time is a nontrivial task. Whether to invest in a new technology or a new product or enter a new market is a strategic decision of immense importance. Corporate management must assess strategic options with proper analytical tools that can help determine whether to commit to a particular strategic path, given scarce or costly resources, or whether to stay flexible. Oftentimes, firms need to position themselves flexibly to capitalize on future opportunities as they emerge while limiting potential losses arising from adverse future circumstances. In many cases, corporate managers find themselves in need to revise their decision plans in view of actual market developments when facing an uncertain future; they can then decide to undertake only those projects with sufficiently high prospects in the future to justify commitment at that time. This needs to be balanced with the need to make irreversible strategic commitments to seize first-mover advantage presenting

rivals with a fait accompli to which they have no choice but adapt.

Capital Budgeting Ignoring Strategic Interactions

Net Present Value

Prevailing management approaches simplify matters and often lead to investment decisions that are detrimental to the firm's long-term well-being. Suppose a firm's future cash flow at time t is given by a random variable X_t . Cash flows then evolve as a geometric Brownian motion

$$dX_t = gX_t dt + \sigma X_t dB_t \quad \text{and } X_0 = x,$$

with drift parameter g and volatility σ . The Brownian motion $(B_t; t > 0)$ captures exogenous market uncertainty. The standard criterion used in corporate finance is based on discounted cash flows (DCF) or net present value (NPV). This consists in assessing the current value of a project by discounting the expected future cash flows $\mathbb{E}[X_t]$ at a constant discount rate, $r (> g)$. Management supposedly creates shareholder value by undertaking projects with positive NPV, i.e., projects for which the present value of cash flows, $v(x) := \int_0^\infty e^{-rt} \mathbb{E}[X_t] dt$, exceeds the necessary investment cost, I . In the present case, the firm will invest under the zero-NPV criterion if

$$\frac{x}{r - g} \geq I. \quad (1)$$

This traditional criterion views investment opportunities as now-or-never decisions under passive management. However, this precludes the possibility to adjust future decisions in case the market develops off the expected path. While market uncertainty is factored in through the discount rate, the flexibility management has is typically not properly accounted for.

Real Options Analysis

It has become standard practice in finance and strategy to interpret real investment opportunities as being analogous to financial options. This view

is well accepted among academics and practitioners alike and is at the core of real options analysis (ROA). ROA is an extension of option-pricing theory to real investment situations (Myers 1977; Trigeorgis 1996). This approach effectively allows one to capture the dynamic nature of decision-making since it factors in management's flexibility to revise and adapt its decision in the face of market uncertainty. ROA allows managers with flexibility to adapt to actual market developments as uncertainty gets resolved. Managers may, for example, delay the start (or closure) of a project depending on its prospects. This approach leverages on optimal stopping theory (see, e.g., Bensoussan and Lions 1982; Dixit and Pindyck 1994) and is considered to be more reflective of real decision-making than traditional methods. In the case the firm can delay the decision to invest, for example, the problem is one of optimal stopping:

$$V(s) := \sup_{\tau} \mathbb{E} e^{-r\tau} [v(X_{\tau}) - I].$$

The discount rate is the risk-free interest rate (Dixit and Pindyck 1994; Trigeorgis 1996). The time of managerial action, τ , is now a decision variable – random as the decision-maker faces an uncertain environment. This problem has an analytical solution characterized by a threshold policy, say a trigger $\bar{x}$, given by

$$\frac{\bar{x}}{r - g} = \frac{\beta}{\beta - 1} I \quad (2)$$

where β is the positive root of a quadratic function (see, e.g., Dixit and Pindyck 1994) and $\beta/(\beta - 1) > 1$. When decisions are costly or difficult to reverse, corporate managers would be more cautious and careful to make decisions. A firm should not always commit immediately – even if the NPV criterion (1) indicates so – but wait until the gross project value is sufficiently positive to cover the investment cost I by a factor larger than one, as expressed in (2). Investing prematurely may destroy shareholder value. Real options may justify sometimes undertaking projects with negative (static) net present value

if it creates a platform for growth options or delaying projects with positive NPV.

Accounting for Strategic Interactions in Capital Budgeting

Strategic Uncertainty

As natural monopolies have lost their secular well-protected positions owing to market liberalization in the European Union and elsewhere across the globe, strategic interdependencies have become new key challenge for managers. At the same time, sectors traditionally populated by multiple firms have undergone significant consolidation, often resulting in oligopolistic situations with a reduced number of players (e.g., Facebook in social media, Google in search, Amazon in e-commerce). The 2008 economic crisis has amplified these consolidation pressures. These two ongoing phenomena – liberalization and consolidation – have put high on the corporate agenda the assessment of strategic options under competition. Standard real options analysis often examines investment decisions as if the option holder has a proprietary right to exercise. This perspective may not be realistic in the new oligopolistic environment as several firms may share the right to a related investment opportunity in the industry.

Game Theory

In oligopolistic industries, firms often have difficulty predicting how rivals will behave and make decisions based on beliefs about their likely behavior. A theory that helps characterize beliefs and form predictions about which strategies opponents will follow is helpful in analyzing such oligopolistic situations. Game theory has traditionally been used to frame strategic interactions arising in conflict situations involving parties with different objectives or interests. In competitive markets, firms' actions have no material impact on the market outcomes and conversely on their rivals' decisions. Consequently, competitive equilibria can be derived without the use of game theoretical techniques. Game theory attempts to model behavior in

strategic situations or games in which one party's success based on its choices depends on the choices of other players through influencing one another's welfare. Game theory adopts a different perspective on optimization, as the focus is on the formation of beliefs about rivals' optimal strategies. Finance theory has been primarily concerned with "moves by nature," while game theory focuses on "optimization problems" involving multiple players. To solve a game, one needs to reduce a complex multiplayer problem into a simpler structure that captures the essence of the conflict situation. One can then derive useful predictions about how rivals are likely to react in a given situation. Game theory helped reshape microeconomics by providing analytical foundations for the study of market behavior and has been at the foundation of the Nobel Prize winning research field of industrial organization.

Dynamic game theory (see, e.g., Başar and Olsder 1998) addresses problems in which several parties are in repeated interaction. Strategic management approaches based on dynamic economic theory can provide a richer foundation for understanding developments and competitive reactions within an industry. As firm competitiveness involves interactions among several players (rivals, Suppliers, or clients), game theoretic analysis brings important insights into strategic management in addressing such issues as first- and second-mover advantages, firm entry and exit decisions, strategic commitment, reputation, signaling, and other informational effects. A key lesson is that, when firms react to one another, it may sometimes be appropriate for one firm to take an aggressive stance in expectation that rivals will back off. Dynamic industrial organization includes the analysis of "games of timing" such as preemption games or war of attrition, whereby firms decide on appropriate investment timing under rivalry.

Option Games

The earlier optimal stopping problem with several firms falls in the category of "games of timing" when a firm's entry decision influences another firm's market strategy. Option games are most suitable to help model situations where a firm that has a real option to (dis)invest faces

rivalry. When two (or more) firms decide on entry or more generally compete over a first-mover advantage, they face a "preemption game," while, if they decide on market exit or more generally wait to benefit from a "second-mover advantage," they face a "war of attrition."

In preemption games under uncertainty, the mathematical problem consists in finding a Nash equilibrium solution for the two-player equivalent of the above optimal stopping problem. This solution must also satisfy certain dynamic consistency criteria. For sequential investments, the follower is faced with a single-agent optimal investment timing problem; it will thus enter if the gross project value exceeds the investment cost by a sufficient factor. A firm entering the market early on, i.e., a leader, earns temporary monopoly rents as long as demand remains below the follower's entry threshold. Following the follower's entry, the firms act as a duopoly. A key question is whether the roles of the firms (i.e., leader vs. follower) are set or if they are part of an equilibrium. The case where the roles are set have been studied among others by Bensoussan et al. (2010). It is, however, more common to assume the roles are endogenous (see, e.g., Smets 1991; Grenadier 1996; Thijssen et al. 2012; Riedel and Steg 2017) as discussed below.

As long as the leader's value exceeds the follower's, there is an incentive for one firm to invest, but not necessarily for both of them, leading to a "coordination problem." The competitive pressure will dissipate away the leader's first-mover advantage, leading to a market entry point that is not socially optimal and to rent dissipation. Unfortunately, the multiplayer problem does not involve a simple analytical solution, since at each point a duopolist firm might end up in any of four distinct situations (two-by-two matrix) depending on the rival's entry decision. Option games indicate in each situation which driving force (commitment vs. flexibility) prevails and whether to go ahead with the investment or wait and see. Main drivers of the prevailing market equilibrium include the riskiness of the venture, σ , the magnitude of the first-mover advantage, and the exclusive or shared ability to reap the benefits of the investment vis-à-vis rivals. When firms can grasp a large first-mover advantage from investing early

but cannot differentiate themselves sufficiently from each other, they may be tempted to wage a preemptive war, investing prematurely at an early market stage that actually kills option value. If firms are more on an equal footing but do not see much benefit from investing early, they may prefer to wait and invest (jointly) at a later stage when the future market is sufficiently mature. If, however, one firm has a comparative cost advantage (e.g., a radical or drastic technological superiority), participants may prefer a consensual leader-follower investment arrangement involving less option value destruction.

ROA has traditionally focused on modeling timing decisions. Increasingly, academics account for richer strategy (see, e.g., Bensoussan and Chevalier-Roignant 2019). Following Huisman and Kort (2015), several papers discuss cases in which the two competing firms decide on their investment lump sums beside their investment times (see review in Huberts et al. 2015). Beside competitors, other stakeholders may be affected by a firm's decision. We should expect a greater convergence between the literature on real options and on principal-agent problems in continuous time (Sannikov 2008; Cvitanic and Zhang 2012).

Summary and Future Directions

Corporate management's strategic tool kit should provide clearer guidance on whether to pursue a wait-and-see stance in the face of uncertain market developments or jump on the first-mover bandwagon to build competitive advantage. We discussed two different modeling approaches that provide complementary perspectives and insights to help management deal with issues of flexibility versus commitment: real options and dynamic game theory. While each approach separately might turn a blind eye to flexibility or commitment, an integrative perspective through "options games" might provide the right balance and serve as a tool kit for adaptive competitive strategy. Both perspectives ultimately aim to derive better insights into industry dynamics under industry conditions characterized by both market and strategic uncertainty. Option games pave the way

for a consistent approach in addressing managerial decision-making, elevating the art of strategy to scientific analysis. Option games integrate in a common, consistent framework recent advances made in these diverse set of disciplines. This emerging field represents a promising strategic management tool that can help guide managerial decisions through the complexity of the modern competitive marketplace.

Cross-References

- [Learning in Games](#)
- [Mean Field Games](#)

Recommended Reading

Smit and Trigeorgis (2004) discuss related trade-offs with discrete-time real option techniques. Grenadier (2000) examines a number of continuous-time models. Chevalier-Roignant and Trigeorgis (2011) synthesize both types of "option games." An overview of the literature is provided in Chevalier-Roignant et al. (2011). Related works in this encyclopedia include Hajek (2015) and Shamma (2015).

Bibliography

- Başar T, Olsder GJ (1998) Dynamic noncooperative game theory. SIAM, Philadelphia
- Bensoussan A, Chevalier-Roignant B (2019) Sequential capacity expansion options. *Oper Res* 67(1):33–57
- Bensoussan A, Lions J-L (1982) Applications of variational inequalities in stochastic control. *Studies in mathematics and its applications*, vol 13. North-Holland, New York
- Bensoussan A, Diltz JD, Hoe S (2010) Real options games in complete and incomplete markets with several decision makers. *SIAM J Finan Math* 1(1):666–728
- Chevalier-Roignant B, Flath CM, Huchzermeier A, Trigeorgis L (2011) Strategic investment under uncertainty: a synthesis. *Eur J Oper Res* 215(3):639–650
- Chevalier-Roignant B, Trigeorgis L (2011) *Competitive strategy: options and games*. The MIT Press, Cambridge, MA
- Cvitanic J, Zhang J (2012) *Contract theory in continuous-time models*. Springer Science & Business Media, Heidelberg
- Dixit AK, Pindyck RS (1994) *Investment under uncertainty*. Princeton University Press, Princeton

- Grenadier SR (1996) The strategic exercise of options: development cascades and overbuilding in real estate markets. *J Finan* 51(5):1653–1679
- Grenadier SR (2000) Game choices: the intersection of real options and game theory. Risk Books, London
- Hajek B (2015) Auctions. In: Baillieul J, Samad T (eds) *Encyclopedia of systems and control*. Springer, London, pp 47–50
- Huberts NF, Huisman KJ, Kort PM, Lavrutich MN (2015) Capacity choice in (strategic) real options models: a survey. *Dyn Games Appl* 5(4):424–439
- Huisman KJ, Kort PM (2015) Strategic capacity investment under uncertainty. *RAND J Econ* 46(2):376–408
- Myers SC (1977) Determinants of corporate borrowing. *J Finan Econ* 5(2):147–175
- Riedel F, Steg J-H (2017) Subgame-perfect equilibria in stochastic timing games. *J Math Econ* 72:36–50
- Sannikov Y (2008) A continuous-time version of the principal-agent problem. *Rev Econ Stud* 75(3):957–984
- Shamma JS (2015) Learning in games. In: Baillieul J, Samad T (eds) *Encyclopedia of systems and control*. Springer, London, pp 620–628
- Smets F (1991) Exporting versus FDI: the effect of uncertainty, irreversibilities and strategic interactions. Ph.D. thesis, Yale University
- Smit HT, Trigeorgis L (2004) Strategic investment: real options and games. Princeton University Press, Princeton
- Thijssen JJ, Huisman KJ, Kort PM (2012) Symmetric equilibrium strategies in game theoretic real option models. *J Math Econ* 48(4):219–225
- Trigeorgis L (1996) Real options: managerial flexibility and strategy in resource allocation. The MIT Press, Cambridge, MA

Oscillator Synchronization

Bruce A. Francis

Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada

Abstract

The nonlinear Kuramoto equations for n coupled oscillators are derived and studied. The oscillators are defined to be synchronized when they oscillate at the same frequency and their phases are all equal. A control-

theoretic viewpoint reveals that synchronized states of Kuramoto oscillators are locally asymptotically stable if every oscillator is coupled to all others. The problem of synchronization in Kuramoto oscillators is closely related to rendezvous, consensus, and flocking problems in distributed control. These problems, with their elegant solution by graph theory, are discussed briefly.

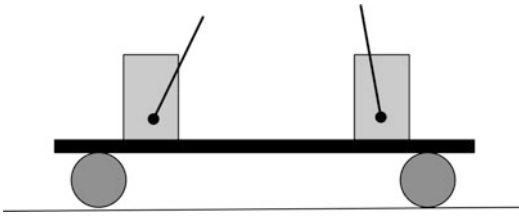
Keywords

Graph theory · Kuramoto model · Laplacian · Oscillator · Synchronization

Introduction

An oscillator is an electronic circuit or other kind of dynamical system that produces a periodic signal. If several oscillators are coupled together in some fashion and the periodic signals that they each produce are of the same frequency and are in phase, the oscillators are said to be synchronized. The book *Sync: The Emerging Science of Spontaneous Order*, by Strogatz, introduces a wide variety of phenomena where oscillators synchronize. Some examples from biology: networks of pacemaker cells in the heart, circadian pacemaker cells in the suprachiasmatic nucleus of the brain, metabolic synchrony in yeast cell suspensions, groups of synchronously flashing fireflies, and crickets that chirp in unison. Engineering examples include clock synchronization in distributed communication networks and electric power networks with synchronous generators.

A very simple example of oscillator synchronization was discovered by Christiaan Huygens, the prominent Dutch scientist and mathematician who lived in the 1600s. One of his contributions was the invention of the pendulum clock, where a pendulum swings back and forth with a constant frequency. Huygens observed that two pendulum clocks in his house synchronized after some time. The explanation for this phenomenon is that the pendula were coupled mechanically through the wooden frame of the house. The same principle can be observed by a fun, simple experiment. As



Oscillator Synchronization, Fig. 1 Two metronomes on a board that is on two pop cans. After the metronomes are let go at the same frequency but at different times, they soon become synchronized and tick in unison.

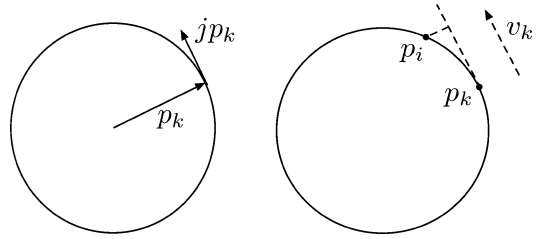
in Fig. 1, put two pop cans on a table, on their sides and parallel to each other. Place a board on top of them, and place two (or more) metronomes on the board. Set the metronomes to tick at the same frequency. Start them off ticking but not in unison. Within a few minutes they will be ticking in unison.

In this essay we derive what are known as the Kuramoto equations, a mathematical model of n oscillators, and then we study when they will synchronize.

The Kuramoto Model

In 1975 the Japanese researcher Yoshiki Kuramoto gave one of the first serious mathematical studies of coupled oscillators. To derive Kuramoto's equations, we begin with a simple hypothetical setup. Imagine n runners going around a circular track. Suppose they're all going at roughly the same speed, but each adjusts his/her speed based on the speeds of his/her nearest neighbors. If some runner passes another, that one tends to speed up to close the gap. The synchronization question is do the runners eventually end up running together in a tight pack?

Idealize the runners to be merely points, numbered $k = 1, \dots, n$. They move on the unit circle in the complex plane. A point on the unit circle can be written as $e^{j\theta}$, where j denotes the unit imaginary number and θ denotes the angle measured counterclockwise from the positive real axis. The position of point k at time t is



Oscillator Synchronization, Fig. 2 Left: The vectors p_k and jp_k . Right: The local velocity v_k

$z_k(t) = e^{j(\omega t + \theta_k(t))}$, where ω is the nominal rotational speed in rad/s, and $\theta_k(t)$ is the difference between the actual angle at time t and the nominal angle ωt . Notice that ω is a constant positive real number and it is the same for all n points. As in circuit theory, it simplifies the mathematics to refer all the positions to the sinusoid $e^{j\omega t}$, and therefore we define the **local position** of point k to be $p_k(t) = z_k(t)/e^{j\omega t}$, i.e., $p_k(t) = e^{j\theta_k(t)}$. Differentiate the local position with respect to time and let “dot” denote d/dt : $\dot{p}_k = e^{j\theta_k} j\dot{\theta}_k$. Define the local rotational velocity $v_k = \dot{\theta}_k$ and substitute into the preceding equation:

$$\dot{p}_k = v_k j p_k. \quad (1)$$

The local velocity v_k could be positive or negative. Notice that if we view p_k as a vector from the origin and view multiplication by j as rotation by $\pi/2$, then jp_k can be viewed as tangent to the circle at the point p_k – see the picture on the left in Fig. 2.

Now we propose a feedback law for v_k in Eq. (1); see the picture on the right in Fig. 2. Take v_k proportional to the projection of p_i onto the tangent at p_k , that is, $v_k = \langle p_i, jp_k \rangle$. Here the inner product between two complex numbers v, w is $\langle v, w \rangle = \text{Re } \bar{v}w$. (You may check that this is equivalent to the usual dot product of vectors in $\mathbb{R}^2$.) Thus from (1) the model to get k to close the gap is $\dot{p}_k = \langle p_i, jp_k \rangle jp_k$.

More generally, suppose that point k pays attention to not just point i but a fixed set of points called its **neighbors**. Let $\mathcal{N}_k$ denote the index set of neighbors of point k and for simplicity assume $\mathcal{N}_k$ does not depend on time. We consider the

control law $v_k = \sum_{i \in \mathcal{N}_k} \langle p_i, j p_k \rangle$ and thereby arrive at the model of the evolution of the positions p_k :

$$\dot{p}_k = \sum_{i \in \mathcal{N}_k} \langle p_i, j p_k \rangle j p_k.$$

However, the Kuramoto model gives the evolution of the angles θ_k rather than the points p_k . To find the equation for θ_k , we observe that

$$\begin{aligned} \langle p_i, j p_k \rangle &= \operatorname{Re}(\bar{p}_i j p_k) \\ &= \operatorname{Re}\left(e^{-j\theta_i} j e^{j\theta_k}\right) \\ &= \sin(\theta_i - \theta_k). \end{aligned}$$

In this way, the controlled points move according to

$$\dot{p}_k = \sum_{i \in \mathcal{N}_k} \sin(\theta_i - \theta_k) j p_k.$$

Substitute in $p_k = e^{j\theta_k}$ and then cancel $j p_k$:

$$\dot{\theta}_k = \sum_{i \in \mathcal{N}_k} \sin(\theta_i - \theta_k), \quad k = 1, \dots, n. \quad (2)$$

This is the **Kuramoto model of coupled oscillators** in terms of the phases of the oscillators. There are n coupled nonlinear ordinary differential equations.

Equation (2) has the vector form $\dot{\theta} = g(\theta)$. There are some variations in the literature about the state space associated with this equation. It is important to get the state space right because otherwise the concepts of stability and synchronization become shaky. The phase angles θ_k are real numbers with units of radians, so at first glance the state space is $\mathbb{R}^n$. But the angles are defined modulo 2π and so their values are restricted to lie in the interval $[0, 2\pi)$. In this way the state space becomes $[0, 2\pi)^n$. For example, if $n = 2$ the state space is the square $[0, 2\pi) \times [0, 2\pi)$ viewed as a subset of the plane $\mathbb{R}^2$. The mapping $\phi \mapsto e^{j\phi}$ is a one-to-one correspondence from the interval $[0, 2\pi)$ to the unit circle in the complex plane $\mathbb{C}$. This unit circle is usually denoted $\mathbb{S}^1$, the superscript signifying the circle's dimension as a manifold. By this correspondence the state space

of (2) is the n -fold product $\mathbb{S}^1 \times \dots \times \mathbb{S}^1$, and this is sometimes called the n -torus, denoted $\mathbb{T}^n$.

To recap, in what follows, the state space is $[0, 2\pi)^n$. This is an n -dimensional manifold rather than a vector space.

Synchronization

Control-theoretic methods, for example, that of Sepulchre et al. (2007), have been insightful. We address now the question of whether or not the oscillators in (2) synchronize, that is, the phases asymptotically converge to a common value. In the state space, $[0, 2\pi)^n$, the set of synchronized states is the set of vectors θ of the form $c\mathbf{1}$, where $c \in [0, 2\pi)$ and $\mathbf{1}$ is the vector of 1's. The simplest case is when every point is a neighbor of every other point, i.e., $\mathcal{N}_k$ contains every integer in the set $1, \dots, n$ except k . Then (2) becomes

$$\dot{\theta}_k = \sum_{i=1}^n \sin(\theta_i - \theta_k), \quad k = 1, \dots, n. \quad (3)$$

Let us show that if the initial phases $\theta_k(0)$ are all close enough together, then $\theta(t)$ converges asymptotically to a synchronized state. This will show that the synchronized states are locally asymptotically stable in a certain sense.

As stated before, Eq. (3) has the form $\dot{\theta} = g(\theta)$. The function $g(\theta)$ is the gradient of a positive definite function. Indeed, let $r e^{j\psi}$ denote the average of the points $e^{j\theta_1}, \dots, e^{j\theta_n}$. Of course, r and ψ are functions of θ , and so we have

$$r(\theta) e^{j\psi(\theta)} = \frac{1}{n} \left(e^{j\theta_1} + \dots + e^{j\theta_n} \right)$$

and therefore

$$r(\theta) = \frac{1}{n} \left| e^{j\theta_1} + \dots + e^{j\theta_n} \right|.$$

The average of n points on the unit circle lives inside the unit disc, and therefore $r(\theta)$ is a real number between 0 and 1. It equals 1 if and only if the n points are equal, that is, the n phases are

equal, and this is the state where the phases are synchronized.

Define the function

$$\begin{aligned} V(\theta) &= \frac{n^2}{2} r(\theta)^2 \\ &= \frac{1}{2} \left| e^{j\theta_1} + \dots + e^{j\theta_n} \right|^2 \\ &= \frac{1}{2} \left(e^{j\theta_1} + \dots + e^{j\theta_n} \right) \left(e^{-j\theta_1} + \dots + e^{-j\theta_n} \right). \end{aligned}$$

Thus

$$\frac{\partial V(\theta)}{\partial \theta_k} = \sin(\theta_1 - \theta_k) + \dots + \sin(\theta_n - \theta_k)$$

and therefore (3) can be written as $\dot{\theta} = \partial V(\theta)/\partial \theta$. This is a gradient equation. If $\theta(0)$ is chosen so that all the phases are close enough together, then $r(\theta(0))$ will be close to 1, and therefore θ will move in a direction to increase $V(\theta)$, that is, increase $r(\theta)$, until in the limit $r(\theta) = 1$ and the phases are synchronized.

There are results, e.g., Sepulchre et al. (2008), when the coupling is not all-to-all. Also, the term “synchronization” is used more generally than just for oscillators Wieland et al. (2011).

Rendezvous, Consensus, Flocking, and Infinitely Many Oscillators

Synchronization of coupled oscillators is closely related to other problems known as rendezvous, consensus, or flocking problems. Phase synchronization is replaced by the requirement of mobile robots gathering at some location, by the requirement of temperature sensors in a sensor network converging to the same temperature estimate, or by the requirement that mobile robots should head in the same direction. The simplest form of these problems has the equations

$$\dot{\theta}_k = \sum_{i \in \mathcal{N}_k} (\theta_i - \theta_k), \quad k = 1, \dots, n. \quad (4)$$

Notice that this can be obtained from the Kuramoto model (2) merely by replacing $\sin(\theta_i - \theta_k)$ by $\theta_i - \theta_k$ in (2), that is, by

linearizing the latter at a synchronized state. We shall continue to call θ_k a phase of an oscillator. When do the phases evolving according to (4) synchronize? The answer to the question involves a lovely collaboration between graph theory and dynamics.

Introduce a directed graph that is in one-to-one correspondence with the neighbor structure. The graph is made up of n nodes, one for each oscillator. From each node there is an arrow to every neighbor of that node; that is, from node k is an arrow to every node in $\mathcal{N}_k$. Denote the adjacency matrix and the degree matrix of the graph by, respectively, A and D . That is, $a_{ij} = 1$ if j is a neighbor of i and d_{ii} equals the sum of the elements on row i of A . The **Laplacian** of the graph is defined to be $L = D - A$. Then (4) is equivalent to simply

$$\dot{\theta} = -L\theta, \quad (5)$$

where θ is still the vector with elements $\theta_1, \dots, \theta_n$. Whether or not synchronization occurs depends on the connectivity of the graph. We stop here and refer the reader to the articles [► Averaging Algorithms and Consensus](#) and [► Flocking in Networked Systems](#)

Suppose there are an infinite but countable number of oscillators in the model (5). When will they synchronize? To answer this, we have to be more specific.

Let us allow an infinite number of oscillators numbered by the integers, positive, zero, and negative. Denote the phases by θ_k and let θ denote the phase vector, whose k th component is θ_k . Assume each oscillator has only finitely many neighbors, let $\mathcal{N}_k$ denote the set of neighbors of oscillator k , and let L be the Laplacian of the associated graph. Finally, let $\theta(t)$ evolve according to the Eq. (5). This equation isn't automatically well posed in the sense that there may not be a solution defined for all $t > 0$. We have to impose a framework so that solutions do indeed exist. One natural space in which to place $\theta(0)$ is ℓ^2 , the Hilbert space of square-summable sequences. If L is a bounded operator on ℓ^2 , then so is e^{-Lt} for every $t > 0$, and hence the phase vector exists and belongs to ℓ^2 for every $t > 0$. Another

natural space in which to place $\theta(0)$ is ℓ^∞ , the Banach space of bounded sequences. Again, a phase vector exists for all $t > 0$ if L is a bounded operator on ℓ^∞ .

The following example is from Feintuch and Francis (2012). Take the neighbor sets to be $\mathcal{N}_k = \{k - 1\}$. The graph is a chain: There is an arrow from node k to node $k - 1$, for every k , and the Laplacian is the infinite matrix with 1 on the diagonal, -1 on the first subdiagonal, and zero elsewhere. This Laplacian is a bounded operator on both ℓ^2 and ℓ^∞ . Now the vector $c\mathbf{1}$, where $\mathbf{1}$ is the vector of all 1's, belongs to ℓ^∞ for every real number c , but it belongs to ℓ^2 only for $c = 0$. So the phases can potentially synchronize at any value in ℓ^∞ , but only at 0 in ℓ^2 . For the example under discussion, if the initial phase vector is in ℓ^2 , then the phases synchronize at 0. By contrast, there exist initial phase vectors in ℓ^∞ such that synchronization does not occur. Even worse, $\lim_{t \rightarrow \infty} \theta(t)$ does not exist. The conclusion is that whether or not the oscillators will synchronize is a difficult question in general.

Summary and Future Directions

The Kuramoto model is a widely used paradigm for coupled oscillators. The model has the form $\dot{\theta} = f(E\theta)$, where θ is the vector of phases, the matrix E maps θ into the vector of possible differences $\theta_i - \theta_k$, and f is a function. The Kuramoto model considered in this essay is not the most general. A more general model allows different frequencies ω_k instead of just one, and also a coupling gain K , leading to the model

$$\dot{\theta}_k = \omega_k + \frac{K}{n} \sum_{i \in \mathcal{N}_k} \sin(\theta_i - \theta_k), \quad k = 1, \dots, n. \quad (6)$$

An important problem associated with the Kuramoto model is to determine which synchronized states are stable. The linearized equation is interesting in its own right and relates to problems of rendezvous, consensus, and flocking.

Reference Dörfler and Bullo (2014) offers some questions for future study. In particular, it would be interesting to extend the Kuramoto model beyond the first-order oscillators of (2). Also, the case of general neighbor sets has much room for exploration.

Asymptotic stability is a robust property. For example, if the origin is asymptotically stable for the system $\dot{x} = Ax$, it remains so if A is perturbed by a sufficiently small amount. This is because the spectrum of a matrix is a continuous function of the matrix. The sketch in Fig. 1 vividly depicts the concept of synchronized oscillators. A topic for future study is that of robustness. Mathematically, if the two metronomes are identical, they will synchronize perfectly – this can be proved. Of course, physically two metronomes cannot be identical, and yet they will synchronize if they are close enough physically. A mathematical study of this phenomenon might be interesting.

Cross-References

- [Averaging Algorithms and Consensus](#)
- [Flocking in Networked Systems](#)
- [Graphs for Modeling Networked Interactions](#)
- [Networked Systems](#)
- [Vehicular Chains](#)

Recommended Reading

The literature on the Kuramoto model is huge – there are now many hundreds of journal papers continuing the study of oscillators using Kuramoto's model. There is space here only to highlight a few sources.

You can find a mathematical study of coupled metronomes in Pantaleone (2002). Also, Pantaleone's webpage [Pantaleone](#) describes some experimental observations. Kuramoto's original paper is Kuramoto (1975). Dörfler and Bullo have recently written a comprehensive survey (Dörfler and Bullo (2014)). Strogatz has written extensively on oscillator synchronization. His book *Sync* is fascinating and is highly recommended (Strogatz 2004). See also Strogatz

(2000) and Strogatz and Stewart (1993). The papers Scardovi et al. (2007) and Dörfler and Bullo (2011) are recommended for more recent results, the latter treating the general model (6).

Getting phases in oscillators to synchronize is a special case of getting the states or outputs of coupled systems asymptotically to converge to a common value. There is a very large number of references on these subjects, a seminal one being Jadbabaie et al. (2003); others are Lin et al. (2007) and Moreau (2005). Regarding infinitely many oscillators, the physics literature treats only a continuum of oscillators, whereas countably many oscillators are the subject of Feintuch and Francis (2012).

Acknowledgments I greatly appreciate the help from Luca Scardovi, Florian Dörfler, and Francesco Bullo.

Bibliography

- Dörfler F, Bullo F (2011) On the critical coupling for Kuramoto oscillators. *SIAM J Appl Dyn Syst* 10(3):1070–1099
- Dörfler F, Bullo F (2014) Synchronization in complex networks of phase oscillators: a survey. *Automatica*, 50(6), June 2014. To appear.
- Feintuch A, Francis B (2012) Infinite chains of kinematic points. *Automatica* 48:901–908
- Jadbabaie A, Lin J, Morse AS (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans Automatic Control* 48(6):988–1001
- Kuramoto Y (1975) Self-entrainment of a population of coupled nonlinear oscillators. In: Araki H (ed) Volume 39 of International symposium on mathematical problems in theoretical physics, Kyoto. Lecture Notes in Physics. Springer, p 420
- Lin Z, Francis BA, Maggiore M (2007) State agreement for coupled nonlinear systems with time-varying interaction. *SIAM J Control Optim* 46:288–307
- Moreau L (2005) Stability of multi-agent systems with time-dependent communication links. *IEEE Trans Automatic Control* 50:169–182
- Pantaleone J. Webpage. <http://salt.uaa.alaska.edu/jim/>
- Pantaleone J (2002) Synchronization of metronomes. *Am J Phys* 70(10):992–1000
- Scardovi L, Sarlette A, Sepulchre R (2007) Synchronization and balancing on the N-torus. *Syst Control Lett* 56:335–341
- Sepulchre R, Paley DA, Leonard NE (2007) Stabilization of planar collective motion: all-to-all communication. *IEEE Trans Automatic Control* 52(5):811–824
- Sepulchre R, Paley DA, Leonard NE (2008) Stabilization of planar collective motion with limited communication. *IEEE Trans Automatic Control* 53(3): 706–719
- Strogatz SH (2000) From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D* 143:1–20
- Strogatz SH (2004) *Sync: the emerging science of spontaneous order*. Hyperion Books, New York
- Strogatz SH, Stewart I (1993) Coupled oscillators and biological synchronization. *Sci Am* 269:102–109
- Wieland P, Sepulchre R, Allgower F (2011) An internal model principle is necessary and sufficient for linear output synchronization. *Automatica* 47:1068–1074

Output Regulation Problems in Hybrid Systems

Sergio Galeani

Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma “Tor Vergata”, Roma, Italy

Abstract

This entry discusses some of the salient features of the output regulation problem for hybrid systems, especially in connection with the steady-state characterization. In order to better highlight such peculiarities, the discussion is mostly focused on the simplest class of linear time-invariant systems exhibiting such behaviors. In comparison with the usual regulation theory, the role played by the zero dynamics and by the presence of more inputs than outputs is particularly striking.

Keywords

Disturbance rejection · Hybrid systems · Internal model principle · Output regulation · Tracking · Zero dynamics

Introduction

Output regulation is one of the most classical problems in control theory, and its celebrated solution in the linear time-invariant case (Davison

1976; Francis and Wonham 1976) is characterized by remarkable elegance and ideas (like the internal model principle). While the extension to nonlinear systems is still an active field of investigation, the study of output regulation for hybrid systems is also being actively pursued, and several surprising results have already appeared for the linear case, suggesting that a richer structure arises in hybrid output regulation problems due to the interplay between flow and jump dynamics.

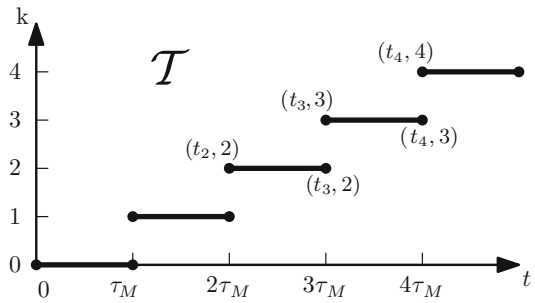
The problem can be stated as follows. A known *exosystem* $\mathcal{E}$ with initial state belonging to a suitably defined set $\mathcal{W}_0$ produces a signal w possibly affecting both the *plant* $\mathcal{P}$ and the *compensator* $\mathcal{C}$; the compensator has to guarantee that for any initial state of $\mathcal{E}$ in $\mathcal{W}_0$:

- all closed-loop responses are bounded;
- the output e of $\mathcal{P}$ asymptotically converges to zero.

In order to avoid trivialities, the *exosystem* $\mathcal{E}$ is assumed to be such that its state evolution from nonzero initial states in $\mathcal{W}_0$ is bounded and not asymptotically converging to zero, both in forward and in backward time.

Two typical embodiments of the output regulation problem are the *disturbance rejection* and the *reference tracking* problems. In *disturbance rejection*, w acts as a disturbance on $\mathcal{P}$ and cannot be measured by $\mathcal{C}$, and the output e from which the effect of w has to be canceled is the actual plant output. In *reference tracking*, w contains the references to be tracked by an output y_r of $\mathcal{P}$, so that w can be assumed to be known by $\mathcal{C}$; by defining the regulated output e as $e = y_r - r$, the reference tracking problem is cast as an output regulation problem.

The solution of an output regulation problem entails the solution of two subproblems: the definition of a set of *zero output steady-state solutions* and the *asymptotic stabilization* of such solutions (or at least making them *attractive*; in many cases of interest, the achievement of this last objective actually yields asymptotic stabilization). As a matter of fact, the stabilization subproblem is already widely studied and



Output Regulation Problems in Hybrid Systems, Fig. 1 Hybrid time domain $\mathcal{T}$ for a “sampled data” hybrid system. Dots indicate $(t, k) \in \mathcal{T}$ when jumps occur (see section “Hybrid Steady-State Generation” for the t_k notation)

described per se; for this reason, after some short remarks in section “Stabilization Obstructions in Hybrid Regulation,” the remainder of this presentation will focus only on steady-state-related issues, for the simplest class of systems which exhibit the most peculiar and interesting phenomena of hybrid steady-state behavior (see in particular section “Key Features in Hybrid vs Classical Output Regulation”). For concreteness, only hybrid systems $\mathcal{E}, \mathcal{P}$ characterized by linear time-invariant (flow and jump) dynamics will be considered; following Goebel et al. (2012b, Chap. 2), a two-dimensional parameterization of hybrid time $(t, k) \in \mathbb{R} \times \mathbb{N}$ will be used, with t measuring the flow of (usual) time and k counting the number of jumps experienced by the solution (see Fig. 1 for a specific example). So, the exosystem $\mathcal{E}$ will be described at time (t, k) by

$$\dot{w} = Sw, \quad (w, t, k) \in \mathcal{C}_{\mathcal{E}}, \quad (1a)$$

$$w^+ = Jw, \quad (w, t, k) \in \mathcal{D}_{\mathcal{E}}, \quad (1b)$$

and the plant $\mathcal{P}$ will be described at time (t, k) by

$$\dot{x} = Ax + Bu + Pw, \quad (x, u, t, k) \in \mathcal{C}_{\mathcal{P}}, \quad (2a)$$

$$x^+ = Ex + Rw, \quad (x, u, t, k) \in \mathcal{D}_{\mathcal{P}}, \quad (2b)$$

$$e = Cx + Qw, \quad (2c)$$

with $x(t, k) \in \mathbb{R}^n$, $u(t, k) \in \mathbb{R}^m$, $e(t, k) \in \mathbb{R}^p$, $w(t, k) \in \mathbb{R}^q$, and suitably defined flow sets $\mathcal{C}_{\mathcal{E}}$, $\mathcal{C}_{\mathcal{P}}$ and jump sets $\mathcal{D}_{\mathcal{E}}$, $\mathcal{D}_{\mathcal{P}}$.

Stabilization Obstructions in Hybrid Regulation

The achievement of asymptotic stabilization of the desired (zero output) steady-state responses for the considered class of linear hybrid systems crucially depends on whether the plant $\mathcal{P}$ and the exosystem $\mathcal{E}$ have synchronous jump times or not. While assuming synchronous jumps may appear restrictive, it is easy to see that *jumps are always synchronous steady state*, and then the findings (especially necessary conditions) about the synchronous case are fundamental for understanding the asynchronous case.

Asynchronous Jumps

Typically, jumps in $\mathcal{P}$ and $\mathcal{E}$ will be asynchronous, and this will cause the undesirable phenomenon that genuinely close trajectories will look “distant” around each jump when the distance is measured according to the usual Euclidean norm. The simplest illustration of such phenomenon consists in considering two trajectories of the same system starting from ε -close initial conditions. Consider the system

$$\dot{v} = 1, \quad v \in [0, 1], \quad v^+ = 0, \quad v \notin (0, 1),$$

with the initial states $v_0 = 0$ and $v_1 = \varepsilon$, $0 < \varepsilon < 1$. The two ensuing solutions at time (t, k) are immediately computed as

$$v(t, k; v_0) = t - k, \quad t \in [k, k + 1],$$

$$v(t, k; v_1) = \begin{cases} t - k + \varepsilon, & t \in [k, k + 1 - \varepsilon], \\ t - k + \varepsilon - 1, & t \in [k + 1 - \varepsilon, k + 1], \end{cases}$$

Hence, the (Euclidean) distance between the two solutions at time (t, k) is given by

$$d(t, k) = \begin{cases} \varepsilon, & t \in [k, k + 1 - \varepsilon], \\ (1 - \varepsilon), & t \in [k + 1 - \varepsilon, k + 1]; \end{cases}$$

in other words, choosing $\varepsilon > 0$ as small as desired, arbitrarily close initial conditions generate trajectories which are apart by a finite amount (as close as desired to 1) during the arbitrarily small time intervals where $t \in [k + 1 - \varepsilon, k + 1]$. Since stability deals with trajectories remaining close forever and attractivity deals with trajectories getting closer and closer, examples such as the one above pose serious issues when defining (let alone establish) stability and attractivity in the hybrid case. Similar problems arise not only in output regulation problems but also in other areas like state tracking, observers, and general interconnections of hybrid systems.

However, intuition suggests (and mathematics confirms, by using a suitable notion of “distance”) that such trajectories are close indeed. Several approaches have been proposed in order to overcome such difficulty. Considering as an example a bouncing ball tracking another bouncing ball, the problematic time intervals are those between the bounce of the first ball hitting the ground and the bounce of the other ball; in such a case, the modified distances are defined by either

- Allowing to exclude sufficiently short “problematic” intervals (possibly requiring that their length asymptotically tends to zero); see, e.g., Galeani et al. (2008, 2012a)
- Considering alternative “mirrored” trajectories computed as if the last jump did not happen; see, e.g., Forni et al. (2013a)
- Using a “stretched” distance function δ such that when point a is in the jump set and its image via the jump map is $g(a)$, then $\delta(a, b) = \delta(g(a), b)$; see, e.g., Biemond et al. (2013).

While the first approach has been proposed first, the other two (which are strongly related) have the advantage of providing (under mild additional hypotheses) global control Lyapunov functions. Finally, it is worth noting that the most adequate tools to address similar issues for general hybrid systems are the “graphical distance” among hybrid arcs and related concepts (see Goebel et al. 2012b, Chap. 5).

Synchronous Jumps

When synchronous jumps are considered, the above issue disappears, and asymptotic stabilization becomes a much simpler matter. Although synchronous jumps look more like an exception than a rule in hybrid systems, they are very reasonable for specific classes of problems.

In order to have synchronous jumps, some authors have considered the use of “jump inputs” which *impose a jump at a certain time*, which can be physically reasonable in some systems, e.g., two tanks separated by a movable wall, assuming that when the wall is removed the fluid reaches the equilibrium configuration almost instantaneously.

Another relevant class consists of “sampled data” systems, whose jumps are essentially due to digital components which operate at a fixed sampling rate, which will be considered in the rest of this entry. In such a case, letting τ_M be the sampling period, the time domain of the hybrid system is fixed as (see Fig. 1)

$$\mathcal{T} := \{(t, k) : t \in [k\tau_M, (k+1)\tau_M], k \in \mathbb{Z}\}, \quad (3)$$

all jumps happen exactly for (t, k) with $t = (k+1)\tau_M$, and then (1) can be simplified as

$$\dot{w} = Sw, \quad (4a)$$

$$w^+ = Jw, \quad (4b)$$

and (2) can be simplified as

$$\dot{x} = Ax + Bu + Pw, \quad (5a)$$

$$x^+ = Ex + Rw, \quad (5b)$$

$$e = Cx + Qw, \quad (5c)$$

since flow and jump times are clear from the context.

For the latter class of systems, by using linear time-invariant hybrid control laws and observers (and an easily provable separation principle), it is easily shown that:

- Under a hybrid stabilizability hypothesis, state feedback stabilization of (5) is easily achieved.
- Output feedback stabilization of (5) from e is also trivial under an additional hybrid detectability hypothesis.
- Under hybrid detectability of the cascade of (4) and (5), w can be asymptotically estimated from e .

Due to the above facts, it can be assumed without loss of generality that (5) is asymptotically stable (equivalently, that all eigenvalues of $Ee^{A\tau_M}$ have modulus strictly less than one). Asymptotic stability then yields *incremental stability*, since letting $\hat{x}$ and $\check{x}$ denote two motions under the same inputs u, w and only differing in their initial states, it is immediate to see that their difference $\tilde{x} := \hat{x} - \check{x}$ evolves as

$$\dot{\tilde{x}} = \dot{\hat{x}} - \dot{\check{x}} = A\hat{x} + Bu + Pw$$

$$- (A\check{x} + Bu + Pw),$$

$$\tilde{x}^+ = \hat{x}^+ - \check{x}^+ = E\hat{x} + Rw - (E\check{x} + Rw),$$

that is, $\dot{\tilde{x}} = A\tilde{x}$, $\tilde{x}^+ = E\tilde{x}$, and so it is just a free motion of the plant, asymptotically converging to zero. Incremental stability implies that *regulation is achieved as soon as it is shown that for any exogenous input w it is possible to find an input u and an initial state of (5) such that e is identically zero*, since then any other motion arising from a different initial state will asymptotically converge to the motion with identically zero e . Moreover, it is easy to see that asymptotic stability of the origin actually implies uniform, global, and exponential stability of any trajectory for such systems.

Hybrid Steady-State Generation

From this point on, the rest of the presentation will be focused only on the case where the problem data are of the form (3) to (5), since this allows to provide an uncluttered view on some peculiar features of hybrid steady-state motions,

without the burden of having to take care of delicate stability issues arising in more general contexts.

Based on the preceding discussion, there is no loss of generality at this point in assuming that:

- *Plant (5) is asymptotically stable*, which is equivalent to all eigenvalues of $Ee^{A\tau_M}$ having a magnitude strictly less than one.
- *Exosystem (4) is Poisson stable*, which is equivalent to all eigenvalues of $Je^{S\tau_M}$ having a magnitude equal to one.

It is also customary to distinguish between *full information* and *error feedback* regulation, where in the first case controller, $\mathcal{C}$ has access to the complete state (w, x) of the cascade of $\mathcal{E}$ and $\mathcal{P}$, whereas in the second case, $\mathcal{C}$ can only measure the output e of $\mathcal{P}$.

Having assumed asymptotic stability of plant $\mathcal{P}$, the only role of compensator $\mathcal{C}$ consists in generating the correct steady-state input, since then, by incremental stability of $\mathcal{P}$, asymptotic regulation is ensured from any initial state. Recalling the expression of $\mathcal{T}$ in (3), for the following developments, it is useful to define the jump times t_k and the elapsed time of flow since last jump σ as

$$t_k := k\tau_M, \quad \sigma(t, k) := t - k\tau_M;$$

the arguments of $\sigma(t, k)$ will usually be omitted since clear from the context. Note that σ satisfies $\dot{\sigma} = 1$, $\sigma^+ = 0$, and it is often explicitly introduced as an additional *timer* variable.

The Full Information Case

Consider the full information regulator:

$$u(t, k) = \Gamma(\sigma)w(t, k) \quad (6)$$

and the candidate steady-state motion and input:

$$\begin{bmatrix} x_{ss}(t, k) \\ u_{ss}(t, k) \end{bmatrix} = \begin{bmatrix} \Pi(\sigma) \\ \Gamma(\sigma) \end{bmatrix} w(t, k). \quad (7)$$

Requiring that such expressions actually characterize a response of the considered plant, and that the associated output is zero, amounts to ask that:

- during flows, $\dot{x}_{ss}(t, k)$ has to satisfy the two equations:

$$\begin{aligned} \dot{x}_{ss}(t, k) &= \dot{\Pi}(\sigma)w(t, k) + \Pi(\sigma)\dot{w}(t, k), \\ \dot{x}_{ss}(t, k) &= Ax_{ss}(t, k) + Bu_{ss}(t, k) \\ &\quad + Pw(t, k); \end{aligned}$$

- at jumps, $x_{ss}^+(t, k)$ has to satisfy the two equations:

$$\begin{aligned} x_{ss}^+(t_{k+1}, k) &= \Pi(0)w^+(t_{k+1}, k), \\ x_{ss}^+(t_{k+1}, k) &= Ex_{ss}(t_{k+1}, k) + Rw(t_{k+1}, k); \end{aligned}$$

- for the output e_{ss} to be identically zero:

$$0 = Cx_{ss}(t, k) + Qw(t, k).$$

Substituting (7) in the above conditions and considering that such relations should hold for all values of w , the following *hybrid regulator equations* are obtained:

$$\dot{\Pi}(\sigma) + \Pi(\sigma)S = A\Pi(\sigma) + B\Gamma(\sigma) + P, \quad (8a)$$

$$\Pi(0)J = E\Pi(\tau_M) + R, \quad (8b)$$

$$0 = C\Pi(\sigma) + Q. \quad (8c)$$

Equations (8) are necessary and sufficient for (7) to solve the output regulation problem under the considered assumptions. Once a solution of (8) is available, the full information regulator (6) is just a time-varying static feedforward controller which provides as input the steady-state input u_{ss} characterized as in (7); in fact, since (5) is incrementally stable (as follows from its asymptotic stability, which was assumed without loss of generality), its output response under the control law (6) must converge to the output response associated to (7).

For later use, note that in the non-hybrid case where $\mathcal{P}$ and $\mathcal{E}$ only flow

$$\dot{w} = Sw, \quad (9a)$$

$$\dot{x} = Ax + Bu + Pw, \quad (9b)$$

$$e = Cx + Qw, \quad (9c)$$

the candidate steady state (7) is replaced by

$$\begin{bmatrix} x_{ss}(t) \\ u_{ss}(t) \end{bmatrix} = \begin{bmatrix} \Pi \\ \Gamma \end{bmatrix} w(t), \quad (10)$$

and (8) reduces to the celebrated *regulator equations* (or *Francis equations*)

$$\Pi S = A\Pi + B\Gamma + P, \quad (11a)$$

$$0 = C\Pi + Q, \quad (11b)$$

and, as above, assuming without loss of generality that the plant is asymptotically stable, the full information regulator reduces to the time-invariant static feedforward controller:

$$u(t, k) = \Gamma w(t, k). \quad (12)$$

The Error Feedback Case

If the exosystem's state is not measured, a *dynamic* compensator $\mathcal{C}$ is introduced, described by

$$\dot{\xi} = F\xi + Ge, \quad (13a)$$

$$\xi^+ = L\xi, \quad (13b)$$

$$u = H\xi. \quad (13c)$$

This compensator is also supposed to flow and jump according to the a priori fixed time domain $\mathcal{T}$ considered for the plant. The candidate steady-state motion including ξ is

$$\begin{bmatrix} x_{ss}(t, k) \\ \xi_{ss}(t, k) \\ u_{ss}(t, k) \end{bmatrix} = \begin{bmatrix} \Pi(\sigma) \\ \Sigma(\sigma) \\ \Gamma(\sigma) \end{bmatrix} w(t, k). \quad (14)$$

Mimicking previous arguments, that is, imposing that (14) is a response associated with zero output e , leads to the conclusion that (8) must be complemented with:

$$\dot{\Sigma}(\sigma) + \Sigma(\sigma)S = F\Sigma(\sigma), \quad (15a)$$

$$\Sigma(0)J = L\Sigma(\tau_M). \quad (15b)$$

$$\Gamma(\sigma) = H\Sigma(\sigma), \quad (15c)$$

Equations (8) and (15) are necessary and sufficient for (13) to solve the output regulation problem under the considered assumptions; they generalize the corresponding conditions for the non-hybrid case where $\mathcal{P}$ and $\mathcal{E}$ only flow (see (9)) and (13) and (14) are replaced by

$$\dot{\xi} = F\xi + Ge, \quad (16a)$$

$$u = H\xi, \quad (16b)$$

$$\begin{bmatrix} x_{ss}(t) \\ \xi_{ss}(t) \\ u_{ss}(t) \end{bmatrix} = \begin{bmatrix} \Pi \\ \Sigma \\ \Gamma \end{bmatrix} w(t), \quad (16c)$$

and (15) reduces to

$$\Sigma S = F\Sigma, \quad (17a)$$

$$\Gamma = H\Sigma. \quad (17b)$$

Relations (17) are an expression of the *internal model principle*, stating that in order to achieve error feedback regulation, the compensator $\mathcal{C}$ must include a suitable “copy” of the exosystem, enabling $\mathcal{C}$ to reproduce the signal $u_{ss} = \Gamma w$ (used in the full information case) as $u_{ss} = H\Sigma\xi$, without measuring w ! More specifically (denoting by M' , $\text{Im}(M)$, the transpose and the image of matrix M), Eq. (17b) requires that $\text{Im}(\Gamma') \subset \text{Im}(\Sigma')$, namely, that the row space of Σ is “sufficiently rich”; in turn, (17a) is a homogeneous Sylvester equation whose solution Σ can be nonzero if and only if F and S have common eigenvalues (hence, F must “copy” S , and the larger the $\text{Im}(\Sigma')$, the more common eigenstructure must be shared between F and S). A similar interpretation holds for (15), which expresses the *hybrid internal model principle*.

Key Features in Hybrid vs Classical Output Regulation

While the previous section mainly aimed at showing how the classical theory generalizes in the hybrid case (at least for a special class of hybrid systems), the aim of this section is to point out some of the striking differences between the two cases. Before proceeding further, and in order to keep focus on the characterization of the steady-state response, it is worth mentioning here that although time-varying systems will be considered, no issue regarding nonuniform stability (like in general nonautonomous systems) arises since the timer σ just ranges in the compact set $[0, \tau_M]$ due to the assumed periodic structure of $\mathcal{T}$ (see also the end of section “Synchronous Jumps”).

Comparing the classical and the hybrid output regulator and considering that $\mathcal{P}$ and $\mathcal{E}$ are time invariant, it seems somewhat strange that in the output feedback case, the linear time-invariant regulator (16a) and (16b) generalizes to a hybrid linear time-invariant regulator (13), whereas in the full information case, the linear time-invariant regulator (12) generalizes to a hybrid linear *time-varying* regulator (6).

One argument in favor of the *time-varying* regulator (6) is based on the following consideration. It is well-known that (11) has a unique solution in the case of a square plant ($m = p$) under the *nonresonance condition* between the zeros of $\mathcal{P}$ and the eigenvalues of $\mathcal{E}$, requiring that

$$\text{rank} \begin{bmatrix} A - sI & B \\ C & 0 \end{bmatrix} = n + p, \quad \forall s \in \Lambda(S),$$

where $\Lambda(S)$ denotes the spectrum of S . In such a case, (11) amounts to a system of $nq + pq$ linear equations in $nq + mq$ unknowns (the elements of Π, Γ), which might be expected to be satisfied since $m \geq p$. If one were trying to use the unique constant solution (Π, Γ) of (11) as a solution of (8), clearly (8a) and (8c) would be satisfied, but then (8b) would impose other nq equations on Π which would unlikely be satisfied. For this reason, apparently, the additional degree of freedom offered by choosing time-dependent Π

and Γ might be of help. In fact, it can be shown that if $m = p$ and under a hybrid nonresonance condition (involving $Ee^{A\tau_M}$ and $Je^{S\tau_M}$) between $\mathcal{P}$ and $\mathcal{E}$, (8a) and (8b) have a unique solution for any choice of $\Gamma(\sigma)$, so that the design boils down to satisfying (8c) by choosing $\Gamma(\sigma)$; but is this always possible? In order to answer this nontrivial question, a different path must be followed. While a complete formal analysis can be performed, the following discussion will be mainly based on showing the simplest examples exhibiting the pathologies of interest.

Consider the system with $\tau_M = 1$ (so that $t_k = k$, for all $k \in \mathbb{Z}$) and

$$\dot{w} = 0, \quad (18a)$$

$$w^+ = -w, \quad (18b)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w, \quad (18c)$$

$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 2e & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad (18d)$$

$$e = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - w. \quad (18e)$$

The unique steady-state solution achieving output regulation is easily computed. By (18a) and (18b),

$$w(t, k) = (-1)^k w(0, 0);$$

then, by (18e), it appears that $e_{ss} = 0$, $\forall (t, k) \in \mathcal{T}$ implies

$$x_{2,ss}(t, k) = w(t, k) = (-1)^k w(0, 0),$$

$$\forall (t, k) \in \mathcal{T},$$

which in turn implies that $\dot{x}_{2,ss} = 0$ for all $t \in (k, k+1)$, $k \in \mathbb{Z}$ and the unique steady-state input

$$u_{ss} = 2x_{2,ss} - w.$$

Since (18d) implies that $x_{1,ss}(t_{k+1}, k+1) = x_{2,ss}(t_{k+1}, k) = w(t_{k+1}, k)$ and (18c) implies that $x_{1,ss}(t, k) = -e^{-(t-k)} x_{1,ss}(t_k, k)$, for $t \in$

(t_k, t_{k+1}) , it follows that

$$x_{1,ss}(t, k) = -e^{-(t-t_k)} w(t_k, k), \quad t \in (t_k, t_{k+1}), \quad (19)$$

which finally is coherent with the jump equation for $x_{2,ss}$ in (18d) since

$$\begin{aligned} x_{2,ss}(t_{k+1}, k+1) &= 2ex_{1,ss}(t_{k+1}, k) + x_{2,ss}(t_{k+1}, k) \\ &= 2e(-e^{-1})w(t_k, k) + w(t_k, k) \end{aligned} \quad (20a)$$

$$= -w(t_k, k) \quad (20b)$$

$$= (-1)^{k+1} w(0, 0). \quad (20c)$$

Before commenting the meaning of the above derived steady-state evolution, it is worth noting that (18) might actually derive from an original system with (18c) replaced by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w, \quad (21)$$

under the preliminary state feedback

$$u = -x_1 + v. \quad (22)$$

Such a feedback renders the subspace $\{x : x_2 = 0\}$ unobservable (when the system only flows) and reveals that the dynamics of x_1 in (18c) is the *flow zero dynamics* of $\mathcal{P}$, that is, the zero dynamics of $\mathcal{P}$ when jumps are inhibited. Having set the stage, several interesting observations can be made now.

The flow zero dynamics samples the exogenous signal w at jumps and then evolves according to its own modes (see (19)). In fact, while in the classical case (10), the state and input at steady state can be expressed as a constant matrix times the current value of w , the real nature of the time dependence of Γ and Π in (7) is linked to this phenomenon of *sampling* $w(t_k, k)$ and *propagating along the zero dynamics*. A suitable analysis shows that $\Pi(\sigma)$ and $\Gamma(\sigma)$ contain products of matrices with rightmost factor $e^{-S\sigma}$ (which recovers $w(t_k, k) = e^{-S\sigma} w(t, k)$ from the current value $w(t, k)$ of w) and leftmost factor containing the fundamental matrix of the flow

zero dynamics. It is worth mentioning that the “motion along the zeros” in the present context is strongly related to the same kind of motions used for perfect tracking in non-hybrid systems. The above insight about the nature of the dependence on σ in (7) also reveals why in the output feedback case (13) such dependence is not needed: the required modes of the flow zero dynamics in that case are provided by copying them in the compensator dynamics!

An even stronger consequence of the analysis above is a **flow zero dynamics internal model principle**, which essentially states that any output feedback compensator solving the output regulation problem must be able to produce as free responses (during flow) a suitable subset of the natural modes of the flow zero dynamics (and a suitably modified version applies to the feedforward static compensator (6)). Note that while the *classical internal model principle* requires only *exact knowledge of the exosystem modes* (which is kind of a mild requirement, especially when the exosystem models references, or constant offsets), the *flow zero dynamics internal model principle* requires the *exact knowledge of the modes of the flow zero dynamics*, which typically depends on unknown plant parameters; clearly, this fact is a serious obstruction to achieve robust regulation. To date, three approaches have been proposed to deal with this obstruction; in order of increasing generality and complexity, they are:

- (i) assume that the modes of the flow zero dynamics are unaffected by parameter variations;
- (ii) assume that the flow zero dynamics is decoupled during flows;
- (iii) use an adaptive internal model tuned on an online estimate of the modes of the flow zero dynamics.

The solution in (i), proposed in Marconi and Teel (2013) (and adopted in subsequent works by the same authors, e.g., Cox et al. (2016)), implies both *structurally restricted* and *coordinated* plant parameter variations (such that if one parameter changes, the other parameters also have to change to keep the modes of the flow

zero dynamics fixed). The solution in (ii), proposed in Carnevale et al. (2017), only imposes another kind of *structurally restricted* variations, often satisfied in physical systems where the flow zero dynamics is actually associated to a part of the system that only interacts at jumps but is decoupled during flows; the parameters are otherwise allowed to vary independently. Finally, the solution in (iii), proposed in de Carolis et al. (2020), solves the general problem, imposing no restriction on plant parameter variations but requiring an essentially nonlinear compensation.

A final point, also raising serious issues about what can be robustly achieved (and how) in the setting of hybrid output regulation, is the fact that **generically, existence of solutions is not robust to arbitrarily small parameter variations**. In particular, looking again at the computations in (20), it should be clear that the involved functions are all fixed by previous reasonings, whereas satisfaction of (20) crucially depends on exact cancellations of certain coefficients. Any small variations of such coefficients in (18d) imply that the problem admits no steady state yielding zero output. This fact is in sharp contrast with classical regulation, where the nonresonance condition ensures existence of (different) solutions for small parameter variations. It has to be noted, though, that **under additional conditions, robust existence of solutions is guaranteed if the plant is fat**, that is, $m > p$. Using again the previous example, this is the case if an additional input is introduced

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w,$$

since then even a constant (suitably chosen) value of u_1 can be used to ensure that when the time to jump arrives, the value of x_1 is such to ensure a correct jump for x_2 (remember that since x_1 is unobservable during flows, its motion can be changed as wished if this helps with ensuring that the observable x_2 achieves zero output). Again, if *structurally restricted* and *coordinated* plant parameter variations are considered, the problem can be addressed for square systems as in Marconi and Teel (2013) and Cox et al. (2016); for

general parameter variations, fat plants have to be considered with the approach in Carnevale et al. (2017) and de Carolis et al. (2020).

Summary and Future Directions

The investigation of the output regulation problem for hybrid systems is still at a very early stage. While the issues of stabilization of the manifold where regulation is achieved seem to be a relatively better understood topic (possibly drawing from a richer literature on stabilization of hybrid systems), the geometry and design of such manifold appear to involve several much more intricate issues, whose understanding will be crucial in order to achieve more complete solutions.

Already in the very simplified case of linear dynamics and synchronous jumps, the important role played by the whole flow zero dynamics for feasibility (existence of solutions in the nominal parameter values) and by the availability of more inputs than outputs for well posedness (existence of solutions for slightly perturbed parameter values) marks a strong difference with the linear non-hybrid case, where both properties are granted by satisfaction of the nonresonance condition, which only involves the spectrum of the zero dynamics, even for square plants.

While the expected final goal of this investigation should hopefully lead to the design of robust output regulators based on a suitable internal model principle, a deeper understanding of the structure of the steady-state motion achieving regulation, as well as of the effect of additional inputs in shaping it, seems to be an important preliminary step toward such goal.

Cross-References

- [Hybrid Dynamical Systems, Feedback Control of](#)
- [Nonlinear Zero Dynamics](#)
- [Regulation and Tracking of Nonlinear Systems](#)

Recommended Reading

Foundational contributions on classical output regulation are Francis and Wonham (1976) and Davison (1976); more recent monographs include Huang (2004), Trentelman et al. (2001) and Saberi et al. (2000). Goebel et al. (2012b) provides a solid introduction to a powerful and elegant framework for hybrid systems, including a thorough discussion of stability issues related to those mentioned here. Regulation problems (mainly reference tracking) for classes of hybrid systems with asynchronous jumps are presented in Biemond et al. (2013), Forni et al. (2013a), Morarescu and Brogliato (2010), and Galeani et al. (2008, 2012a). The class of linear systems with synchronous jumps considered in sections “Hybrid Steady State Generation” and “Key Features in Hybrid vs Classical Output Regulation” has been proposed in Marconi and Teel (2010, 2013), and the issues related to flow zero dynamics, fat plants and robustness have been discussed in Carnevale et al. (2012a, b, 2013), partly developing remarks contained in Galeani et al. (2008, 2012a).

Bibliography

- Biemond J, van de Wouw N, Heemels W, Nijmeijer H (2013) Tracking control for hybrid systems with state-triggered jumps. *IEEE Trans Autom Control* 58(4):876–890
- Carnevale D, Galeani S, Menini L (2012a) Output regulation for a class of linear hybrid systems. Part 1: trajectory generation. In: Conference on decision and control, Maui, pp 6151–6156
- Carnevale D, Galeani S, Menini L (2012b) Output regulation for a class of linear hybrid systems. Part 2: stabilization. In: Conference on decision and control, Maui, pp 6157–6162
- Carnevale D, Galeani S, Sassano M (2013) Necessary and sufficient conditions for output regulation in a class of hybrid linear systems. In: Conference on decision and control, Florence, pp 2659–2664
- Carnevale D, Galeani S, Menini L, Sassano M (2017) Robust hybrid output regulation for linear systems with periodic jumps: semi-classical internal model design. *IEEE Trans Autom Control* 62(12):6649–6656
- Cox N, Marconi L, Teel AR (2016) Isolating invisible dynamics in the design of robust hybrid internal models. *Automatica* 68(6):56–68
- Davison E (1976) The robust control of a servomechanism problem for linear time-invariant multivariable systems. *IEEE Trans Autom Control* 21(1):25–34
- de Carolis G, Galeani S, Sassano M (2020) Robust hybrid output regulation for linear systems with periodic jumps: the non-semiclassical case. *IEEE Control Syst Lett* 4:25–30
- Forni F, Teel AR, Zaccarian L (2013a) Follow the bouncing ball: global results on tracking and state estimation with impacts. *IEEE Trans Autom Control* 58(6):1470–1485
- Francis B, Wonham W (1976) The internal model principle of control theory. *Automatica* 12(5):457–465
- Galeani S, Menini L, Potini A, Tornambe A (2008) Trajectory tracking for a particle in elliptical billiards. *Int J Control* 81(2):189–213
- Galeani S, Menini L, Potini A (2012a) Robust trajectory tracking for a class of hybrid systems: an internal model principle approach. *IEEE Trans Autom Control* 57(2):344–359
- Goebel R, Sanfelice R, Teel A (2012b) Hybrid dynamical systems: modeling, stability, and robustness. Princeton University Press, Princeton
- Huang J (2004) Nonlinear output regulation: theory and applications, vol 8. Society for Industrial Mathematics, Philadelphia
- Marconi L, Teel AR (2010) A note about hybrid linear regulation. In: Conference on decision and control, Atlanta, pp 1540–1545
- Marconi L, Teel AR (2013) Internal model principle for linear systems with periodic state jumps. *IEEE Trans Autom Control* 58(11):2788–2802
- Morarescu I, Brogliato B (2010) Trajectory tracking control of multiconstraint complementarity lagrangian systems. *IEEE Trans Autom Control* 55(6):1300–1313
- Saberi A, Stoorvogel A, Sannuti P (2000) Control of linear systems with regulation and input constraints. Springer, London
- Trentelman HL, Stoorvogel AA, Hautus MLJ (2001) Control theory for linear systems. Springer, London

Parallel Robots

Frank C. Park
Robotics Laboratory, Seoul National University,
Seoul, Korea

Abstract

Parallel robots are closed chains consisting of a fixed and moving platform that are connected by a set of serial chain legs. Parallel robots typically possess both actuated and passive joints and may even be redundantly actuated. Although more structurally complex and possessing a smaller workspace, parallel robots are usually designed to exploit one or more of the natural advantages they possess over their serial counterparts, e.g., higher stiffness, increased positioning accuracy, and higher speeds and accelerations. In this chapter we provide an overview of the kinematic and dynamic modeling of parallel robots, a description of their singularity behavior, and basic methods developed for their control.

Keywords

Closed kinematic chain · Closed loop mechanism · Parallel manipulator

Introduction

A parallel robot refers to a kinematic chain in which a fixed platform and moving platform are connected to each other by several serial chains, or legs. The legs, which typically have the same kinematic structure, are connected to the fixed and moving platforms at points that are distributed in a geometrically symmetric fashion. The Stewart-Gough platform (Fig. 1) is a well-known example of a parallel robot: each of the six legs is a *UPS* structure (i.e., consisting of rigid links serially connected by a universal, prismatic, and spherical joint), with the prismatic joint actuated. Other examples of parallel robots include the $6 \times RUS$ platform of Fig. 2, the haptic interface device of Fig. 3, and the eclipse mechanism of Fig. 4.

Parallel robots can be regarded as a special class of closed chain mechanisms (i.e., chains that contain one or more closed loops) and are purposely designed to exploit the specific advantages afforded by the closed chain structure, e.g., for improved stiffness, greater positioning accuracy, or higher speed. Parallel robots should be distinguished from two or more cooperating serial robots that may form closed loops during execution of a task (e.g., a robotic hand grasping an object). Some of the fastest velocities and accelerations recorded by industrial robots have been achieved by parallel robots, primarily by

placing the actuators on the fixed platform and thereby minimizing the mass of the moving parts.

Many of the model-based techniques developed for the control of traditional serial chain robots are also applicable to a large class of parallel robots. On the other hand, kinematic and dynamic models for parallel robots are inherently more complex. Parallel robots also possess features not found in serial robots, e.g., passive joints, the possibility of redundant actuation, and a diverse range of singularity behavior, that need to be considered when designing a control law. We therefore begin with a brief

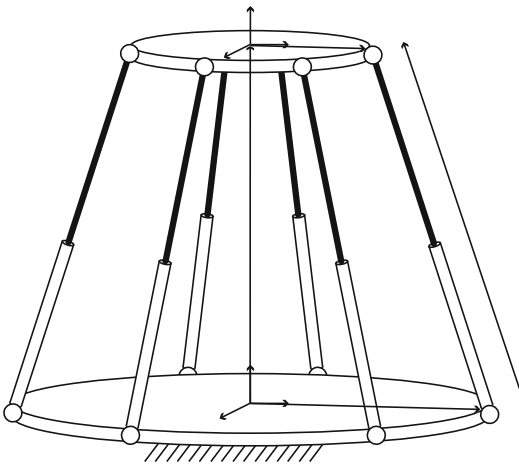
overview of the kinematic and dynamic modeling of parallel robots before discussing their control.

Modeling

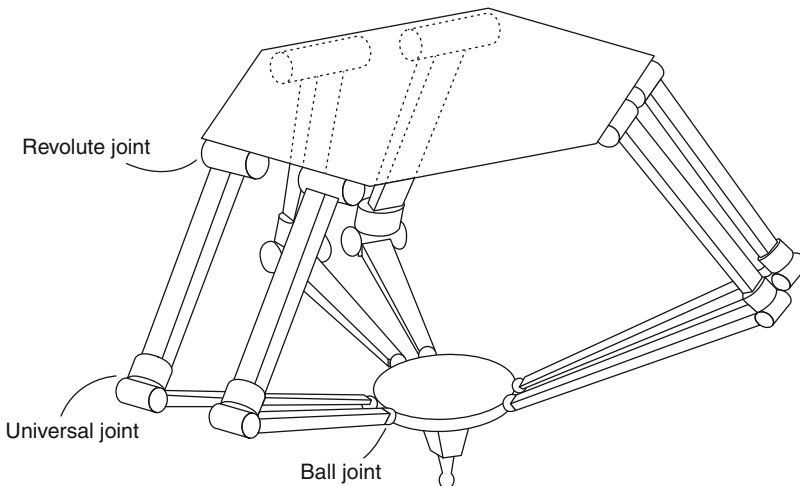
Kinematics

Whereas the kinematic degrees of freedom, or mobility, of a serial chain robot can be obtained as the sum of the degrees of freedom of each of the joints, the situation is somewhat more complex for parallel robots and closed chains in general, since only a subset of the joints can be independently actuated. The mobility of a parallel robot corresponds to the total degrees of freedom of the joints that can be independently actuated. In some cases the number of actuated joint degrees of freedom may exceed the kinematic degrees of freedom, in which case we say that the robot is redundantly actuated.

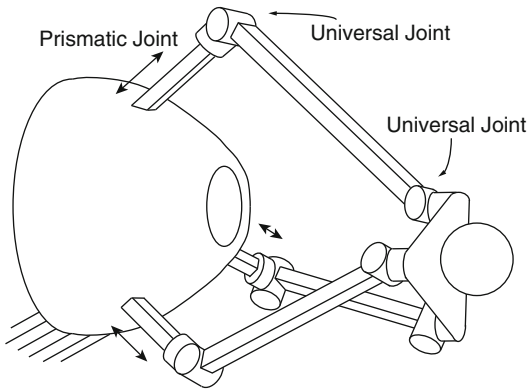
A parallel robot with a designated end-effector frame also has a notion of forward and inverse kinematics. While for serial chains the forward kinematics is a well-defined mapping and the inverse kinematics can typically have multiple solutions, for parallel robots the situation is less straightforward. For the Stewart-Gough platform of Fig. 1, in which the leg lengths can be adjusted by actuating the prismatic joints, the inverse



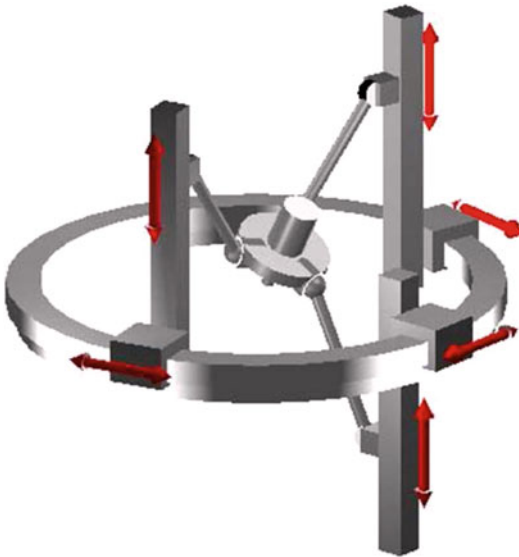
Parallel Robots, Fig. 1 Stewart-Gough platform



Parallel Robots, Fig. 2 $6 \times RUS$ platform



Parallel Robots, Fig. 3 A $3 \times PUU$ haptic interface



Parallel Robots, Fig. 4 The $3 \times PPRS$ eclipse parallel mechanism

kinematics is unique and straightforward to obtain, whereas the forward kinematics will have multiple solutions. For other types of parallel robots in which the legs themselves contain one or more closed loops, both the forward and inverse kinematics can have multiple solutions.

The notion of kinematic singularities for parallel robots is also much more involved than the case for serial robots. Whereas kinematic singularities for serial chain robots are characterized by configurations at which the forward kinematics Jacobian (i.e., the linear mapping relating joint velocities to end-effector frame velocities)

becomes singular, for parallel robots and closed chains in general, there exist other notions of singularities not found in serial chains. For example, given a parallel robot with kinematic mobility m – if the parallel robot consists only of one degree-of-freedom joints, this implies that exactly m joints can be actuated – there may exist configurations in which these m joints cannot be independently actuated. Conversely, even if the m actuated joints are each fixed to some value, the parallel robot may fail to be a structure, i.e., some of the links may be able to move.

In the above scenario, choosing a different set of m actuated joints may remedy this situation, in which case such singularities are referred to as actuator singularities. Configurations at which singularity behavior occurs regardless of which joints are actuated are denoted configuration singularities. The final class of singularities are end-effector singularities, which correspond to the usual serial chain notion of kinematic singularity, in which the end-effector loses one or more degrees of freedom of available motion.

Dynamics

In the case of a parallel robot whose actuated degrees of freedom coincides with its kinematic mobility m , it is possible to choose an independent set of generalized coordinates of dimension m , denoted $q \in \mathbb{R}^m$ and typically identified with the actuated joints, and to express the dynamics in the standard form

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau, \quad (1)$$

where $\tau \in \mathbb{R}^m$ denotes the vector of input joint torques, $M(q)$ denotes the $n \times n$ mass matrix, the matrix-vector product $C(q, \dot{q})\dot{q}$ denotes the vector of Coriolis terms, and $G(q) \in \mathbb{R}^m$ denotes the vector of gravitational forces. The structure of the dynamic equations is identical to that for serial chain robots. Also like the case for serial chain robots, the Coriolis matrix term $C(q, \dot{q}) \in \mathbb{R}^{m \times m}$ is not unique, so that one should ensure that the correct $C(q, \dot{q})$ is used in, e.g., any control law whose stability depends on the matrix $\dot{M}(q) - 2C(q, \dot{q})$ being skew-symmetric.

It is also important to keep in mind that the q must satisfy the kinematic constraint equations imposed by the loop closure constraints. That is, if $\theta \in \mathbb{R}^n$ denotes the vector of all joints (both actuated and passive), then $q \in \mathbb{R}^m$, $m \leq n$, will be a subset of θ whose values can only be obtained by solution of the kinematic constraint equations; depending on the nature of the kinematic constraints, one may have to resort to iterative numerical methods.

If the parallel robot is redundantly actuated, then the dynamics are subject to a further set of constraints on the input torques. Letting q_e denote the set of independent generalized coordinates and q_a be the vector of all actuated joints, the vector of actuated joint torques τ_a must then further satisfy $S^T \tau_a = W^T \tau$, where τ denotes the vector of joint torques for an equivalent tree structure system that moves identically to the redundantly actuated parallel robot and W and S are defined, respectively, by

$$S = \frac{\partial \theta}{\partial q_e}, \quad W = \frac{\partial q_a}{\partial q_e}. \quad (2)$$

Compared to the dynamics for serial chain robots, the dynamics for parallel robots is, in general, considerably more complex and computationally involved. The recursive algorithms that are available for computing the inverse and forward dynamics of serial chain robots can also be used to develop similar recursive algorithms for parallel robot dynamics; however, the computations will be considerably more involved and require multiple iterations.

Motion Control

Exactly Actuated Parallel Robots

For parallel robots whose actuated degrees of freedom match the kinematic mobility (this excludes the set of all redundantly actuated parallel robots), most control laws developed for serial chain robots are also applicable. This is not altogether surprising in light of the similarity in the structure of the kinematic and dynamic equations between serial and parallel robots.

Control laws for serial robots are also covered in this handbook, and we refer the reader to ► [Linear Matrix Inequality Techniques in Optimal Control](#) for the essential details. Here we summarize the most basic control laws and point out any additional computational or other requirements that are needed when applying these laws to parallel robots. Note that other control laws and techniques developed for serial chain robots, e.g., robust, sliding mode, can also be applied with the same additional considerations and requirements outlined below:

1. **Computed torque control:** Computed torque control for parallel robots has the same control law structure as for serial robots, i.e.,

$$\tau = M(q) (-K_p e - K_v \dot{e}) + \tau_{ff}, \quad (3)$$

where e denotes the tracking error, K_p and K_v are the proportional and derivative feedback gain matrices, and τ_{ff} denotes the feed-forward term required to cancel the nonlinear dynamics. Robust versions of computed torque control are also applicable to parallel robots under the same set of conditions, e.g., establishing appropriate bounds on the mass matrix eigenvalues and on the norm of the Coriolis matrix.

2. **Augmented PD control:** The augmented PD control law for serial robots is also applicable to parallel robots, i.e.,

$$\tau = -K_p e - K_v \dot{e} + M(q) \ddot{q}_d + C(q, \dot{q}) \dot{q} + G(q), \quad (4)$$

where q_d is the reference trajectory to be tracked and K_p and K_v are the proportional and derivative feedback gains. Asymptotic stability is also established under the same conditions.

3. **Adaptive control:** Because the dynamic equations for parallel robots are also linear in the link mass and inertial parameters, i.e.,

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \Phi(q, \dot{q}, \ddot{q}) p, \quad (5)$$

where p denotes the vector of link mass and inertial parameters, adaptive control laws developed for serial robots can also be used.

4. **Other control methods:** There exist numerous control methods developed for serial robots, e.g., task space or operational space control, sliding mode control, and various nonlinear control techniques; with few exceptions most of these algorithms can also be applied to exactly actuated parallel robots with minimal modification.

Redundantly Actuated Parallel Robots

As described earlier, parallel robots exhibit a much more diverse range of singularity behavior than their serial counterparts, many of which depend on the choice of actuated joints (actuator singularities). One way to eliminate actuator singularities is via redundant actuation, i.e., the total degrees of freedom of the actuated joints exceeds the kinematic mobility of the mechanism. Redundant actuation offers some protection in the event of failed actuators and, when combined with an appropriate control law, offers an effective means of reducing joint backlash, increasing speed and payload and stiffness, controlling compliance through the generation of internal forces, and even improving power efficiency (as an analogy, the human musculoskeletal system is redundantly actuated by antagonistic muscles). Of course, redundant actuation introduces a new set of control challenges, since the control inputs must be designed so as not to conflict with the kinematic constraints inherent in the parallel robot; loosely speaking, the actuated joints can no longer be independently controlled, since the consequences of unintended antagonistic actuation may be catastrophic.

The control of cooperating manipulators (see ► [Optimal Control and Mechanics](#)) has a long history in robotics, and many of the control techniques developed for such multi-arm systems can also be applied to redundantly actuated parallel robots. One can also apply the control strategies developed for exactly actuated parallel robots to the redundantly actuated case, but modifications are necessary to account for the different structure of the dynamic equations.

Like all model-based control algorithms, the above control laws are subject to model uncertainties. Whereas in serial chains the effects of model uncertainty simply lead to errors in tracking, for redundantly actuated parallel robots, the consequences can lead to internal forces in addition to end-effector tracking errors. Perhaps the most significant effect of any modeling errors is that, unlike the serial chain case, the kinematic errors can potentially alter the shape of the configuration space (recall that the configuration space will in general be a curved space for closed chains) and also interfere with any PD feedback introduced into the control. The development of control laws that are robust to such modeling errors and disturbances remains an open and ongoing area of research in parallel robot control.

Force Control

Both hybrid force-position control and impedance control are well-known and widely applied concepts in serial robots and can be extended in a straightforward manner to exactly actuated parallel robots. Recall that the basic feature of hybrid force-position control is that the task space is decomposed into force- and position-controlled directions, whereas in impedance control, the goal is have the robot maintain a certain desired spatial stiffness in the task space. Controllers that combine aspects of force-position and impedance control have also been proposed and developed for both serial robots and exactly actuated parallel robots. Modeling errors will cause deviations in both the force- and position-controlled directions – leading to motions in force-controlled directions and forces in position-controlled directions – which can be addressed by, e.g., a switching control strategy.

The problem of force control for redundantly actuated parallel robots, which encompasses both force-position and impedance control, has also received some attention in the literature. The main difference with the exactly actuated case is that internal forces can now be generated,

which requires a more detailed and coordinate-invariant examination of stiffness. Control methods that combine elements of force-position and impedance control for redundantly actuated parallel robots have received only limited attention in the literature.

Cross-References

- [Linear Matrix Inequality Techniques in Optimal Control](#)
- [Optimal Control and Mechanics](#)
- [Optimal Control via Factorization and Model Matching](#)
- [Optimal Sampled-Data Control](#)

Recommended Reading

The monograph (Merlet 2006) offers a detailed and comprehensive treatment of all aspects of parallel robots, with a particularly thorough treatment of the kinematics and singularity analysis. Mueller (2008) provides an excellent survey of the dynamics and control of redundantly actuated parallel robots and is based on the preceding work (Mueller 2005). Cheng et al. (2003) examines in detail the dynamic model for redundantly actuated parallel robots and the basic control strategies; Nakamura and Ghodoussi (1989) also examines dynamic models for redundantly actuated parallel robots. Stiffness analysis and control of redundantly actuated parallel robots are addressed in Yi and Freeman (1993), Chakarov (2004), and Fasse and Gosselin (1998). Analysis of specific parallel robots engaged in various control tasks includes Caccavale et al. (2003), Honegger et al. (1997), Kim et al. (2001), and Satya et al. (1995). The basic references on robot control are Murray et al. (1994), Spong et al. (2006), Anderson and Spong (1988), and Ghorbel (1995) focuses on PD control for closed chains.

Bibliography

- Anderson RJ, Spong MW (1988) Hybrid impedance control of robotic manipulators. *IEEE J Robot Autom* 4:549–556
- Caccavale F, Siciliano B, Villani L (2003) The Tricept robot: dynamics and impedance control. *IEEE/ASME Trans Mechatron* 8:263–268
- Chakarov D (2004) Study of the antagonistic stiffness of parallel manipulators with actuation redundancy. *Mech Mach Theory* 39:583–601
- Cheng H, Yiu Y-K, Li Z (2003) Dynamics and control of redundantly actuated parallel manipulators. *IEEE Trans Mechatron* 8(4):483–491
- Fasse ED, Gosselin CM (1998) On the spatial impedance control of Gough-Stewart platforms. In: *Proceedings of the IEEE international conference on robotics and automation*, Leuven, pp 1749–1754
- Ghorbel F (1995) Modeling and PD control of closed-chain mechanical systems. In: *Proceedings of the IEEE conference on decision and control*, New Orleans
- Honegger M, Codourey A, Burdet E (1997) Adaptive control of the Hexaglide, a 6-DOF parallel manipulator. In: *Proceedings of the IEEE international conference on robotics and automation*, Washington, DC, pp 543–548
- Kim J, Park FC, Ryu SJ, Kim J, Hwang JC, Park C, Iurascu CC (2001) Design and analysis of a redundantly actuated parallel mechanism for rapid machining. *IEEE Trans Robot Autom* 17(4):423–434
- Merlet JP (1988) Force feedback control of parallel manipulators. In: *Proceedings of the IEEE international conference on robotics automation*, Philadelphia, pp 1484–1489
- Merlet JP (2006) *Parallel robots*. Springer, Heidelberg
- Mueller A (2005) Internal prestress control of redundantly actuated parallel manipulators and its application to backlash avoiding control. *IEEE Trans Robot* 21(4):668–677
- Mueller A (2008) Redundant actuation of parallel manipulators. In: Wu H (ed) *Parallel manipulators: towards new applications*. I-Tech Publishing, Vienna
- Murray R, Li ZX, Sastry S (1994) *A mathematical introduction to robotic manipulation*. CRC Press, Boca Raton
- Nakamura Y, Ghodoussi M (1989) Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators. *IEEE Trans Robot Autom* 5(3):294–302
- Satya SM, Ferreira PM, Spong MW (1995) Hybrid control of a planar 3-DOF parallel manipulator for machining operations. *Trans N Am Manuf Res Inst/SME* 23:273–280
- Spong MW, Hutchinson S, Vidyasagar M (2006) *Robot modeling and control*. Wiley, New York
- Yi B-J, Freeman RA (1993) Geometric analysis of antagonistic stiffness in redundantly actuated parallel mechanisms. *J Robot Syst* 10:581–603

Particle Filters

Fredrik Gustafsson

Division of Automatic Control, Department of
Electrical Engineering, Linköping University,
Linköping, Sweden

Abstract

The particle filter computes a numeric approximation of the posterior distribution of the state trajectory in nonlinear filtering problems. This is done by generating random state trajectories and assigning a weight to them according to how well they predict the observations. The weights are instrumental in a resampling step, where trajectories are either kept or thrown away. This exposition will focus on explaining the main principles and the main theory in an intuitive way, illustrated with figures from a simple scalar example. A real-time application is used to graphically show how the particle filter solves a nontrivial nonlinear filtering problem.

Keywords

Estimation · Kalman filter · Nonlinear
filtering · Sequential Monte Carlo

Introduction

The particle filter computes an arbitrarily good solution to nonlinear filtering problems. The goal in nonlinear filtering is to compute the posterior distribution of the state vector in a dynamic model, given measurements that are related to the state. Bayes rule provides a recursive but computationally intractable solution. Monte Carlo (MC) methods can essentially solve all Bayesian inference problems. However, for nonlinear filtering, the complexity

increases exponentially in time. The MC approach would be to generate a large number of state trajectories (called particles) and their corresponding sequences of predicted measurements and then weighs together the trajectories according to how well the predicted and actual measurement sequences match each other.

With increasing time, the fit is deemed to be poor, since the state space increases exponentially in time. This is usually referred to as the depletion (or degeneracy) problem. The approach in the particle filter is to simulate only one step at the time and then resample the trajectories if needed. For this reason, the particle filter is sometimes referred to as a sequential Monte Carlo method. The resampling step keeps the trajectories that give a good fit, while the bad ones are discarded. The novel idea in the particle filter when it was first published in 1993 was the introduction of this resampling step.

Depletion is still a problem, despite the resampling step. Mitigating depletion has ever since the beginning been the most pressing issue in applied particle filtering. This tutorial will present the basic particle filter algorithm and discuss ways to avoid depletion problems both in general terms and in a simple example.

The particle filter computes an approximation to the Bayes optimal filter, conditioned on a sequence of observations and a nonlinear non-Gaussian system. It is important to note that the PF approximates the posterior distribution of the state trajectory, from which the mean and covariance are easily extracted. In contrast, the extended Kalman filter (EKF) computes the mean and covariance for an approximate dynamical system (linearized with Gaussian noise). The unscented Kalman filter (UKF) likewise also approximates the mean and covariance. Both EKF and UKF can only approximate unimodal (one peak) posterior distributions. There are filter bank approximations, like the interacting multiple model (IMM) algorithm, that can keep track of a given number of modes in the posterior. However, the PF does this in a more natural way.

The Basic Particle Filter

Nonlinear filtering aims at estimating the distribution of a state sequence $x_{1:N} = (x_1, x_2, \dots, x_N)$ from a sequence of observations $y_{1:N} = (y_1, y_2, \dots, y_N)$, given a state space model of the form

$$x_{k+1} = f(x_k, v_k) \quad \text{or} \quad p(x_{k+1}|x_k), \quad (1a)$$

$$y_k = h(x_k, e_k) \quad \text{or} \quad p(y_k|x_k). \quad (1b)$$

Here, v_k denotes process noise, and e_k is the measurement noise. The stochastic variables v_k, e_k for all k and x_0 are assumed mutually independent, with known distributions p_v, p_e , and p_{x_0} , which are all being part of the model specification.

The particle filter (PF) works with a set of random trajectories. Each trajectory is formed recursively by iteratively simulating the model with some randomness and then updating the likelihood of each trajectory based on the observation. In words, we first evaluate the set of particles at hand by comparing how well they predict the current observation. In this way, the particles are assigned a weight. We keep the particles with large weight and throw away the particles with small weight, using a stochastic resampling procedure. After this step, we get a smaller set of particles, where many particles have several replicas. We then simulate each particle to the next observation time using the dynamical model. After this prediction step, all particles will be unique (because they are based on different realizations of the process noise). Below, the basic algorithm (sometimes called bootstrap PF or sequential importance resampling (SIR) PF) is summarized.

- Define a set of random states (particles) by sampling $x_0^{(i)} \sim p_{x_0}(x_0)$.
- Iterate in $k = 0, 1, \dots$:

- 1 Measurement update: Compute the weight $\omega_k^{(i)} = p_e(y_k - h(x_k^{(i)}))$ and normalize so $\sum_{i=1}^N \omega_k^{(i)} = 1$.

- 2 Resampling: Resample each particle with probability $\omega_k^{(i)}$.
- 3 Time update: Simulate one time step by taking $v_k^{(i)} \sim p_v(v_k)$ and then set $x_{k+1}^{(i)} = f(x_k^{(i)}, v_k^{(i)})$.

The main design parameter here is the number N of particles. A common trick to make the filter more robust is to increase the variance of p_v and p_e above. This is called dithering (or jittering) and is a practical way to get more robust nonlinear filters.

To illustrate some of the aspects and for later reference, a simple example will be introduced.

Example: First-Order Linear Gaussian Model

The Kalman filter (KF) provides the posterior distribution in an analytical form for linear Gaussian models and is thus suitable for evaluations and comparisons. A linear Gaussian model looks like

$$x_{k+1} = Fx_k + v_k, \quad v_k \sim N(0, Q). \quad (2a)$$

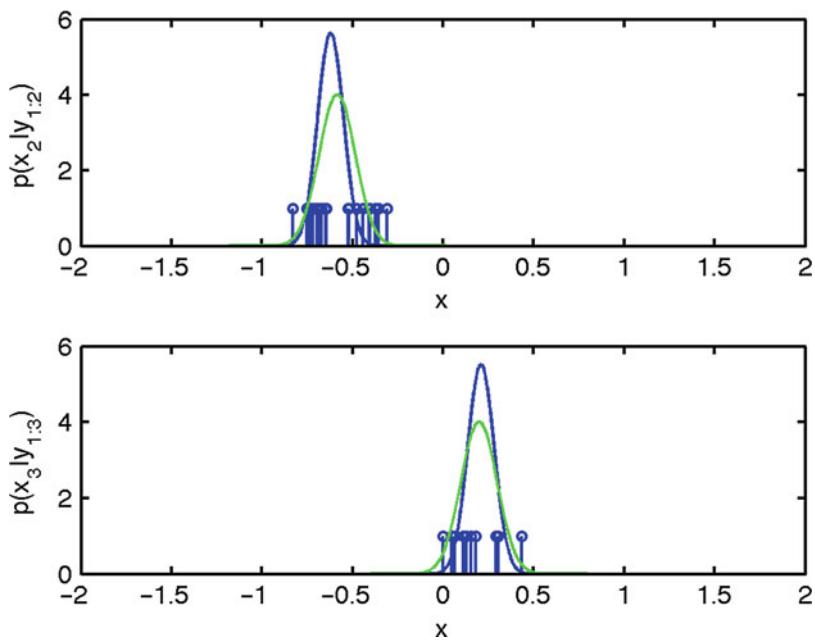
$$y_k = Hx_k + e_k, \quad e_k \sim N(0, R), \quad (2b)$$

$$x_0 \sim N(\mu_0, P_0), \quad (2c)$$

We will use the scalar case for the illustrations, and the figures that follow are based on $F = 0.9$, $H = 1$, $Q = 1$, $R = 0.01$, $P_0 = 1$. The particle filter in the scalar case simplifies to the Matlab algorithm in Table 1. Figure 1 compares the sample-based representation of the PF with the Gaussian distribution provided by the KF for the first two time steps. This shows how well the marginal distribution $p(x_k|y_{1:k})$ is approximated by the samples $x_k^{(i)}$ from the PF. A rule of thumb is that 30 samples are needed to approximate a univariate Gaussian distribution. As will be discussed later, the number of samples is effectively only 10 here, which explains the small deviation of the Gaussian functions.

Particle Filters, Table 1 Matlab code for scalar linear Gaussian model

```
% Simulation
y=filter([0 H],[1 -F],[sqrt(P0)*randn(1,1);...
        sqrt(Q)*randn(N-1,1)]+sqrt(R)*randn(N,1);
% Particle filter
x=mu0+sqrt(P0)*randn(N,1);
for k=1:K
    w=exp(-(y(k)-H*x).^2/R);           % Measurement update
    w=w/sum(w);                       % Normalization
    xhat(k)=w'*x;                     % Estimate
    P(k)=w'*(x-xhat(k)).^2;           % Variance
    x=resample(x,w);                  % Resampling
    x=F*x+sqrt(Q)*randn(N,1);         % Time update
end
```



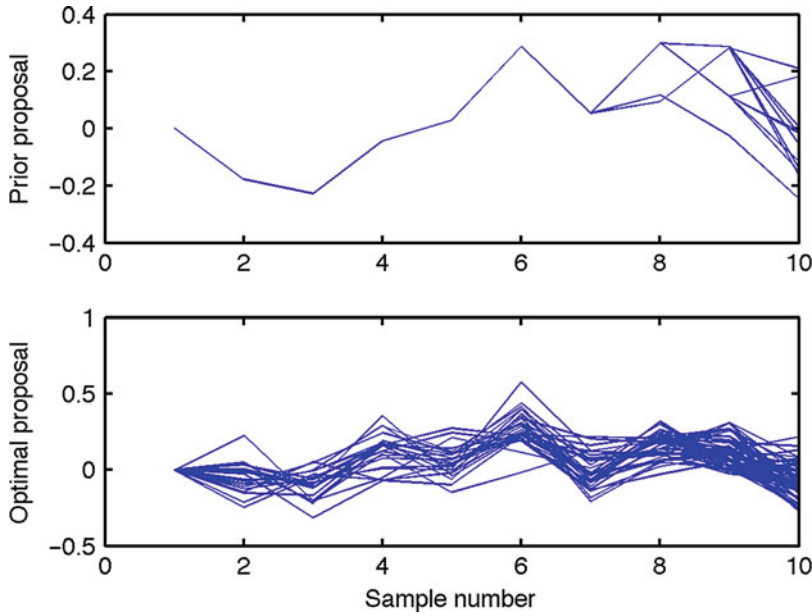
Particle Filters, Fig. 1 Set of samples $\{x_k^{(i)}\}_{i=1:N}$ compared to first a Gaussian approximation of the particles (blue) and second to the true posterior distribution provided by the Kalman filter (green)

To illustrate the fundamental depletion problem in the PF, the set of trajectories $x_{1:k}^{(i)}$ that approximates the posterior (smoothing) distribution $p(x_k | y_{1:k})$ is illustrated in Fig. 2. The upper plot shows a case where the trajectories are all the same initially. The behavior in the upper plot is typical for the basic particle filter in cases where the measurements are more informative than the state transition model (small measurement noise, $R < Q$). The lower plot shows a particle filter that is working better, and

the modification is explained in the following section.

Proposal Distributions

The time update in the basic PF predicts particles in step 4 according to the dynamic model. The most general derivation of the particle filter allows for sampling from a more general proposal (also called importance) distribution. This proposal distribution can be any function that



Particle Filters, Fig. 2 Set of trajectories $\{x_{1:10}^{(i)} - x_{1:10}\}_{i=1:N}$ for two different proposal distributions, one bad one (prior) leading to particle depletion and one good

one (likelihood). Note that the smoothing distribution $p(x_k | y_{1:10})$ can be approximated with the set of particles $\{x_k^{(i)}\}_{i=1:N}$ using the marginalization principle

can be sampled from, and it can depend on both the previous state and the current measurement. From a filtering perspective, it may appear as “cheating” to look at the next measurement when doing the time update, but one has to look at a full cycle of the iteration scheme.

If a proposal distribution of the functional form $q(x_k | x_{k-1}, y_k)$ is used, then steps 1 and 3 have to be modified as follows:

- 1 *Weight update:* Time and measurement updates:

$$w_{k|k-1}^{(i)} \propto w_{k-1|k-1}^{(i)} \frac{p(x_k^{(i)} | x_{k-1}^{(i)})}{q(x_k^{(i)} | x_{k-1}^{(i)}, y_k)}, \quad (3a)$$

$$w_{k|k}^{(i)} \propto w_{k|k-1}^{(i)} p(y_k | x_k^{(i)}). \quad (3b)$$

- 3 *Prediction:* Generate samples from the proposal

$$x_{k+1}^{(i)} \sim q(x_{k+1} | x_k^{(i)}, y_{k+1}) \quad (3c)$$

The most natural proposal distributions are the following:

- The prior $q(x_{k+1} | x_k^{(i)}, y_{k+1}) = p(x_{k+1} | x_k^{(i)})$, as used in the basic PF.
- The likelihood $q(x_{k+1} | x_k^{(i)}, y_{k+1}) \propto p(y_{k+1} | x_{k+1})$. For the model (2), the proposal becomes $N(y_k/h, R_k/h^2)$.
- The optimal (minimizing weight variance) choice $q(x_{k+1} | x_k^{(i)}, y_{k+1}) \propto p(y_{k+1} | x_{k+1}) p(x_{k+1} | x_k^{(i)})$. For the model (2), the optimal proposal is provided by one cycle of the Kalman filter, initialized with the particle $x_k^{(i)}$.

The optimal proposal keeps the weights constant, and this would in theory avoid depletion, where depletion is interpreted as excessive weight variance. Figure 2 compares the set of trajectories for the prior and likelihood proposals, respectively. Apparently, the likelihood proposal is to prefer here, since it suffers less from depletion in the particle history. The practical limitation with the last two alternatives is that one has to be able to sample from the likelihood, so in practice there

needs to be more measurements than states in the model.

Adaptive Resampling

Resampling is crucial to avoid depletion. Without resampling, all trajectories except for one will get zero weight quite quickly. However, there is no need to resample at every iteration. Actually, resampling increases the weight variance, which is undesired. The question is how to decide if resampling is needed. The efficient number of particles estimated as

$$N_{\text{eff}}(k) = \frac{1}{\sum_{i=1}^N (w_{k|k}^{(i)})^2}, \quad (4)$$

is one suitable indicator. If all particles have the same weight $w_{k|k}^{(i)} = 1/N$, then $N_{\text{eff}}(k) = N$. Conversely, if one weight is one and all other zero, then $N_{\text{eff}}(k) = 1$. Thus, N_{eff} can be interpreted as a measure of how many particles that actually contribute to the solution.

Figure 3 shows the evolution of $N_{\text{eff}}(k)$ for prior and likelihood proposals, respectively. With resampling in every iteration, the likelihood proposal performs very well with $N_{\text{eff}}(k) \approx N$, while the prior proposal effectively uses only 10 % of the particles. Thus, $N_{\text{eff}}(k)$ is a good indicator of the quality of the proposal distribution.

In Fig. 1, $N = 100$ so effectively 10 samples are contributing to the Gaussian approximation, which as mentioned before is too small a number to get a good result.

As a comparison, Fig. 3 also shows N_{eff} if resampling is never used, then $N_{\text{eff}}(k)$ normally decreases over time. The likelihood proposal does not decrease as fast as the prior proposal, and for this very short data sequence resampling is really not needed at all.

In summary, resampling increases weight variance and decreases the performance of the filter. On the other hand, without resampling the effective number of particles converges monotonously to only one. So, the idea of adaptive resampling is natural. The key idea is to resample only if

the effective number of particles is small. The usual rule of thumb is that resampling is needed if $N_{\text{eff}}(k) < 2N/3$.

Resampling was the main contribution in Gordon et al. (1993) to get a working algorithm.

Marginalization

The posterior distribution approximation provided by the particle filter converges with the number of particles. In theory, the convergence rate is g_k/N , where g_k is a polynomial function in time k . In practice, it appears that the required number of particles increases very quickly with state dimension. Unless a very good proposal distribution is found, the practical limit for the state dimension is around 3–4 as a rough rule of thumb.

For applications with a large number of states, one can in many cases still use the particle filter. The idea is to find a linear Gaussian substructure in the model and then divide the state vector x_k into two parts: x_k^l for the states that appear linearly and x_k^n for the remaining states. Bayes rule provides the factorization

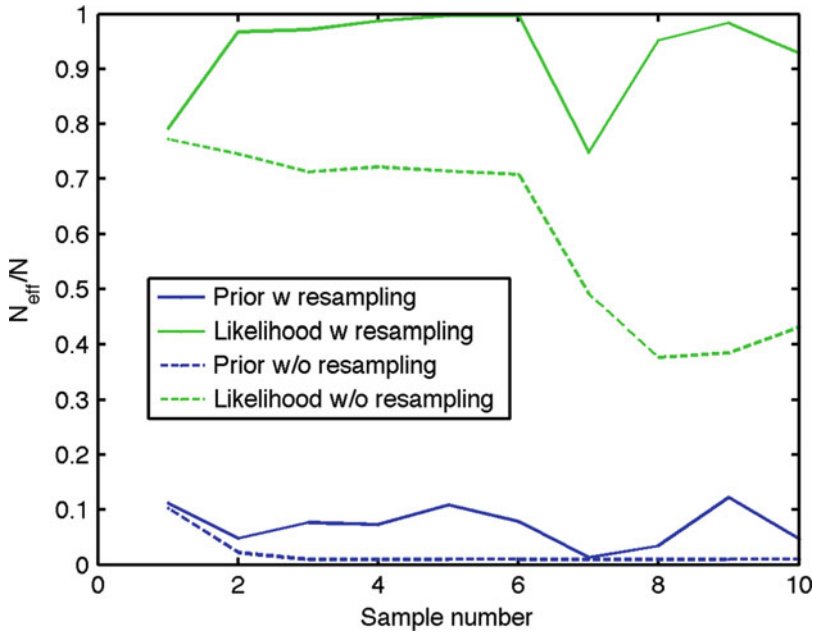
$$p(x_k^l, x_{1:k}^n | y_{1:k}) = p(x_k^l | x_{1:k}^n, y_{1:k}) p(x_{1:k}^n | y_{1:k}) \quad (5)$$

With a linear Gaussian substructure for x_k^l , given the whole trajectory $x_{1:k}^n$, then the Kalman filter applies and provides a Gaussian distribution for the first factor in (5). The second factor is resolved using a marginalization procedure so that the particle filter can be applied.

The bottom line is that each particle is associated with one Kalman filter. The method is called Rao-Blackwellized particle filter, or marginalized particle filter, in literature.

Illustrative Application: Navigation by Map Matching

A nontrivial application where the PF solves a nonlinear filtering problem, where Kalman filter-based approaches would fail, is described in



Particle Filters, Fig. 3 The efficient number of particles $N_{\text{eff}}(k)/N$ for prior proposal (blue) and likelihood proposal (green) ($N = 1,000$)

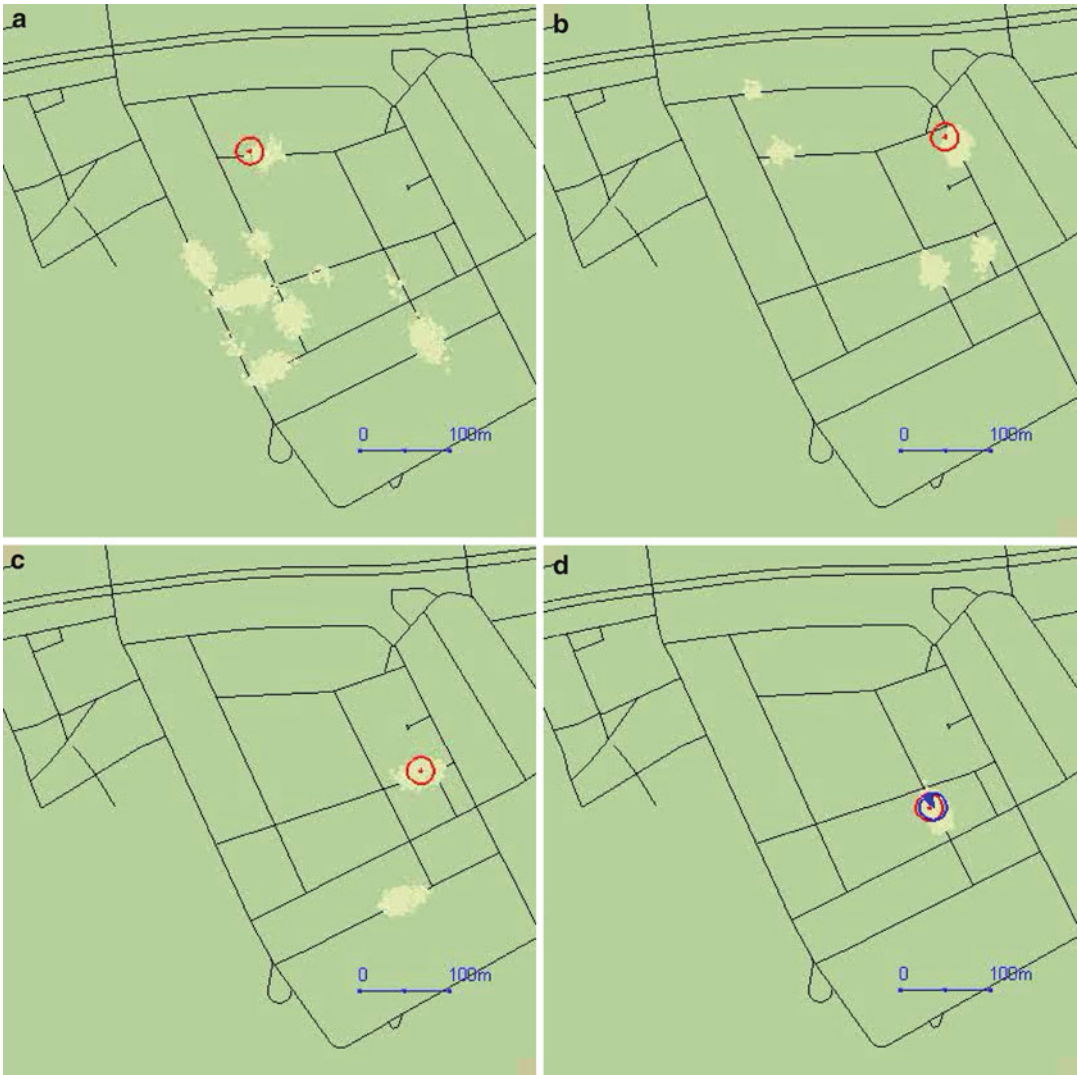
Forssell et al. (2002). The problem is to compute a robust estimate of the position of a car, without using infrastructure such as cellular networks or satellites. The approach is based on measuring wheel speeds on one axle and from that dead reckon a nominal trajectory using standard odometric formulas. A road map is then used as a measurement, to rule out impossible or unlikely maneuvers. There is no numeric measurement y_k in this approach, but the likelihood $p(y_k|x_k)$ in the model (1b) is large when x_k corresponds to a road position and decays quickly to zero outside the road network. Figure 4 illustrates the particle cloud gradually focuses around the true position with time. In particular, the number of modes in the posterior distributions is rather high initially, but decreases over time, in particular after each turn.

The particle filter is sometimes believed to be too computer intensive for real-time applications. As described in Forssell et al. (2002), this demonstrator implemented a particle filter on a pocket computer anno 2001 with $N = 15,000$ particles running in 10 Hz. Thus, the computational

complexity of the particle filter should not be overemphasized in practice.

Summary and Future Directions

The particle filter can be seen as a black-box solution to the nonlinear filtering problem, where any nonlinear dynamical model with arbitrary noise distributions can be plugged in. The main tuning parameter is the number of particles, and the PF will work in theory if this number is large enough. In practice, there are many tricks the user has to be aware of to mitigate the curse of dimensionality (depletion) that occurs for large state spaces (more than three) or long time sequences (more than a couple of samples). One engineering trick is dithering, to increase the variance in the involved noise distributions from their nominal values. More theoretical ways to mitigate depletion include clever choices of proposal distributions to sample from and marginalization (to solve a subset of the estimation problem with a Kalman filter).



Particle Filters, Fig. 4 Car navigation using wheel speed and a street map. The figures illustrate of how the particle representation of the position posterior distribution of position (course state is now shown) improves over time. After four turns, the posterior is essentially unimodal, and

a position marker can be shown. The *circle* denotes GPS position, which is only used for comparison. (a) After first turn. (b) After second turn. (c) After third turn. (d) After fourth turn

Current and future research directions include the issues above. A further trend concerns the related smoothing problem, which is interesting in itself, but which has turned out to be instrumental in joint state and parameter estimation problems. There are also many attempts to make the particle filter more robust, including ideas of filter banks and invoking a second layer of sampling algorithms to implement the proposal

distribution. There is also a trend to use the particle filter as a computational engine for more complex problems, such as the simultaneous localization and mapping (SLAM) problem, and to approximate the probability hypothesis density (PHD) for multi-sensor multi-target tracking. Finally, there are a large number of papers reporting on applications in traditional as well as new disciplines.

Cross-References

- ▶ [Estimation, Survey on](#)
- ▶ [Extended Kalman Filters](#)
- ▶ [Kalman Filters](#)
- ▶ [Nonlinear Filters](#)
- ▶ [Nonlinear System Identification Using Particle Filters](#)

Recommended Reading

Particle filtering (PF) as a research area started with the seminal paper (Gordon et al. 1993) and the independent developments in Kitagawa (1996) and Isard and Blake (1998). The state of the art is summarized in the article collection Doucet et al. (2001), the surveys Liu and Chen (1998), Arulampalam et al. (2002), Djuric et al. (2003), Cappé et al. (2007), Gustafsson (2010), and the monograph Ristic et al. (2004).

Bibliography

- Arulampalam S, Maskell S, Gordon N, Clapp T (2002) A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans Signal Process* 50(2):174–188
- Cappé O, Godsill SJ, Moulines E (2007) An overview of existing methods and recent advances in sequential Monte Carlo. *IEEE Proc* 95:899
- Djuric PM, Kotecha JH, Zhang J, Huang Y, Ghirmai T, Bugallo MF, Miguez J (2003) Particle filtering. *IEEE Signal Process Mag* 20:19
- Doucet A, de Freitas N, Gordon N (eds) (2001) *Sequential Monte Carlo methods in practice*. Springer, New York
- Forssell U, Hall P, Ahlqvist S, Gustafsson F (2002) Novel map-aided positioning system. In: *Proceedings of FISITA*, Helsinki, number F02-1131
- Gordon NJ, Salmond DJ, Smith AFM (1993) A novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc Radar Signal Process* 140:107–113
- Gustafsson F (2010) Particle filter theory and practice with positioning applications. *IEEE Trans Aerosp Electron Mag Part II Tutor* 7:53–82
- Isard M, Blake A (1998) Condensation – conditional density propagation for visual tracking. *Int J Comput Vis* 29(1):5–28
- Kitagawa G (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J Comput Graph Stat* 5(1):1–25

Liu JS, Chen R (1998) Sequential Monte Carlo methods for dynamic systems. *J Am Stat Assoc* 93:1032–1044

Ristic B, Arulampalam S, Gordon N (2004) *Beyond the Kalman filter: particle filters for tracking applications*. Artech House, London

Passivity and Passivity Indices for Switched and Cyber-physical Systems

Hasan Zakeri and Panos J. Antsaklis
Department of Electrical Engineering,
University of Notre Dame, Notre Dame,
IN, USA

Abstract

Cyber-physical systems (CPS), in which the interactions with the environment are enabled by computational and communication components, pose great challenges in design and analysis for control engineers. The operation of these systems is monitored, coordinated, controlled, and integrated by a computing and communication core interacting with the physical world. Given the large scale of such systems, it is important to develop tools that can guarantee system level properties, such as stability, from the properties of individual components and their interactions. Passivity and dissipativity are energy-like concepts and widely used in control design. They quantify the “energy” consumption of a dynamical system and have shown great promise in the design of cyber-physical systems because of their *compositionality* properties (Antsaklis et al., *Eur J Control* 19(5):379–388, 2013). Passivity indices are measures of passivity and extend the applicability of passivity-based tools to nonpassive systems as well. In the present entry, an overview of applications of passivity and passivity indices in the design of cyber-physical systems is given. Passiv-

ity indices and passivation methods are also discussed for linear, nonlinear, hybrid, and networked systems.

Keywords

Passivity · Passivity indices · Dissipativity · Switched systems · Hybrid systems · Cyber-physical systems · Passivation

Introduction

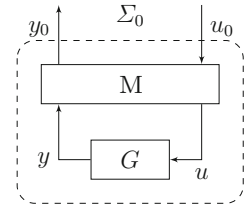
This entry extends the discussion in the companion entry *Passivity, Dissipativity, and Passivity Indices* to the cases of switched and cyber-physical systems. Passivity-based tools have been studied for linear and nonlinear systems for a long time, but their extension to switched and cyber-physical systems is subject of recent research. This extension provides tools to analysis and design complex systems in a simpler manner.

Definitions of passivity indices and their basic properties given in the companion entry are extended here. Specific values of the passivity indices of a system can be assigned via compensation and can be used in design approaches that involve not only passive but also nonpassive systems. A review of design tools based on passivation is also included.

Passivation

Passivity indices can be adjusted by introducing gains in series, feedback, or parallel configurations. A general method is given in Xia et al. (2014) using an input-output transformation matrix. Appropriate design of this matrix, called here *the M-matrix*, guarantees desired passivity levels for the system. This transformation matrix allows the use of a nonpassive controller to guarantee the passivity and stability of a feedback control system. Consider the system G and a general input-output transformation matrix M as shown in Fig. 1. The matrix M is invertible and defined as

Passivity and Passivity Indices for Switched and Cyber-physical Systems, Fig. 1 M -matrix interconnection



$$M \triangleq \begin{bmatrix} m_{11}I & m_{12}I \\ m_{21}I & m_{22}I \end{bmatrix} \quad (1)$$

where m_{ij} are real scalars. It can be shown that the passivity indices of the system $\Sigma_0 : u_0 \rightarrow y_0$, defined via

$$\begin{bmatrix} u_0 \\ y_0 \end{bmatrix} = M \begin{bmatrix} u \\ y \end{bmatrix},$$

depend on the gain γ of system G and the elements of M . Let G be finite gain stable with gain γ . Then the system Σ_0 is

1. *Passive*, if M in (1) is chosen such that

$$\begin{aligned} m_{11} &= m_{21}, & m_{22} &= -m_{21}, \\ m_{11} &\geq m_{22}\gamma > 0. \end{aligned} \quad (2)$$

2. *Output Feedback Passive (OFP)* with OFP index $\rho_0 = \frac{1}{2} \left(\frac{m_{11}}{m_{21}} + \frac{m_{12}}{m_{22}} \right) > 0$, if

$$m_{21} \geq m_{22}\gamma > 0, \quad m_{11}m_{22} > m_{12}m_{21} > 0. \quad (3)$$

3. *Input Feedforward Passive (IFP)* with IFP index $\nu_0 = \frac{1}{2} \left(\frac{m_{21}}{m_{11}} + \frac{m_{22}}{m_{12}} \right) > 0$, if

$$m_{11} \geq m_{12}\gamma > 0, \quad m_{12}m_{21} > m_{11}m_{22} > 0. \quad (4)$$

4. *Passive with passivity levels* $\delta_0 = \frac{1}{2} \frac{m_{11}}{m_{21}} > 0$ and $\epsilon_0 = \frac{a}{2} \frac{m_{21}}{m_{11}} > 0$, if

$$m_{11} > 0, \quad m_{12} = 0, \quad m_{21} \geq \frac{m_{22}\gamma}{\sqrt{1-a}} > 0, \quad (5)$$

where $0 < a < 1$ is an arbitrary real number.

As an example, consider the plant $H = \frac{s+1}{s-1}$ which is unstable and nonpassive with OFP index $\rho = -1$. The controller $C = \frac{1}{0.2s+1}e^{-0.1s}$ is

nonpassive as well but finite-gain stable with $\gamma = 1$ and IFP index $\nu = -0.293$. The loop in this configuration is unstable. This situation could represent an actual physical system controlled by a human operator modeled as controller C . To guarantee stability, based on the above passivation method, one can set $m_{11} = 0.5, m_{12} = 0.4, m_{21} = 50$, and $m_{22} = 20$ to make the controller, C , IFP and stabilize the closed-loop system. In this example, the matrix M acts as an interface to passivate C and is placed between C and G .

Passivity and Passivity Indices in Switched and Hybrid Systems

Passivity has been extended to switched systems. There are several definitions of passivity for switched systems, and they can be classified into two main approaches. In the first approach, arbitrary switching is considered, and each subsystem is assumed passive. This is an important assumption that guarantees passivity for the case of no switching. Aside from that, different definitions impose different assumptions on the system regarding the energy added due to switching. In the second approach, nonpassive subsystems are allowed under certain conditions.

The definition of passivity through a common storage function, while useful, might not cover all practical considerations for a system. The passivity for each subsystem may have its own physical meaning and its own storage function. Here, we will focus on passivity and dissipativity of switched systems with multiple storage functions (sometimes referred to as *decomposable dissipativity*).

Consider the switched system of the form

$$\begin{aligned}\dot{x} &= f_\sigma(x, u_\sigma), \\ y &= h_\sigma(x)\end{aligned}\quad (6)$$

where σ is the switching signal taking values in

$$M = \{1, 2, \dots, s\}$$

which may depend on time or the state, or on both, or be governed by a higher-level hybrid feedback in the loop. The switching signal σ

generates a switching sequence

$$\Sigma = \{x_0; (i_0, t_0), (i_1, t_1), \dots, (i_n, t_n), \dots \mid i_n \in M, n \in \mathbb{N}\} \quad (7)$$

in which t_0 and x_0 are the initial time and state, respectively. This sequence implies that when $t \in [t_k, t_{k+1})$, then $\sigma(t) = i_k$, that is, the i_k th system is active, or the system is in *mode* i . It is assumed that the state of the switched system (6) does not jump at switching instances.

In Zhao and Hill (2008), passivity for switched systems is defined and used to prove stability results. Specifically, it was shown that passive systems are Lyapunov stable, that negative feedback induces asymptotic stability, and that output strictly passive (OSP) systems are asymptotically stable. System (6) is said to be dissipative under the switching law Σ if there exist positive-definite continuous functions $V_1(x), V_2(x), \dots, V_s(x)$, called storage functions, locally integrable functions $w_i^j(u_i, h_i)$, $1 \leq i \leq s$, called supply rates, and locally integrable functions $w_j^i(x, u_i, h_i, t)$, $1 \leq i, j \leq s, i \neq j$, called cross-supply rates, such that the following conditions hold

1. Each subsystem i is dissipative with respect to w_i^i while active, i.e.,

$$\begin{aligned}V_{i_k}(x(t)) - V_{i_k}(x(r)) \\ \leq \int_r^t w_{i_k}^{i_k}(u(\tau), h_{i_k}(x(\tau))) d\tau, \\ t_k \leq r \leq t < t_{k+1}\end{aligned}$$

2. Each subsystem j is dissipative with respect to w_j^j when the i th subsystem is active, i.e., for $k = 0, 1, 2, \dots, t_k \leq r \leq t < t_{k+1}$,

$$\begin{aligned}V_j(x(t)) - V_j(x(r)) \leq \int_r^t w_j^{i_k}(x(\tau), u_{i_k}(\tau), \\ h_{i_k}(x(\tau)), \tau) d\tau,\end{aligned}\quad (8)$$

for $j \neq i_k, k = 0, 1, 2, \dots, t_k \leq r \leq t < t_{k+1}$.

3. And for any i, j , there exist $u_i(t) = \alpha_i(x(t), t)$ and $\phi_j^i(t) \in \mathcal{L}_1^+[0, \infty)$, which may depend on u_i and the switching sequence, such that

$$f_i(0, \alpha_i(0, t)) \equiv 0, \quad \forall t \geq t_0 \quad (9)$$

$$w_i^j(u_i(t), h_i(x(t))) \leq 0 \quad \forall t \geq t_0 \quad (10)$$

and

$$w_j^j(x(t), u_i(t), h_i(x(t)), t) - \phi_j^i(t) \leq 0, \\ \forall j \neq i, \forall t \geq t_0.$$

In this framework, because all subsystems share the same state variables, the storage functions of subsystems are still changing on the time intervals even when they are inactive. The active subsystem drives the state which, in turn, causes the change of the storage functions of inactive subsystems. This “changing” energy of any inactive subsystem, though not necessarily real energy, is viewed as “exported energy” from the active subsystem and is characterized by cross-supply rates. Unlike the classical definition of dissipativity, this definition requires positive definite storage functions to induce stability and output stabilization. Here, passivity and passivity indices are defined from earlier definitions of IF-OFP systems by setting

$$w_j^j(x, u_j, h_j, t) = u_j^T h_j - \delta_j u_j^T u_j - \epsilon_j h_j^T h_j, \\ j = 1, 2, \dots, s$$

for some δ_j, ϵ_j .

It is straightforward to define constant passivity indices using the above definition of passivity; however, it is restrictive for switched systems to have constant indices regardless of the active subsystem. The alternative is to redefine the indices to be time varying. As the switched system changes the active subsystem, the value of the indices changes accordingly. In the case of a constant index, if the subsystems have OFP index set $\{\rho_i\}$ (IFP index set $\{v_i\}$), then the switched system is OFP with index $\rho = \min\{\rho_i\}$ (IFP with index $v = \min\{v_i\}$). Formally, if each

subsystem i has constant OFP index ρ_i (or IFP index v_i), then the overall switched system is OFP with index $\rho(t) = \rho_\sigma$ (IFP with index $v(t) = v_\sigma$), where σ denotes the index of the active subsystem (McCourt and Antsaklis 2010).

For more general switched systems in which some modes can be nonpassive, passivity can be achieved in the presence of nonpassive modes as well. Since the dynamics of some modes are not passive, by classical definitions, the system is not passive. However, if the nonpassive modes are active infrequently enough, the system can still be passive. In such a situation, a nonpassive system can be rendered locally passive using feedback control laws if and only if its zero dynamics are locally passive. This definition is consistent with the traditional definition of passivity in the sense that useful properties, such as stability and compositionality, are preserved.

The M-matrix method can also be generalized to switched controllers consisting of N nonpassive subsystems. In this case, a switched controller is placed in feedback configuration of a nonpassive, unstable plant, and the passivation matrix M is also time varying and switches among constant values whenever the controller switches. Due to the complexity of the switched system, it might be difficult to switch among different passivation matrices, so a single passivation matrix can be designed to compensate for the lack of passivity in the system. Placing the passivated switched controller in the negative feedback interconnection of a nonpassive and unstable plant can make the closed-loop system passive and stable.

Passivity Indices in Networked Control Systems

Networks are an essential part of CPS, making possible the integration of many distributed components. The elements in a CPS are usually spatially distributed and communicate over a wired or wireless link. The use of communication links in large-scale systems brings many advantages including ease of maintenance, flexibility, scalability, and lower costs. However, employing

shared multipurpose communication networks, as opposed to traditionally dedicated connections, brings out many challenges caused by packet dropouts, signal quantization, time delays, fading channels, and limited information transfer.

Delays are the most common effects of communication networks, and it is known that delay can disrupt passivity. A wave variable transformation can be employed to compensate for the effects of delays when interconnecting passive systems. With some modification, wave variable transformations are also applicable to the case of networked switched systems where two switched systems connected in negative feedback communicate over a network with time delays. The main idea in this transformation is to treat the delayed network as a two-port network. The two-port network being passive ensures the passivity of the whole interconnection. The interconnection, including the transformations, is depicted in Fig. 2. The wave variable transformation is given as

$$\begin{aligned} \begin{bmatrix} u_1 \\ \hat{v}_1 \end{bmatrix} &= \frac{1}{\sqrt{2b}} \begin{bmatrix} I & bI \\ -I & bI \end{bmatrix} \begin{bmatrix} y_{2d} \\ y_1 \end{bmatrix} \\ \begin{bmatrix} \hat{u}_2 \\ \hat{v}_2 \end{bmatrix} &= \frac{1}{\sqrt{2b}} \begin{bmatrix} I & bI \\ -I & bI \end{bmatrix} \begin{bmatrix} y_2 \\ y_{1d} \end{bmatrix} \end{aligned} \quad (11)$$

where

$$\hat{u}_2 = u_1(t - T_2) = u_2, \quad \hat{v}_1 = v_2(t - T_1) = v_1 \quad (12)$$

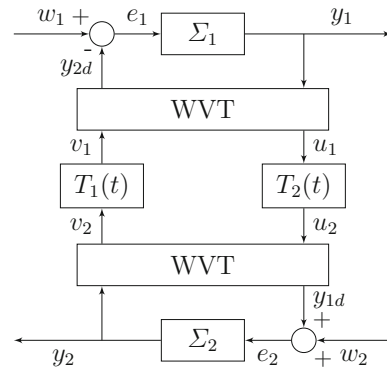
where the network delays T_1 and T_2 and the design parameter b are constant. For time-varying delays, the following modified transformation

$$\begin{aligned} \hat{u}_2 &= g_1(t)u_1(t - T_2(t)) = g_1(t)u_2(t) \\ \hat{v}_1 &= g_2(t)v_2(t - T_1(t)) = g_2(t)v_1(t) \end{aligned} \quad (13)$$

is used, where the design functions $g_1(t)$, $g_2(t)$ and delays $T_1(t)$, $T_2(t)$ satisfy

$$\begin{aligned} g_1^2(t) &\leq 1 - \frac{dT_2}{dt}, \quad g_2^2(t) \leq 1 - \frac{dT_1}{dt} \\ \frac{dT_1}{dt} &\leq 1, \quad \frac{dT_2}{dt} \leq 1 \end{aligned}$$

Similar transformations can be applied to the interconnection of two passive hybrid automata



Passivity and Passivity Indices for Switched and Cyber-physical Systems, Fig. 2 Two systems connected over a network with delays using wave variable transform

connected over a network with time-varying or constant delays to passivate, and under certain assumptions stabilize, the interconnection (Agarwal et al. 2016).

In the presence of packet dropouts, the networked control system is modeled as a discrete-time switched nonlinear system that switches between two modes: an uncontrolled mode in which the system evolves in open loop and a controlled mode in which a control input is applied to the system. If the ratio of the time steps for which the system evolves in open loop versus the time steps for which the system evolves in closed loop is bounded by a critical ratio, then the nonlinear system is locally passive.

Quantization is another widely seen phenomenon in digital controllers and communication channels. The control framework of Zhu et al. (2012) maintains passivity for switched and non-switched systems under quantization, based on the use of an input-output coordinate transformation to recover the passivity property.

Event-Triggered Networked Control Systems

Event-triggered control is an efficient network control method that limits the communication loads on the network and provides a balance between performance, communication

constraints, and control actions. In a typical event-triggered feedback framework, the information, whether it is a measurement or a control action, is transmitted only when necessary. Compared to time-driven control where a constant sampling period is applied to guarantee stability in the worst case scenario, the possibility of reducing the number of computations and thus of transmissions while guaranteeing desired levels of performance makes event-triggered control very appealing in networked control CPS. Passivity and passivity indices-based control combined with event-triggered networked control systems (NCS) provide a powerful tool for designing CPS.

When two systems are in feedback configuration with event-triggered communication in between, the passivity indices of the interconnection, as well as the finite gain $\mathcal{L}_2$ stability, relate to the passivity indices of each subsystem and the triggering condition. This relation is well documented in the literature (Yu and Antsaklis 2013). Based on this relation, the triggered control can be designed to achieve passive interconnections. This extends the compositionality properties of passivity to event-triggered systems.

Many of the challenges mentioned above for networked control systems that are not event triggered are also present in the event-triggered case as well. When network delays, quantization errors, and/or data loss are present in the event-triggered structure, passivity and passivity indices can be used to drive triggering conditions or to passivate and stabilize the system (Rahnama et al. 2018).

Further Readings

Basic definitions of dissipativity and passivity as well as their properties are given in Willems (1972). The discussion on passivity, particularly for nonlinear systems, is expanded in Hill and Moylan (1976) which lays the foundation for passivity indices. A review of applications of passivity in CPS is given in Antsaklis et al. (2013). The passivation method is based on Xia et al. (2014). Examples of applications of this method

to passivate a model of the human driver in a vehicle to guarantee passivity and performance exist in the literature as well (Xia et al. 2015).

Passivity indices of switched systems have various definitions. The definition we focused on here is from Zhao and Hill (2008). Passivity of switched systems with nonpassive subsystems is studied in Wang et al. (2014). The extensions to finite automata and hybrid systems can be found in Agarwal et al. (2017) and Haddad and Chellaboina (2001).

Several articles address passivity-based analysis and design of networked control system. For passivity of a NCS in the presence of packet dropouts, see Wang et al. (2015). The stability conditions for an event-triggered networked system based on the passivity properties of subsystems have been explored in Yu and Antsaklis (2011) and Zhu et al. (2017), with focus on passivity and passivation of event-triggered feedback interconnected systems of two input-feedforward, output-feedback passive systems. For a detailed illustration of the relationships among stability, robustness, and passivity levels of plant and controller, as well as an analysis of robustness against packet dropouts and passivity levels for the entire event-triggered networked control system, see Rahnama et al. (2018) and the references therein.

Cross-References

- [Event-Triggered and Self-Triggered Control](#)
- [Networked Control Systems: Architecture and Stability Issues](#)
- [Passivity-Based Control](#)
- [Passivity, Dissipativity, and Passivity Indices](#)
- [Stability Theory for Hybrid Dynamical Systems](#)

Bibliography

- Agarwal E, McCourt MJ, Antsaklis PJ (2016) Dissipativity of hybrid systems: feedback interconnections and networks. In: 2016 American control conference (ACC), Boston, pp 6060–6065
- Agarwal E, McCourt MJ, Antsaklis PJ (2017) Dissipativity of finite and hybrid automata: an overview. In:

- 2017 25th mediterranean conference on control and automation (MED), pp 1176–1182
- Antsaklis PJ, Goodwine B, Gupta V, McCourt MJ, Wang Y, Wu P, Xia M, Yu H, Zhu F (2013) Control of cyberphysical systems using passivity and dissipativity based methods. *Eur J Control* 19(5): 379–388
- Haddad WM, Chellaboina V (2001) Dissipativity theory and stability of feedback interconnections for hybrid dynamical systems. *Math Probl Eng* 7(4): 299–335
- Hill D, Moylan P (1976) The stability of nonlinear dissipative systems. *IEEE Trans Autom Control* 21(5): 708–711
- McCourt MJ, Antsaklis PJ (2010) Control design for switched systems using passivity indices. In: *Proceedings of the 2010 American control conference*, pp 2499–2504
- Rahnama A, Xia M, Antsaklis PJ (2018) Passivity-based design for event-triggered networked control systems. *IEEE Trans Autom Control* 63(9):3072–3077
- Wang Y, Gupta V, Antsaklis PJ (2014) On passivity of a class of discrete-time switched nonlinear systems. *IEEE Trans Autom Control* 59(3): 692–702
- Wang Y, Xia M, Gupta V, Antsaklis PJ (2015) On feedback passivity of discrete-time nonlinear networked control systems with packet drops. *IEEE Trans Autom Control* 60(9):2434–2439
- Willems JC (1972) Dissipative dynamical systems part I: general theory. *Arch Ration Mech Anal* 45(5): 321–351
- Xia M, Antsaklis PJ, Gupta V (2014) Passivity indices and passivation of systems with application to systems with input/output delay. In: *53rd IEEE conference on decision and control*, pp 783–788
- Xia M, Rahnama A, Wang S, Antsaklis PJ (2015) On guaranteeing passivity and performance with a human controller. In: *2015 23rd mediterranean conference on control and automation (MED)*, pp 722–727
- Yu H, Antsaklis PJ (2011) Event-triggered real-time scheduling for stabilization of passive and output feedback passive systems. In: *Proceedings of the 2011 American control conference*, pp 1674–1679
- Yu H, Antsaklis PJ (2013) Event-triggered output feedback control for networked control systems using passivity: achieving L2 stability in the presence of communication delays and signal quantization. *Automatica* 49(1):30–38
- Zhao J, Hill DJ (2008) Dissipativity theory for switched systems. *IEEE Trans Autom Control* 53(4):941–953
- Zhu F, Yu H, McCourt MJ, Antsaklis PJ (2012) Passivity and stability of switched systems under quantization. In: *Proceedings of the 15th ACM international conference on hybrid systems: computation and control, HSCC'12*. ACM, New York, pp 237–244
- Zhu F, Xia M, Antsaklis PJ (2017) On passivity analysis and passivation of event-triggered feedback systems using passivity indices. *IEEE Trans Autom Control* 62(3):1397–1402

Passivity, Dissipativity, and Passivity Indices

Hasan Zakeri and Panos J. Antsaklis
Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

Passivity and dissipativity are energy-like concepts, widely used in control design, that capture the “energy” consumption of a dynamical system and therefore relate closely to the physical world. Passivity indices of a system are measures of its passivity margins and represent shortage and excess of passivity in a system. With the aid of passivity indices, one can measure how passive a system is, or how far from passivity it is. Passivity indices extend all the analysis and design methods based on passivity to nonpassive systems as well. One of the advantages of using passivity is its tight relationship to stability. Another is its compositionality, which, together with its generality, makes it possible to use passivity in a wide range of complex control systems. In the present entry, an overview of dissipativity and passivity is given. Passivity indices of a system and their relation to stability are defined, and methods to find the indices are presented.

Keywords

Passivity · Passivity indices · Dissipativity · Stability · Energy

Passivity and Passivity Indices

Consider a continuous-time dynamical system $\mathbf{H} : u \rightarrow y$, where $u \in \mathcal{U} \subseteq \mathbb{R}^m$ denotes the input and $y \in \mathcal{Y} \subseteq \mathbb{R}^p$ denotes the corresponding output. Let the system be described by the state space equations:

$$\begin{aligned}\dot{x} &= f(x, u) \\ y &= h(x, u),\end{aligned}\quad (1)$$

where $f(\cdot, \cdot)$ and $h(\cdot, \cdot)$ are Lipschitz mappings of proper dimensions, $x \in \mathbb{R}^n$, and assume the origin is an equilibrium point of the system; i.e., $f(0, 0) = 0$ and $h(0, 0) = 0$. The more general concept of dissipativity is defined first. The system (1) is called *dissipative with respect to the supply rate function* $w(u, y)$, if there exists a nonnegative real-valued scalar function $V(x)$, called the *storage function*, such that $V(0) = 0$ and for all $x_0 \in \mathcal{X}$, all $t_1 \geq t_0$, and all $u \in \mathbb{R}^m$

$$V(x(t_1)) \leq V(x(t_0)) + \int_{t_0}^{t_1} w(u(t), y(t)) dt. \quad (2)$$

where $x(t_0) = x_0$ and $x(t_1)$ is the state at t_1 resulting from initial condition x_0 and input function $u(\cdot)$ (Willems 1972). If $V(x)$ is differentiable, then condition (2) is equivalent to

$$\dot{V}(x) \triangleq \frac{\partial V}{\partial x} \cdot f(x, u) \leq w(u(t), y(t)). \quad (3)$$

If (2) is satisfied with

$$w(u, y) = y^T Q y + 2y^T S u + u^T R u \quad (4)$$

then the system is called QSR-dissipative. If (2) is satisfied with supply rate $w(u, y) = u^T y$, then the system is called passive. The storage function can be seen as a generalization of the energy stored in the system, and thus, the dissipation inequality (2) implies that the energy stored in the system at every moment, described by the storage function, is less than or equal to the energy supplied to the system plus the initially stored energy. The notion of energy here is a generalization of the physical energy, though, in many physical systems, the two can be the same.

A related definition of passivity that involves only input-output maps is given below. A system $\mathcal{H} : u \mapsto y$ is considered passive, if for every $u \in \mathcal{U}$, we have $\langle u, \mathcal{H}u \rangle \geq \beta$, for some $\beta \leq 0$ (which its value represents the initially stored energy). In the case of continuous-time

systems, the inner product $\langle \cdot, \cdot \rangle$ is defined as $\langle u(t), y(t) \rangle = \lim_{T \rightarrow \infty} \int_0^T u(t)^T y(t) dt$ (where the integral exists). For more details, as well as the existence of the inner product, see Hill and Moylan (1980) and van der Schaft (2017).

Passivity can be defined for a variety of systems, including linear, nonlinear, hybrid, and continuous-time or discrete-time systems. Passivity is particularly appealing in cyber-physical systems (CPS) design because it applies to heterogeneous systems and also it is preserved when systems are combined in parallel or feedback (Bao and Lee 2007). Passivity indices are introduced as measures of passivity, and they extend passivity-based tools to nonpassive systems as well.

System (1) is called *input feedforward passive (IFP)* if it is dissipative with respect to supply rate function $w(u, y) = u^T y - \nu u^T u$ for some $\nu \in \mathbb{R}$. The *input feedforward passivity index* for system (1) is the largest ν for which the system is IFP. If $\nu > 0$, the system is called input strictly passive (ISP). IFP index is also the largest gain that can be applied to the system via a negative feedforward interconnection such that the overall system is passive. For an input-output map $\mathcal{H}$, the IFP index is defined as the largest ν such that $\langle u, \mathcal{H}u \rangle \geq \beta + \nu \langle u, u \rangle$ for some $\beta \leq 0$.

System (1) is called *Output Feedback Passive (OFP)* if it is dissipative with respect to supply rate function $w(u, y) = u^T y - \rho y^T y$ for some $\rho \in \mathbb{R}$. The *output feedback passivity index* for system (1) is the largest ρ for which the system is OFP. If $\rho > 0$, the system is called output strictly passive (OSP). The OSP index is also the largest gain that can be applied to the system via a positive feedback interconnection such that the overall system is passive. For an input-output map $\mathcal{H}$, the OFP index is defined as the largest ρ such that $\langle u, \mathcal{H}u \rangle \geq \beta + \rho \langle \mathcal{H}u, \mathcal{H}u \rangle$ for $\beta \leq 0$.

System (1) is said to have IF-OFP levels (ε, δ) , if it is dissipative with respect to supply rate function $w(u, y) = (1 + \varepsilon\delta)u^T y - \delta y^T y - \varepsilon u^T u$ for some $\varepsilon, \delta \in \mathbb{R}$. Note that passivity indices ρ and ν are unique, but that is not the case for ε and δ . In an alternative definition, this system is said to have passivity levels

(ε', δ') , if it is dissipative with respect to supply rate function $w(u, y) = u^\top y - \delta' y^\top y - \varepsilon' u^\top u$ for some $\varepsilon', \delta' \in \mathbb{R}$. If $\varepsilon' > 0$ and $\delta' > 0$, the system is called very strictly passive (VSP).

It should be noted that although the emphasis here is on continuous-time systems, passivity and passivity indices are defined similarly for discrete-time systems.

Stability

Passivity and passivity indices have strong relations to stability. It is known that if the system (1) is passive with a positive definite storage function $V(x)$, then the autonomous system (with $u = 0$) is Lyapunov stable. Output strictly passive systems are $\mathcal{L}_2$ stable. Moreover, if system G is strictly passive with output passivity index ρ , then G is finite-gain $\mathcal{L}_2$ stable with gain $\gamma \leq \frac{1}{\rho}$.

Consider the feedback interconnection of Fig. 1, and suppose that each feedback component G_i has passivity levels $(\varepsilon'_i, \delta'_i)$. Then, the closed-loop map is finite gain $\mathcal{L}_2$ stable if $\varepsilon'_1 + \delta'_2 > 0$ and $\varepsilon'_2 + \delta'_1 > 0$. If each system has IF-OSP indices $(\varepsilon_i, \delta_i)$, then the interconnection is $\mathcal{L}_2$ stable if the following matrix is positive definite:

$$A = \begin{bmatrix} (\delta_1 + \varepsilon_2)I & \frac{1}{2}(\delta_1\varepsilon_1 - \delta_2\varepsilon_2)I \\ \frac{1}{2}(\delta_1\varepsilon_1 - \delta_2\varepsilon_2)I & (\delta_2 + \varepsilon_1)I \end{bmatrix} > 0 \quad (5)$$

Passivity in Linear Systems

In linear time-invariant (LTI) systems, there are strong relations among passivity and passivity indices, positive real and strictly positive

real systems, and the Kalman-Yakubovich-Popov lemma. A continuous-time LTI system, represented by an $m \times m$ rational and proper transfer function matrix $H(s)$, is positive real (PR) if the following conditions are satisfied:

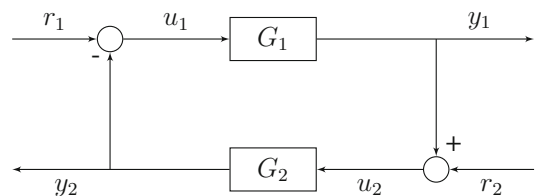
- (i) All elements of $H(s)$ are analytic in $\text{Re}[s] > 0$.
- (ii) $H(s)$ is real for all real positive values of s .
- (iii) $H^\top(s^*) + H(s) \geq 0$ for $\text{Re}[s] > 0$, where s^* is the complex conjugate of s .

Furthermore, $H(s)$ is strictly positive real (SPR) if $\exists \sigma > 0$ such that $H(s - \sigma)$ is positive real. Finally, $H(s)$ is strongly positive real if $H(s)$ is SPR and $D + D^\top > 0$ where $D := \lim_{s \rightarrow \infty} H(s)$.

The definition of PR implies that the poles of $H(s)$ are in the closed left-half plane; therefore any minimal realization of $H(s)$ is Lyapunov stable. Strictly positive real implies that the poles of $H(s)$ are in the open left-half plane; therefore the system is $\mathcal{L}_2$ stable with an asymptotically stable minimal realization. Positive realness and strictly positive realness can be verified directly in the frequency domain. For LTI systems, the property of passivity is equivalent to the property of positive realness. Under mild technical assumptions, these systems are Lyapunov stable. For LTI systems, strict passivity is equivalent to strict positive realness. For asymptotically stable systems, strongly positive real is ISP. While OSP systems are passive and $\mathcal{L}_2$ stable, it should be noted that this relationship is sufficient only: systems that are passive and $\mathcal{L}_2$ stable are not necessarily OSP (see Kottenstette et al. 2014, Theorem 8 for constructive examples of such systems). It can also be stated that if a minimal realization of a transfer function is VSP, then the transfer function is strongly positive real.

Passivity, Dissipativity, and Passivity Indices,

Fig. 1 The general feedback interconnection of two dynamical systems G_1 and G_2



Computation of Passivity Indices

The use of linear matrix inequalities (LMIs) in passivity theory comes from earlier work on the positive real lemma, also known as the KYP (Kalman, Yakubovich, and Popov) Lemma. A linear time-invariant system is passive if and only if it is passive with a quadratic energy storage function (Khalil 2002). Therefore, a linear system described by

$$\begin{aligned}\dot{x} &= Ax + Bu, \\ y &= Cx + Du\end{aligned}\quad (6)$$

is passive if and only if there exists a positive-definite matrix P such that

$$\begin{aligned}\max \quad & \rho \\ \text{s.t.} \quad & P > 0, \\ & \begin{bmatrix} A^\top P + PA + \rho C^\top C & PB - 0.5C^\top + \rho C^\top D \\ B^\top P - 0.5C + \rho D^\top C & PB - 0.5C^\top + \rho C^\top D + \rho D^\top D - 0.5(D + D^\top) \end{bmatrix} \leq 0.\end{aligned}\quad (9)$$

The passivity indices for a linear system can also be found in the frequency domain. The IFP index for a stable linear system $G(s)$ can be computed as

$$\nu = \frac{1}{2} \min_{\omega \in \mathbb{R}} \underline{\lambda}(G(j\omega) + G^*(j\omega)), \quad (10)$$

where $\underline{\lambda}$ denotes the minimum eigenvalue. If the system $G(s)$ is minimum phase, then its OFP index is

$$\rho = \frac{1}{2} \min_{\omega \in \mathbb{R}} \underline{\lambda}(G^{-1}(j\omega) + [G^{-1}(j\omega)]^*), \quad (11)$$

where $G^{-1}(s)$ denotes the inverse of system $G(s)$.

The transfer function $G(s)$ may be rational or irrational. In the case of a single-input single-output (SISO) system, we can test the passivity of $G(s)$ using the Nyquist plot. If the Nyquist plot of $G(s)$ is in the closed right-hand half of the complex plane, then the system is passive; otherwise, the system is not passive. Similarly, IFP index ν is the abscissa of the leftmost point

$$\begin{bmatrix} A^\top P + PA & PB - C^\top \\ B^\top P - C & -D - D^\top \end{bmatrix} \leq 0. \quad (7)$$

Passivity indices can be found using similar LMIs. The IFP index ν is the solution to the following optimization problem

$$\begin{aligned}\max \quad & \nu \\ \text{s.t.} \quad & P > 0, \\ & \begin{bmatrix} A^\top P + PA & PB - 0.5C^\top \\ B^\top P - 0.5C & -0.5(D + D^\top) - \nu I \end{bmatrix} \leq 0.\end{aligned}\quad (8)$$

Similarly, OFP index ρ is solution to the following optimization

on the Nyquist diagram of $G(s)$ and the OFP index ρ is the abscissa of the leftmost point on the Nyquist diagram of $G^{-1}(s)$.

Defining the OFP index as mentioned above imposes a minimum phase condition on the system. Now consider a SISO system with transfer function $H(s)$ and corresponding frequency response $H(j\omega)$. If H is OSP, then its OFP index ρ satisfies the following inequality

$$0 < \rho < \inf_{\omega \in [0, \infty)} \frac{\Re[H(j\omega)]}{\Re[H(j\omega)^2 + \Im[H(j\omega)]^2]}. \quad (12)$$

There are sum of squares (SOS) methods to determine the passivity indices of nonlinear systems (Zakeri and Antsaklis 2019b). There are also relations between passivity indices of a nonlinear system and its approximations, e.g., linearization (Xia et al. 2015). The indices for switched and networked control systems are discussed in a companion entry titled “► [Passivity and Passivity Indices for Switched and Cyber-Physical Systems](#)”.

Further Readings

Basic definitions of dissipativity and passivity as well as their properties are given in Willems (1972). The discussion on passivity, particularly for nonlinear systems, is expanded in Hill and Moylan (1976) laying the foundation for passivity indices. For more details on the relation of passivity and passivity indices with stability, see Khalil (2002) and Haddad and Chellaboina (2008). A detailed discussion on the relation of passivity, positive realness, and dissipativity is presented in Kottenstette et al. (2014). Data-driven methods for approximating passivity indices have also been introduced, as well as application of such approximations in fault and attack mitigation. See Romer et al. (2018) for an offline iterative method and Zakeri and Antsaklis (2019a) for online estimation and adaptive control reconfiguration method. A review of applications of passivity in CPS is given in Antsaklis et al. (2013) and Zakeri and Antsaklis (2019c).

Cross-References

- [KYP Lemma and Generalizations/Applications](#)
- [Lyapunov's Stability Theory](#)
- [Nonlinear Zero Dynamics](#)
- [Passivity and Passivity Indices for Switched and Cyber-Physical Systems](#)
- [Passivity-Based Control](#)
- [Stability: Lyapunov, Linear Systems](#)

Bibliography

- Antsaklis PJ, Goodwine B, Gupta V, McCourt MJ, Wang Y, Wu P, Xia M, Yu H, Zhu F (2013) Control of cyberphysical systems using passivity and dissipativity based methods. *Eur J Control* 19(5):379–388
- Bao J, Lee PL (2007) *Process control: the passive systems approach*. Advances in industrial control. Springer, London
- Haddad WM, Chellaboina V (2008) *Nonlinear dynamical systems and control: a Lyapunov-based approach*. Princeton University Press, Princeton
- Hill D, Moylan P (1976) The stability of nonlinear dissipative systems. *IEEE Trans Autom Control* 21(5):708–711
- Hill DJ, Moylan PJ (1980) Dissipative dynamical systems: basic input-output and state properties. *J Frankl Inst* 309(5):327–357
- Khalil HK (2002) *Nonlinear systems*, 3rd edn. Prentice Hall, Upper Saddle River
- Kottenstette N, McCourt MJ, Xia M, Gupta V, Antsaklis PJ (2014) On relationships among passivity, positive realness, and dissipativity in linear systems. *Automatica* 50(4):1003–1016
- Romer A, Montenbruck JM, Allgöwer F (2018) Some ideas on sampling strategies for data-driven inference of passivity properties for MIMO systems. In: *American Control Conference (ACC)*, 2018, Milwaukee, pp 6094–6100
- van der Schaft A (2017) *L2-gain and passivity techniques in nonlinear control*. Communications and control engineering. Springer International Publishing, Cham
- Willems JC (1972) Dissipative dynamical systems part I: general theory. *Arch Ration Mech Anal* 45(5):321–351
- Xia M, Antsaklis PJ, Gupta V, McCourt MJ (2015) Determining passivity using linearization for systems with feedthrough terms. *IEEE Trans Autom Control* 60(9):2536–2541
- Zakeri H, Antsaklis PJ (2019a) A data-driven adaptive controller reconfiguration for fault mitigation: a passivity approach. In: *2019 27th Mediterranean Conference on Control and Automation (MED)*, pp 25–30
- Zakeri H, Antsaklis PJ (2019b) Passivity and passivity indices of nonlinear systems under operational limitations using approximations. *arXiv:190100852 [cs]* 1901.00852
- Zakeri H, Antsaklis PJ (2019c) Recent advances in analysis and design of cyber-physical systems using passivity indices. In: *2019 27th Mediterranean Conference on Control and Automation (MED)*, pp 31–36

Passivity-Based Control

Romeo Ortega¹ and Pablo Borja²

¹Laboratoire des Signaux et Systèmes, Centrale Supélec, CNRS, Plateau de Moulon, Gif-sur-Yvette, France

²Faculty of Science and Engineering – Engineering and Technology Institute Groningen, University of Groningen, Groningen, The Netherlands

Abstract

Stabilization of physical systems by shaping their energy function is a well-established technique whose roots date back to the

work of Lagrange and Legendre. Potential energy shaping for fully actuated mechanical systems was first introduced in Takegaki and Arimoto (Trans ASME J Dyn Syst Meas Control 12:119–125, 1981) more than 30 years ago. In Ortega and Spong (Automatica 25(6):877–888, 1989) it was proved that passivity was the key property underlying the stabilization mechanism of these designs, and the, now widely popular, term of passivity-based control was coined. In this chapter we summarize the basic principles and some of the main developments of this controller design technique.

Keywords

Stabilization · Lyapunov function · Energy shaping · Storage function · Dissipation

Introduction

Energy is one of the fundamental concepts in science and engineering practice, where it is common to view dynamical systems as energy transformation devices. This perspective is particularly useful in studying *complex nonlinear* systems by decomposing them into simpler subsystems which, upon interconnection, add up their energies to determine the full system's behavior. The action of a controller may be also understood in energy terms as another dynamical system – typically implemented in a computer – interconnected with the process to modify its behavior. Then, the control problem can be recast as finding a dynamical system and an interconnection pattern such that the overall energy and dissipation functions take the desired form. This “energy-shaping plus dissipation” approach is the essence of the controller design technique – known as passivity-based control (PBC) – that is reviewed in this chapter.

We consider dynamical systems represented as multiports with port variables $(u, y) \in \mathbb{R}^m \times \mathbb{R}^m$. In physical systems, the port variables are conjugated, in the sense that their product represents power. In this case, passivity captures the

well-known property that the energy that can be extracted from the system is bounded, that is, there exists $\beta \in \mathbb{R}$ such that

$$-\int_0^t u^\top(s)y(s)ds \leq \beta.$$

Motivated by practical applications, we assume that the system admits a state-space representation given in the input-state-output form

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x),\end{aligned}\tag{1}$$

where $x \in \mathbb{R}^n$ is the state, and we view the system as a mapping $\Sigma : u \mapsto y$. It is said to be passive if there exists a function $H(x) \geq 0$ – called the storage function – such that the power-balance inequality

$$\dot{H} \leq u^\top y\tag{2}$$

holds. In physical systems a reasonable candidate for $H(x)$ is the energy of its energy-storing elements, e.g., capacitors, inductors, dampers, and masses.

An algebraic description of passive systems of the form (1) is given in Hill and Moylan (1980), where it is shown that a necessary condition for a system to be passive is that

$$h(x) = g^\top(x)\nabla H(x).\tag{3}$$

A geometric characterization of systems that can be rendered passive via state feedback is reported in Byrnes et al. (1991).

The objective of PBC, in the simplest static state-feedback formulation, is to find a function $\hat{u} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ such that the system (1) in closed-loop with the control law $u = \hat{u}(x) + v$ satisfies the new power balance inequality

$$\dot{H}_d \leq v^\top y_d,\tag{4}$$

where $(v, y_d) \in \mathbb{R}^m \times \mathbb{R}^m$ are the new port variables and $H_d(x) \geq 0$ is the *desired* storage function. This first step of PBC is known as *energy shaping*. The second step in PBC, known

as *damping injection*, is to set $v = -K_{\text{DI}}y_d$, with $K_{\text{DI}} > 0$, then $\dot{H}_d \leq -y_d^\top K_{\text{DI}}y_d$.

Chapter notation All mappings are supposed smooth. Consider the mapping $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$. We define the ij -th element of the $n \times m$ matrix $\nabla_x F(x)$ as $(\nabla_x F)_{ij} := \frac{\partial F_j}{\partial x_i}$. When clear from the context, the subindex in ∇ is omitted. For any F and the distinguished element $x^* \in \mathbb{R}^n$, we define the constant matrix $F^* := F(x^*)$. Consider the case $n \geq m$ and $\text{rank}\{F\} = m$. Then, the pseudoinverse of F is denoted by $F^\dagger$, that is, $F^\dagger := (F^\top F)^{-1}F^\top$ and $F^\dagger F = I_m$. The left annihilator of F is represented as $F^\perp$, where $\text{rank}\{F^\perp\} = n - m$, and $F^\perp F = \mathbf{0}_{(n-m) \times m}$.

Equilibrium Stabilization via PBC

PBC is often used for Lyapunov stabilization of equilibria. For this task, the key observation is that if $H_d(x)$ verifies

$$x^* = \arg \min H_d(x), \quad (5)$$

where $f(x^*) + g(x^*)\hat{u}(x^*) = \mathbf{0}_n$, then x^* is a *stable* equilibrium of the system in closed-loop with $u = \hat{u}(x)$, with Lyapunov function $H_d(x)$. Moreover, adding the *damping injection* step, the equilibrium x^* will be *asymptotically* stable if y_d is a detectable output for the closed-loop system – see van der Schaft (2016, Corollary 4.2.2).

Energy-Balancing PBC and the Dissipation Obstacle

The most natural way to carry out the energy shaping is to fix $y_d = y$, to look for a function $H_a : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$, solution of the equation

$$\dot{H}_a = -h^\top(x)\hat{u}(x), \quad (6)$$

and to define $H_d(x) = H(x) + H_a(x)$. This variation of PBC is called energy balancing (EB) (Ortega et al. 2001), because $H_d(x)$ consists of the sum of the systems energy and the energy provided by the controller, e.g., $y^\top u$. In spite of

this appealing interpretation, EB-PBC has two serious shortcomings. On one hand, the Eq. (6), which is a partial differential equation (PDE) of the form

$$[\nabla H_a(x)]^\top [f(x) + g(x)\hat{u}(x)] = -h^\top(x)\hat{u}(x),$$

is not amenable for an easy interpretation – because of its explicit dependence on the feedback signal that we are looking for. On the other hand, it is applicable only to systems that are not constrained by the *dissipation obstacle*, that is, systems where dissipation is absent in steady state and can, therefore, be stabilized extracting a finite amount of energy from the controller. Indeed, it is clear that a necessary condition for the existence of a solution of (6) is that $h^\top(x^*)\hat{u}(x^*) = 0$.

The main domain of application of EB-PBC is in regulation of the position $q \in \mathbb{R}^\ell$ of mechanical systems, where $\ell = \frac{n}{2}$. In this case, the dissipation obstacle is conspicuous by its absence because the passive output is velocity $\dot{q}$. Moreover, since in this case we are only interested in adding a new function $V_a(q)$ to the potential energy function, the PDE to be solved reduces to $G^\perp(q)\nabla V_a(q) = \mathbf{0}_{\ell-m}$, where the input matrix is of the form $g(q, \dot{q}) = [\mathbf{0}_{\ell \times m}^\top G^\top(q)]^\top$. In the particular case of fully actuated mechanical systems, it is possible to remove the open-loop potential energy and assign a desired one. This idea was introduced in the pioneering work Takegaki and Arimoto (1981).

Generating Alternative Passive Outputs

One way to overcome the dissipation obstacle is to enforce the power balance inequality (4) with outputs $y_d \neq y$. To achieve this end, it is convenient to restrict our attention to *port-Hamiltonian* (pH) systems, that is systems where

$$f(x) = [\mathcal{J}(x) - \mathcal{R}(x)]\nabla H(x), \quad (7)$$

where $\mathcal{J}(x) = \mathcal{J}^\top(x)$ is the interconnection matrix and $\mathcal{R}(x) \geq 0$ is the dissipation matrix. See van der Schaft (2016) for an extensive discussion, and well-founded practical motivation,

of this class of systems. The following result, which characterizes all passive outputs of the pH system with storage function $H(x)$, was recently established in Zhang et al. (2018).

Lemma 1 Consider the pH system (1) and (7). Introduce the factorization $\mathcal{R}(x) = \phi^\top(x)\phi(x)$, where $\phi(x) \in \mathbb{R}^{q \times n}$, with $q \geq \text{rank} \{\mathcal{R}(x)\}$ and define $y_d := h(x) + j(x)u$. The following statements are equivalent.

- (S1) The mapping $u \mapsto y_d$ is passive with storage function $H(x)$.
- (S2) The mappings $h(x)$ and $j(x)$ can be expressed as

$$\begin{aligned} h(x) &= [g(x) + 2\phi^\top(x)w(x)]^\top \nabla H(x) \\ j(x) &= w^\top(x)w(x) + D(x), \end{aligned}$$

for some mappings $w(x) \in \mathbb{R}^{q \times m}$ and $D(x) \in \mathbb{R}^{m \times m}$, with $D(x) = -D^\top(x)$.

One particularly attractive output that allows us to overcome the dissipation obstacle is the so-called *power-shaping* output. With a suitable selection of the mappings $w(x)$ and $D(x)$ in Proposition 1, it is possible to generate the output

$$\begin{aligned} y_d &= -g^\top(x)[\mathcal{J}(x) - \mathcal{R}(x)]^{-\top} \{[\mathcal{J}(x) \\ &\quad - \mathcal{R}(x)]\nabla H(x) + g(x)u\}, \end{aligned}$$

where, for ease of presentation, it is assumed that the matrix $\mathcal{J}(x) - \mathcal{R}(x)$ is full rank. Since $y_d = \mathbf{0}_m$ at the equilibrium, it is clear that the dissipation obstacle is absent for the system $u \mapsto y_d$.

Interconnection and Damping Assignment (IDA)-PBC

A variation of PBC that has been very successful in many practical applications is IDA-PBC. The main result of IDA-PBC is the following proposition, whose proof may be found in Ortega et al. (2002a).

Proposition 1 Consider the system (1). Assume there are matrices $\mathcal{J}_d(x) = -\mathcal{J}_d^\top(x)$, $\mathcal{R}_d(x) = \mathcal{R}_d^\top(x) \geq 0$, and a function $H_d(x)$, that verify the

matching equation

$$g^\perp(x)f(x) = g^\perp(x)[\mathcal{J}_d(x) - \mathcal{R}_d(x)]\nabla H_d. \quad (8)$$

Then, the closed-loop system with $u = \hat{u}(x)$, where

$$\hat{u}(x) := g^\dagger(x)\{[\mathcal{J}_d(x) - \mathcal{R}_d(x)]\nabla H_d - f(x)\},$$

takes the pH form $\dot{x} = [\mathcal{J}_d(x) - \mathcal{R}_d(x)]\nabla H_d$. Moreover, if $x^* \in \{x \mid g^\perp(x)f(x) = \mathbf{0}_{n-m}\}$ and (5) holds, it is a stable equilibrium.

Conversely, if there exists $\hat{u}(x)$ that globally asymptotically stabilizes (1), then there exist $\mathcal{J}_d(x)$, $\mathcal{R}_d(x)$, and $H_d(x)$ which satisfy (8).

The key step in the IDA-PBC design is, of course, the solution of (8), and there are several approaches to carry out this task. In the non-parameterized IDA-PBC, we fix $\mathcal{J}_d(x)$ and $\mathcal{R}_d(x)$, and (8) becomes a PDE for $H_d(x)$. In algebraic IDA-PBC (Fujimoto and Sugie 2001), we fix $H_d(x)$, and (8) becomes an algebraic equation in $\mathcal{J}_d(x)$ and $\mathcal{R}_d(x)$. In some applications, it is of interest to fix the structure of the desired energy function, for instance, for mechanical systems, it is reasonable to propose

$$H_d(q, p) = \frac{1}{2}p^\top M_d^{-1}(q)p + V_d(q);$$

whence (8) becomes a PDE in the desired inertia matrix $M_d(q)$ and the desired potential energy function $V_d(q)$ – this approach is called parameterized IDA-PBC (Ortega et al. 2002b), with a Lagrangian version given in Bloch et al. (2000). Clearly, fixing this structure imposes some particular constraints on $\mathcal{J}_d(x)$ and $\mathcal{R}_d(x)$. Finally, for nonlinear systems of the form $\dot{x} = f(x, u)$, Poincaré's Lemma states that a necessary and sufficient condition for the solution of the matching equation

$$\nabla H_d(x) = [\mathcal{J}_d(x) - \mathcal{R}_d(x)]^{-1} f(x, \hat{u}(x))$$

is that the right-hand side is a gradient vector field. Fixing $\mathcal{J}_d(x)$ and $\mathcal{R}_d(x)$, this condition translates into a PDE directly for $\hat{u}(x)$.

Proportional-Integral-Derivative (PID)-PBC

PID controllers overwhelmingly dominate engineering applications where the control objective is to regulate some measurable signal y around a constant desired value y^* . Commissioning of PIDs reduces to the suitable selection of the controller gains, which is a difficult task for wide-ranging operating systems, where the validity of a linearized approximation is limited, compromising the stability of the closed-loop. Although gain scheduling, auto-tuning, and adaptation provide some help to overcome this problem, they suffer from well-documented drawbacks. In contrast to this scenario, in PID-PBC, where the PID is wrapped around a passive output, the gain tuning step is trivialized. Indeed, PIDs

$$\begin{aligned}\dot{x}_c &= y \\ u &= -K_P y - K_I x_c - K_D \dot{y},\end{aligned}\quad (9)$$

define, (output-strictly) passive maps $u \mapsto -y$, with storage function $\frac{1}{2}y^\top K_D y + \frac{1}{2}x_c^\top K_I x_c$, for all positive gains. Therefore, convergence of the output to zero and $\mathcal{L}_2$ -stability of the closed-loop system is always guaranteed – and the designers task is only to select the gains that ensure best transient performance.

Shifted Passivity

It is often the case that the reference output $y^* \neq 0$, that suggests to wrap the PID around the error signal $y - y^*$. Unfortunately, for general nonlinear systems, passivity of the mapping $u \mapsto y$ does not imply passivity of $(u - u^*) \mapsto (y - y^*)$ – a property called “shifted passivity” in van der Schaft (2016). The strongest conditions under which the implication holds for pH systems have been reported in Monshizadeh et al. (2019) where, in particular, the following easily verifiable necessary and sufficient condition is given for systems with quadratic nonlinearities.

Lemma 2 Consider the pH system (1) and (7) with g and $\mathcal{R}$ constants,

$$H(x) = \frac{1}{2}x^\top Qx, \quad Q > 0, \quad \mathcal{J}(x) = \mathcal{J}_0 + \sum_{i=1}^n \mathcal{J}_i x_i.$$

Fix

$$\begin{aligned}(x^*, u^*) &\in \{(x, u) \in \mathbb{R}^n \times \mathbb{R}^m \mid \\ &[\mathcal{J}(x) - \mathcal{R}]\nabla H(x) + gu = \mathbf{0}_n\}.\end{aligned}$$

The system satisfies $\dot{\mathcal{H}} \leq (y - y^*)^\top (u - u^*)$, with $\mathcal{H}(x) := \frac{1}{2}(x - x^*)^\top Q(x - x^*)$, if and only if

$$\begin{aligned}&\left(\sum_{i=1}^n \mathcal{J}_i - \mathcal{R}\right)Qx^*e_i^\top Q^{-1} \\ &+ \left[\left(\sum_{i=1}^n \mathcal{J}_i - \mathcal{R}\right)Qx^*e_i^\top Q^{-1}\right]^\top \\ &- 2\mathcal{R} \leq 0,\end{aligned}$$

where $e_i \in \mathbb{R}$ is the i -th element of the orthogonal basis.

Leveraging Lemma 2, we can confidently wrap a PID around the shifted output $y - y^*$. One important observation is that, in the implementation of the PID, there is no need to know u^* . Indeed, by shifting the storage function, we can establish passivity of the map $u \mapsto y - y^*$.

Lyapunov Stabilization via PID-PBC

Another scenario of practical interest is when the control objective cannot be captured by the behavior of an output signal, for instance, when it is desired to drive the full system *state* to a desired equilibrium x^* . To treat this case, it is necessary to create a Lyapunov function for the closed-loop system, an approach taken for pH systems in Zhang et al. (2018). The main difficulty in this case is how to ensure the positivity (with respect to x^*) of the function. Indeed, although for a passive system (1) in closed-loop with a PID (9), we can prove that the function

$$U(x, x_c) := H(x) + \frac{1}{2}h^\top(x)K_D h(x) + \frac{1}{2}x_c^\top K_I x_c$$

satisfies $\dot{U} \leq -y^\top K_P y$ and the function $U(x, x_c)$ is not positive definite – therefore, it does not qualify as a Lyapunov function. One way to overcome this obstacle is to find a mapping

$\gamma(x) \in \mathbb{R}^m$ such that the multilevel set

$$\Gamma := \{(x, x_c) \in \mathbb{R}^n \times \mathbb{R}^m \mid x_c = \gamma(x) + \kappa, \kappa \in \mathbb{R}^m\} \quad (10)$$

is *invariant* and the function

$$H_d(x) := H(x) + U(x, \gamma(x) + \kappa) \quad (11)$$

verifies (5), for some $\kappa \in \mathbb{R}^m$. As expected, this involves the solution of a PDE, as shown below.

Lemma 3 Consider the pH system (1), (7), and $\dot{x}_c = y_d$, with y_d defined in Lemma 1. Assume there exists mappings $w(x)$ and $D(x)$ such that the PDE

$$\begin{aligned} & \begin{bmatrix} [\nabla H(x)]^\top F^\top(x) \\ g^\top(x) \end{bmatrix} \\ \nabla \gamma(x) &= \begin{bmatrix} [\nabla H(x)]^\top [g(x) + 2\phi^\top(x)w(x)] \\ w^\top(x)w(x) - D(x) \end{bmatrix} \end{aligned}$$

admits a solution $\gamma(x) \in \mathbb{R}^m$. Then, the set Γ , given in (10), is *invariant*.

Combining Lemmata 1 and 3, we can complete the design of the PID-PBC as follows.

Proposition 2 Consider the pH system (1) and (7) in closed-loop with the PID-PBC

$$u = -K_P y_d - K_I [\gamma(x) + \kappa] - K_D \dot{y}_d.$$

where $\kappa = -(\gamma^* + K_I^{-1}u^*)$ and y_d is defined as in Lemmata 1 and 3. Assume that $H_d(x)$, given in (11), verifies (5). Then, the closed-loop system has a stable equilibrium at x^* with Lyapunov function (11). Moreover, the equilibrium is asymptotically stable if y_d is a detectable output for the closed-loop system.

Standard PBC of Euler-Lagrange (EL) Systems

Another variation of PBC, particularly suitable for systems described by Euler-Lagrange equations of motion is the so-called standard PBC, that was thoroughly explored in Ortega et al. (2013).

Mathematical Model and Passivity Property

In this case we deal with dynamical systems with ℓ degrees-of-freedom, generalized coordinates $q \in \mathbb{R}^\ell$, and external forces $Q \in \mathbb{R}^\ell$, which are described by the EL equations

$$\frac{d}{dt} [\nabla_{\dot{q}} \mathcal{L}(q, \dot{q})] - \nabla_q \mathcal{L}(q, \dot{q}) = F, \quad (12)$$

where $\mathcal{L}(q, \dot{q}) := T(q, \dot{q}) - V(q)$ is the Lagrangian function, $T(q, \dot{q}) = \frac{1}{2} \dot{q}^\top M(q) \dot{q}$, is the kinetic (co-energy) function, with $M(q) > 0$ the generalized inertia matrix, and $V(q)$ is the potential energy function. The vector F is the external forces that take the form $F = -\nabla \mathcal{F}(\dot{q}) + G(q)u$, where $u \in \mathbb{R}^m$, $m \leq \ell$ are the control and dissipation forces and $\mathcal{F}(\dot{q})$ is the Rayleigh dissipation function verifying $\dot{q}^\top \nabla \mathcal{F}(\dot{q}) \geq 0$.

It is well-known that the EL system (12) defines a passive operator $u \mapsto G^\top(q) \dot{q}$ with storage function, the system's total energy $H(q, \dot{q}) = T(q, \dot{q}) + V(q)$, see Proposition 2.5 in Ortega et al. (2013). Moreover, it is possible to prove a “stronger” property. Namely, if we define an $\ell \times \ell$ matrix $C(q, \dot{q})$ – called in the robotics literature the “Coriolis and centrifugal forces” matrix – with ik -th entry

$$C_{ik}(q, \dot{q}) = \sum_j^\ell c_{ijk}(q) \dot{q}_j.$$

where

$$\begin{aligned} c_{ijk}(q) &:= \frac{1}{2} [\nabla_{q_j} m_{ik}(q) + \nabla_{q_i} m_{jk}(q) \\ &\quad - \nabla_{q_k} m_{ij}(q)] \end{aligned}$$

are the Christoffel symbols of the first kind (Ortega and Spong 1989). Then, the EL system becomes

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + \nabla \mathcal{F}(\dot{q}) + \nabla V(q) = G(q, \dot{q})u, \quad (13)$$

where

$$\begin{aligned} \dot{M}(q) &= C(q, \dot{q}) + C^\top(q, \dot{q}) \\ \Leftrightarrow z^\top [\dot{M}(q) - 2C(q, \dot{q})]z &= 0, \\ \forall z &\in \mathbb{R}^\ell. \end{aligned} \quad (14)$$

Standard PBC

The skew-symmetry property (14), which identifies the work-less forces, is the key component of standard PBC. The main result, for regulation of q , of this technique is given as follows.

Proposition 3 *Consider the EL system (13) verifying (14). Fix the desired position $q^* \in \{q \in \mathbb{R}^\ell \mid G^\perp(q) \nabla V(q) = \mathbf{0}_{\ell-m}\}$. Assume it is possible to find signals $q_d \in \mathbb{R}^\ell$ satisfying the equation*

$$G^\perp(q) [M(q)\ddot{q}_d + C(q, \dot{q})\dot{q}_d + \nabla \mathcal{F}(\dot{q}) + \nabla V(q) - K_D \dot{\tilde{q}} - K_P \tilde{q}] = \mathbf{0}_{\ell-m},$$

where $K_D, K_P > 0$, $\tilde{q} := q - q_d$, and that for the system

$$M(q)\ddot{\tilde{q}} + C(q, \dot{q})\dot{\tilde{q}} + K_D \dot{\tilde{q}} + K_P \tilde{q} = \mathbf{0}_\ell, \quad (15)$$

it is possible to prove that

$$\tilde{q}, \dot{\tilde{q}} \in \mathcal{L}_\infty, \quad \dot{\tilde{q}}(t) \rightarrow \mathbf{0}_\ell \Rightarrow q, \dot{q} \in \mathcal{L}_\infty, \quad q(t) \rightarrow q^*. \quad (16)$$

Under these conditions, the system (13) in closed-loop with the control

$$u = G^\dagger(q) [M(q)\ddot{q}_d + C(q, \dot{q})\dot{q}_d + \nabla \mathcal{F}(\dot{q}) + \nabla V(q) - K_D \dot{\tilde{q}} - K_P \tilde{q}]$$

ensures $q(t) \rightarrow q^*$ with all internal signals bounded.

The gist of the proof is the observation that the error system is given by (15) and that for this system we have the following

$$\begin{aligned} H_d(\tilde{q}) &:= \frac{1}{2} \dot{\tilde{q}}^\top D \dot{\tilde{q}} + \tilde{q}^\top K_P \tilde{q} \Rightarrow \dot{H}_d \\ &= -\dot{\tilde{q}}^\top K_v \dot{\tilde{q}}. \end{aligned}$$

Then, some easy signal chasing and the implication (16) allow us to complete the proof.

The procedure described above is an extension of the well-known controller for fully actuated robot manipulators of Slotine and Li (1988). In that case, $G(q) = I_\ell$ and stabilization (or tracking) of q^* are ensured by selecting

$$q_d := q^* - \Lambda(q - q^*), \quad \Lambda > 0.$$

For underactuated systems, it is not possible to fix all signals q_d to desired values; whence some of them are defined as states of the controller dynamics. In Ortega et al. (2013) the method is applied to a large variety of physical systems, including underactuated mechanical systems, electrical motors, power converters, and levitated systems. Particularly noteworthy is the proof that applying standard PBC to induction motors yields a controller that contains, as a particular case, the industry standard field-oriented control. One major drawback of the method is that the calculation of the controller involves an implicit inversion of the system dynamics; hence its application is restricted to minimum phase systems. This limitation is conspicuous by its absence in IDA-PBC.

Summary and Future Directions

In physical systems, the concepts of energy and dissipation are well defined. Therefore, passivity theory and PBC techniques emerge as natural options to analyze and control, respectively, this kind of systems. However, the range of applicability of PBC is not constrained to physical systems; as a matter of fact, the possibility of decomposing complex nonlinear systems into simpler subsystems makes this approach appealing to tackle down modern problems in control, such as stabilization of biological systems or smart grids.

Cross-References

- [Feedback Stabilization of Nonlinear Systems](#)
- [KYP Lemma and Generalizations/Applications](#)
- [Nonlinear Adaptive Control](#)
- [PID Control](#)
- [Robot Motion Control](#)
- [Stability: Lyapunov, Linear Systems](#)

Bibliography

- Bloch A, Leonard N, Marsden J (2000) Controlled Lagrangians and the stabilization of mechanical sys-

- tems I: the first matching theorem. *IEEE Trans Autom Control* 45(12):2253–2270
- Byrnes CI, Isidori A, Willems JC (1991) Passivity, feedback equivalence and the global stabilization of minimum phase nonlinear systems. *IEEE Trans Autom Control* 36:1228–1240
- Fujimoto K, Sugie T (2001) Canonical transformations and stabilization of generalized Hamiltonian systems. *Syst Control Lett* 42(3):217–227
- Hill D, Moylan P (1980) Dissipative dynamical systems: basic input-output and state properties. *J Frankl Inst* 309:327–357
- Monshizadeh N, Monshizadeh P, Ortega R, van der Schaft A (2019) Conditions on shifted passivity of port-Hamiltonian systems. *Syst Control Lett* 123:55–61
- Ortega R, Spong M (1989) Adaptive motion control of rigid robots: a tutorial. *Automatica* 25(6):877–888
- Ortega R, van der Schaft A, Mareels I, Maschke B (2001) Putting energy back in control. *IEEE Control Syst Mag* 21(2):18–33
- Ortega R, van der Schaft A, Maschke B, Escobar G (2002a) Interconnection and damping assignment passivity-based control of port-controlled hamiltonian systems. *Automatica* 38(4):585–596
- Ortega R, Spong MW, Gomez F, Blankenstein G (2002b) Stabilization of underactuated mechanical systems via interconnection and damping assignment. *IEEE Trans Autom Control* 47(8):1218–1233
- Ortega R, Loria A, Nicklasson PJ, Sira-Ramirez H (2013) Passivity-based control of Euler-Lagrange systems, 3rd edn. Springer, Berlin, Communications and Control Engineering
- Slotine J, Li W (1988) Adaptive manipulator control: a case study. *IEEE Trans Autom Control* 33(11):995–1003
- Takegaki M, Arimoto S (1981) A new feedback for dynamic control of manipulators. *Trans ASME J Dyn Syst Meas Control* 103:119–125
- van der Schaft A (2016) L_2 -gain and passivity techniques in nonlinear control, 3rd edn. Springer, Berlin
- Zhang M, Borja P, Ortega R, Liu Z, Su H (2018) PID passivity-based control of port-Hamiltonian systems. *IEEE Trans Autom Control* 63(4):1032–1044

Pattern Formation in Spin Ensembles

Jr-Shin Li

Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, USA

Abstract

Many applications in quantum science and technology involve controlling the collective

behavior or engineering excitation profiles of a large inhomogeneous ensemble. Developing electromagnetic pulses to achieve these exercises has been a long-standing challenging problem in the field of quantum control. Here, we view this problem through the lens of ensemble control providing some background material on the relevant developments and discussing some specific problem areas where these ideas play a role.

Keywords

Ensemble control · Spin ensembles · Pulse design · Non-harmonic Fourier expansion · Alternating minimization · Coherence transfer

Introduction

The capability to precisely manipulate the time evolution of a quantum ensemble constitutes a significant step toward advancing quantum science and engineering. State-of-the-art quantum technologies can be used to trap and experiment with individual atoms (quantum optics), image brains (magnetic resonance imaging (MRI)), and generate structural and dynamical information of biological macromolecules (nuclear magnetic resonance (NMR) spectroscopy). Many of these applications involve manipulating collective behavior, more specifically engineering excitation patterns, in a large quantum ensemble. A canonical way to enable these applications is through the design and application of time-varying electromagnetic pulses (controls) that are broadcast to the systems at the population level, and inhomogeneities across the entire population of quantum systems, let alone its tremendous scale, form a bottleneck for control-theoretic analysis and design.

From early work using physical intuition (Hahn 1950) to modern methods like composite pulses (Levitt 1986) and gradient flow-based numerical optimization (Khaneja et al. 2005), extensive pulse design techniques have been proposed over the past decades (Bernstein et al.

2004), and the process of innovation is ceaseless. Recent years have witnessed numerous attempts to tackle pulse design problems from a control theory point of view, and these methods have demonstrated promises to establish a systematic and justifiable framework for pulse design (Khaneja et al. 2005; Khaneja et al. 2003; Warren et al. 1993; Li et al. 2011). In this entry, we will view the problem of pulse design through the lens of ensemble control and discuss some specific problem areas where the ensemble control ideas play a critical role.

Control of Spin Ensembles

The evolution of a sample of noninteracting spin- $\frac{1}{2}$ nuclei in a static magnetic field obeys the Bloch equations (Cavanagh et al. 2006). In the presence of external magnetic fields (controls), the bulk magnetization is perturbed from the thermal equilibrium and evolves according to the controlled Bloch model, given by

$$\frac{d}{dt}M(t) = [\omega_0\Omega_z + u(t)\Omega_y + v(t)\Omega_x]M(t), \quad (1)$$

where $M(t) = (M_x(t), M_y(t), M_z(t))$ denotes the magnetization vector at time t , ω_0 is the Larmor frequency, and $(u(t), v(t))$ is the pair of external radiofrequency (rf) fields, referred to as pulses, applied in the y and x axis, respectively; the matrices

$$\Omega_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \Omega_y = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix},$$

$$\Omega_z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

are the generator of rotations around the x , y , and z axis, respectively. Conventionally, $M(t)$ is normalized to 1, i.e., $\|M(t)\| = 1$ for all $t \geq 0$.

The Bloch model in (1) is an idealization assuming that all spins in the ensemble have an identical Larmor frequency and that the rf pulses are applied uniformly to each spin. However, in

practice, e.g., in NMR experiments, variations appear in these quantities across the spin ensemble due to inhomogeneities in chemical environments, relaxation rates, and applied control fields. This leads to the consideration of a modified Bloch model as an ensemble control system of the form

$$\frac{d}{dt}M(t, \omega, \varepsilon) = [\omega\Omega_z + \varepsilon u\Omega_y + \varepsilon v\Omega_x]M(t, \omega, \varepsilon), \quad (2)$$

where $M(t, \omega, \varepsilon)$ denotes the magnetization at time t of the spin characterized by the parameter vector $\beta = (\omega, \varepsilon)$ with $\omega \in [\omega_1, \omega_2]$ and $\varepsilon \in [1 - \delta, 1 + \delta]$, $0 < \delta < 1$, modeling Larmor dispersion and rf inhomogeneity, respectively (Li and Khaneja 2006). In practice, the range of ^{13}C chemical shift leading to Larmor dispersion can be up to 40 kHz, and the rf field at the ends of the coil can be as low as 60% of that at the center of the coil (Li et al. 2011; Bernstein et al. 2004).

Ensemble Control Problems of Pulse Design

A typical problem in quantum control is to steer a large quantum ensemble from an initial state (configuration) $M(0, \omega, \varepsilon)$ at $t = 0$ to, or as close as possible to, a desired target state (configuration) $M_F(\omega, \varepsilon)$ at some time $T > 0$ using a common control input broadcast to the entire population. Practical constraints, such as power and time limitations as well as path and terminal constraints, require the consideration of optimal control problems involving the ensemble system in (2).

Ensemble Controllability

The emergence of ensemble systems was accompanied by a new generation of challenges in control theory, among which the notion of ensemble controllability (Li and Khaneja 2009) has drawn significant attention toward the development of unconventional methods for a comprehensive analysis (Li 2011). Controllability of spin ensembles as in (2) was analyzed through

the mapping of the analysis to a problem of polynomial approximation by utilizing the techniques from Lie theory (Li and Khaneja 2006; Li 2011). This transformation enabled a holistic pulse design methodology. On a practical point of view, controllability of a spin ensemble guarantees the existence of pulses for engineering arbitrary state configurations or patterns in the ensemble.

Pulse-Enabled Pattern Formation

The basic but premier pulse design enabling a wide variety of applications in quantum science and engineering, e.g., NMR spectroscopy, MRI, and quantum information processing, is the broadband pulse that uniformly or selectively excite or invert a spin population in the presence of Larmor dispersion or rf inhomogeneity (Glaser et al. 1998; Pryor 2006). More demanding tasks may be required for more specific applications, for example, fast quantum excitation in protein NMR spectroscopy for mitigation of relaxation effects and multiple slice-selective excitation in MRI for simultaneous scanning in the presence of both Larmor dispersion and rf inhomogeneity (Kobzar et al. 2005). An illustration of a

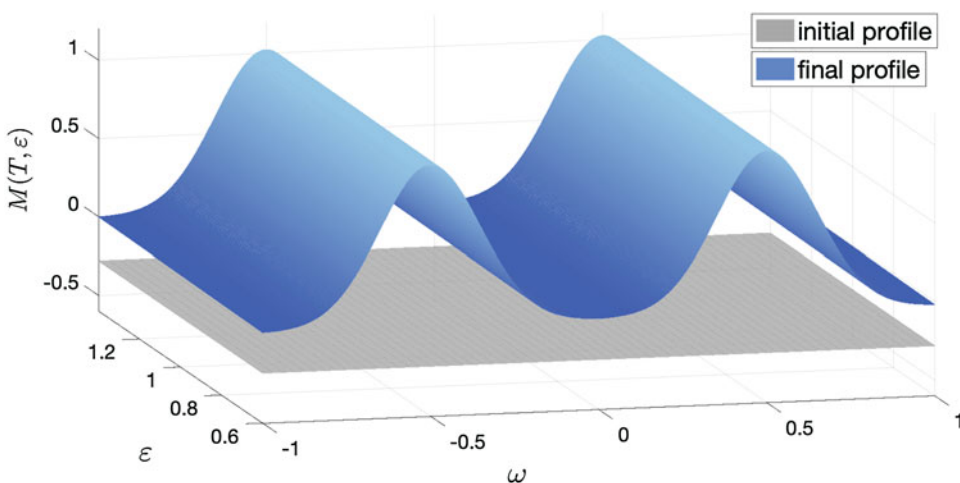
pattern excitation is displayed in Fig. 1, where the excitation profile depends on both Larmor dispersion and rf inhomogeneity.

Generation of Parameter-Dependent Rotations

Since the evolution of the spin ensemble, described in (2), depends on the system parameters, the key to pulse design for achieving the desired state transition or pattern formation is to synthesize parameter-dependent rotations. To fix ideas, let's consider the case of designing uniform and selective excitation pulses in the presence of rf inhomogeneity without Larmor dispersion. In this case, the spin dynamics follow the Bloch model as in (2), which, in the rotating frame with respect to Larmor frequency ω , obeys

$$\frac{d}{dt}M(t, \varepsilon) = \varepsilon[u(t)\Omega_y + v(t)\Omega_x]M(t, \varepsilon). \quad (3)$$

We first observe that one can write the solution to (3) as a product of a sequence of ε -dependent rotations, i.e., $M(t, \varepsilon) = U(t, \varepsilon)M(0, \varepsilon)$, where the total rotation $U(t, \varepsilon) = U_n(t_n, \varepsilon) \cdots U_2(t_2, \varepsilon)U_1(t_1, \varepsilon)$, where $t = \sum_{k=1}^n t_k$. We also know that applying constant controls $u(t) = u_k$ and $v(t) = v_k$ for time t_k to the ensemble system (3)



Pattern Formation in Spin Ensembles, Fig. 1 An (ω, ε) -pattern excitation of a spin ensemble. In this case, a pulse robust to both Larmor dispersion and rf inhomogeneity is sought to excite the ensemble from an initial state to the desired double-excitation pattern

generality is sought to excite the ensemble from an initial state to the desired double-excitation pattern

yields ε -dependent rotations, $\exp(\varepsilon u_k t_k \Omega_y)$ and $\exp(\varepsilon v_k t_k \Omega_x)$ around the y and x axis, respectively. Consequently, a sequence of such constant controls can be used to produce the following ε -dependent rotations:

$$U_{1k} = \exp(-\varepsilon \lambda_k \Omega_x) \exp(\varepsilon \frac{\beta_k}{2} \Omega_y) \exp(\varepsilon \lambda_k \Omega_x), \quad (4)$$

$$U_{2k} = \exp(\varepsilon \lambda_k \Omega_x) \exp(\varepsilon \frac{\beta_k}{2} \Omega_y) \exp(-\varepsilon \lambda_k \Omega_x), \quad (5)$$

for $k = 1, 2, \dots, n$, where $\lambda_k = v_k t_k$ and $\beta_k = 2u_k t_k$, and t_k , u_k , and v_k are parameters. If β_k is small enough such that $U_{1k} U_{2k} \approx U_{2k} U_{1k}$, then the propagator $U_k \doteq U_{2k} U_{1k} \approx \exp[\varepsilon \beta_k \cos(\varepsilon \lambda_k) \Omega_y]$, which approximates an ε -dependent rotation around the y axis of the angle $\varepsilon \beta_k \cos(\varepsilon \lambda_k)$. This approximation is referred to as the *small angle approximation* (Pryor 2006; Zhang and Li 2015). The intuition behind this sequential approximation is to reduce the distortion in each rotation angle caused by rf inhomogeneity.

Following this synthesis procedure, one can generate an ε -dependent total rotation

$$\begin{aligned} U(t, \varepsilon) &= \prod_{k=1}^n U_k(t, \varepsilon) \\ &= \exp \left[\sum_{k=1}^n \varepsilon \beta_k \cos(\varepsilon \lambda_k) \Omega_y \right]. \end{aligned} \quad (6)$$

By an appropriate choice of the parameters β_k and λ_k , any desired ε -dependent rotation around the y axis, $\theta(\varepsilon)$, can be produced by using piecewise constant controls $\{u_k, v_k\}$ such that $\sum_{k=1}^n \varepsilon \beta_k \cos(\varepsilon \lambda_k) \approx \theta(\varepsilon)$. This approximation synthesis procedure illuminates the use of *non-harmonic Fourier series* for pulse design (Zhang and Li 2015; Zhang et al. 2019). Similarly, rotations around the x or the z axis, and thus any three-dimensional rotations, can be synthesized based on the same procedure.

Coherence Transfer in Spin Networks

Coherence transfer between coupled spins is a vital step to multidimensional NMR spectroscopy and quantum information processing (Khaneja et al. 2004; Cavanagh et al. 2006). The small angle approximation and the non-harmonic Fourier series-based approach are directly applicable to design pulses for robust coherence transfer in a network of coupled spins. Here, we illustrate the idea by considering the design of pulses that compensate for variation in the J -coupling constant, which occurs in applications involving large proteins. The coherence transfer can be modeled as an ensemble control problem, given by

$$\frac{d}{dt} X(t, J) = A(t, J) X(t, J), \quad (7)$$

where $X(t, J) \in \mathbb{R}^6$ denotes the state of the two spins,

$$\begin{aligned} A(t, J) &= \begin{bmatrix} 0 & -v(t) & u(t) & 0 & 0 & 0 \\ v(t) & 0 & 0 & -J & 0 & 0 \\ -u(t) & 0 & 0 & 0 & J & 0 \\ 0 & -J & 0 & 0 & 0 & -u(t) \\ 0 & 0 & -J & 0 & 0 & v(t) \\ 0 & 0 & 0 & u(t) & -v(t) & 0 \end{bmatrix} \\ &= J(\Omega_{24} - \Omega_{35}) + u(t)(\Omega_{46} - \Omega_{13}) + v(t)(\Omega_{12} - \Omega_{56}), \end{aligned}$$

with $\Omega_{ij} \in \mathbb{R}^{6 \times 6}$ being the matrix with -1 in the ij th entry and 0 elsewhere, and $J \in [1 - \delta, 1 + \delta]$, $0 < \delta < 1$, denotes the coupling between the two spin systems. Observe that we may treat $P_1 \doteq \Omega_{24} - \Omega_{35}$ and $P_2 \doteq \Omega_{12} - \Omega_{56}$ as Ω_x and Ω_y , respectively, in comparison with the case studied above for the system in (3), and then proceed with the same scheme of small angle approximation and non-harmonic Fourier series expansion to reach the desired J -dependent propagation. Specifically, by letting $U_{1k} = \exp(\lambda_k J P_1) \exp(\frac{\beta_k}{2} P_2) \exp(-\lambda_k J P_1)$ and $U_{2k} = \exp(-\lambda_k J P_1) \exp(\frac{\beta_k}{2} P_2) \exp(\lambda_k J P_1)$, small angle approximations yield

$$U(t, J) = \prod_{k=1}^n U_k(t, J) \\ = \exp\left(\sum_{k=1}^n \beta_k \cos(\lambda_k J) P_2\right). \quad (8)$$

Note that another routine of small angle approximations needs to be implemented to resolve the involvement of the drift term P_2 in (8) (Zhang et al. 2019).

Alternating Optimization for Optimal Pulse Synthesis

The generated parameter-dependent evolution as in (6) and (8) will be used to synthesize optimal pulses. A natural way to achieve this is to formulate the non-harmonic Fourier approximation as an optimization problem on a Hilbert space. For example, the case in (6) is formulated as

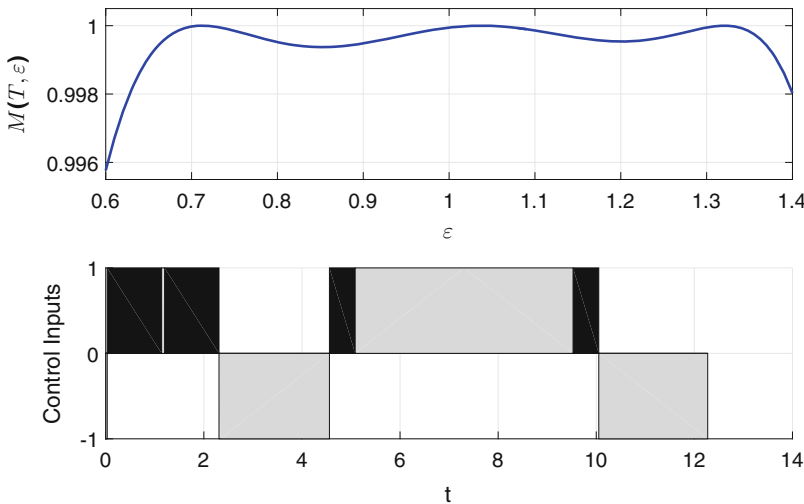
$$\min_{\beta, \lambda} J(\beta, \lambda) = \left\| \phi(\varepsilon) - \sum_{k=1}^n \beta_k \cos(\lambda_k \varepsilon) \right\|_2^2, \quad (9)$$

where $\phi(\varepsilon) = \theta(\varepsilon)/\varepsilon$, $\beta = (\beta_1, \dots, \beta_n) \in \mathbb{R}^n$, $\lambda = (\lambda_1, \dots, \lambda_n) \in \mathbb{R}^n$, and $\|\cdot\|_2$ denotes the L^2 -

norm on the Hilbert space $\mathcal{H} = L^2([1 - \delta, 1 + \delta])$, i.e., $\|f\|_2 = \sqrt{\int_{1-\delta}^{1+\delta} |f(\varepsilon)|^2 d\varepsilon}$. The solution (β, λ) then determines the pulse sequence through the relations $u_k = \beta_k/2t_k$ and $v_k = \lambda_k/t_k$. Most importantly, this optimization problem can be effectively solved by using an alternating minimization approach, alternating between the decision variables β and λ (Zhang and Li 2015).

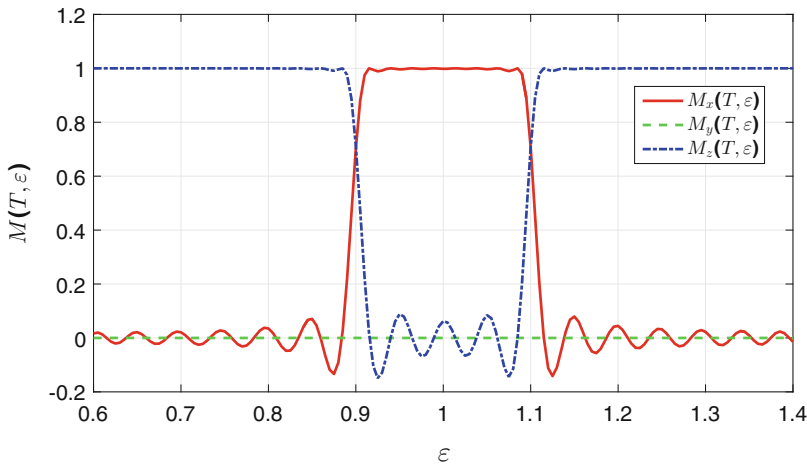
A uniform excitation $(\pi/2)$ pulse that compensates for a 40% variation in the strength of the applied rf field constructed by solving the optimization problem in (9) is illustrated in Fig. 2, where the order of the non-harmonic Fourier series was 3 ($n = 3$), and the piecewise constant control sequences $\{u_k, v_k\}$ were calculated, which steer the spin ensemble in (3) with $\varepsilon \in [0.6, 1.4]$ from $M(0, \varepsilon) = (0, 0, 1)'$ to $M_F(\varepsilon) = (1, 0, 0)'$ at time $T = 1.4$. The duration of the pulse, T , depends on the optimized non-harmonic Fourier coefficients, i.e., $T = \sum_{k=1}^n (|\frac{\beta_k}{u_0}| + 4|\frac{\lambda_k}{v_0}|)$. A selective excitation pulse that created the excitation pattern,

$$\phi(\varepsilon) = \begin{cases} \frac{\pi}{2\varepsilon}, & 0.9 \leq \varepsilon \leq 1.1 \\ 0, & 0.6 \leq \varepsilon < 0.9, \quad 1.1 < \varepsilon \leq 1.4, \end{cases} \quad (10)$$



Pattern Formation in Spin Ensembles, Fig. 2 The final profile of $M_x(T, \varepsilon)$ at $T = 1.4$ of the spin ensemble in (3) for $\varepsilon \in [0.6, 1.4]$ following the uniform $\pi/2$ pulse

constructed using three ($n = 3$) non-harmonic Fourier terms, i.e., $\theta(\varepsilon)/\varepsilon \approx \sum_{k=1}^n \beta_k \cos(\lambda_k \varepsilon)$



Pattern Formation in Spin Ensembles, Fig. 3 The selective excitation pattern was created by using a pulse designed using 15 ($n = 15$) non-harmonic Fourier terms, which generated the desired ε -dependent rotations as in (10)

is shown in Fig. 3, where spins experiencing different rf intensity were selectively excited.

Summary and Future Directions

Ensemble control lays a rigorous and systematic platform for robust control of quantum ensembles. Unconventional control problems, such as pulse design arising from the quantum domain, emerged from other remote fields as well, for example, in neuroscience, cell- and chronobiology, and social networks. These emerging problems require innovations and investments in developing integrated and out-of-the-box methods. Major challenges remain, especially to investigate optimal control and learning problems involving quantum ensembles (in particular open quantum systems) or more general ensemble systems, understand fundamental limits on the ability to control ensemble systems and further explore structural control properties and utilize structures to control ensembles.

Cross-References

- [Control of Quantum Systems](#)
- [Learning Theory: The Probably Approximately Correct Framework](#)

Recommended Reading

The literature on pulse design or quantum control can be found in the comprehensive research and survey articles (Kobzar et al. 2005; Li et al. 2011; Dong and Petersen 2010; Bouten et al. 2007). Motivational and interesting aspects of using polynomial approximation to synthesize pulse sequences can be found in Shinnar and Leigh (1989) and Li et al. (2011). In-depth material on ensemble control can be found in Li and Khaneja (2009) and Li (2011); Li et al. (2013); Wang and Li (2017) and the references therein. The example problems discussed here mostly come from the literature (Zhang and Li 2015; Pryor 2006).

Bibliography

- Bernstein MA, King KF, Zhou XJ (2004) Handbook of MRI Pulse Sequences. 2004. Elsevier/Academic Press, Burlington
- Bouten L, Handel RV, James MR (2007) An introduction to quantum filtering. *SIAM J Control Optim* 46(6):2199–2241
- Cavanagh J, Fairbrother WJ, Palmer I AG, Rance M, Skelton NJ (2006) Protein NMR spectroscopy: principles and practice, 2nd edn. Academic, San Diego
- Dong D, Petersen IR (2010) Quantum control theory and applications: a survey. *IET Control Theory Appl* 4:2651–2671(20)

- Glaser SJ, Schulte-Herbrüggen T, Sieveking M, Schedlet-zky O, Nielsen NC, Sørensen OW, Griesinger C (1998) Unitary control in quantum ensembles, maximizing signal intensity in coherent spectroscopy. *Science* 280:421–424
- Hahn EL (1950) Spin echoes. *Phys Rev* 80:580
- Khaneja N, Luy B, Glaser SJ (2003) Boundary of quantum evolution under decoherence. *Proc Natl Acad Sci* 100:13162–13166
- Khaneja N, Li JS, Kehlet C, Luy B, Glaser SJ (2004) Broadband relaxation-optimized polarization transfer in magnetic resonance. *Proc Natl Acad Sci* 101: 14742–14747
- Khaneja N, Reiss T, Kehlet C, S-Herbruggen T, Glaser SJ (2005) Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. *J Magn Reson* 172: 296–305
- Kobzar K, Luy B, Khaneja N, Glaser SJ (2005) Pattern pulses: design of arbitrary excitation profiles as a function of pulse amplitude and offset. *J Magn Reson* 173(2):229–235
- Levitt MH (1986) Composite pulses. *Prog NMR Spectrosc* 18:61–122
- Li JS (2011) Ensemble control of finite-dimensional time-varying linear systems. *IEEE Trans Autom Control* 56(2):345–357
- Li JS, Khaneja N (2006) Control of inhomogeneous quantum ensembles. *Phys Rev A* 73:030302
- Li JS, Khaneja N (2009) Ensemble control of Bloch equations. *IEEE Trans Autom Control* 54(3): 528–536
- Li JS, Ruths J, Yu TY, Arthanari H, Wagner G (2011) Optimal pulse design in quantum control: a unified computational method. *Proc Natl Acad Sci* 108(5):1879–1884
- Li JS, Dasanayake I, Ruths J (2013) Control and synchronization of neuronal ensembles. *IEEE Trans Autom Control* 58(8):1919–1930
- Pryor B (2006) Fourier decompositions and pulse sequence design algorithms for nuclear magnetic resonance in inhomogeneous fields. *J Chem Phys* 125(19):194111
- Shinnar M, Leigh JS (1989) The application of spinors to pulse synthesis and analysis. *Magn Reson Med* 12: 93–98
- Wang S, Li JS (2017) Fixed-endpoint optimal control of bilinear ensemble systems. *SIAM Journal on Control and Optimization* 55(5):3039–3065
- Warren WS, Rabitz H, Dahleh M (1993) Coherent control of quantum dynamics: the dream is alive. *Science* 259:1581–1589
- Zhang W, Li JS (2015) Uniform and selective excitations of spin ensembles with RF inhomogeneity. In: 54th IEEE conference on decision and control, Osaka, pp 5766–5771
- Zhang W, Narayanan V, Li JS (2019) Robust population transfer for coupled spin ensembles. In: 58th IEEE conference on decision and control, Nice, France, pp 419–424

Perturbation Analysis of Discrete Event Systems

Yorai Wardi¹ and Christos G. Cassandras²

¹School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

²Division of Systems Engineering, Center for Information and Systems Engineering, Boston University, Brookline, MA, USA

Abstract

Perturbation analysis (PA) is a systematic methodology for estimating the sensitivities (gradients) of performance measures in discrete event systems (DES) with respect to various model or control parameters of interest. PA takes advantage of the special structure of DES sample realizations and is based entirely on observable system data. In particular, it does not require knowledge of the stochastic characterizations of the random processes involved and is simple to implement in a nonintrusive manner. PA estimators, therefore, enable implementations for real-time control in addition to off-line optimization. This entry presents the main ideas and statistical properties of PA techniques for both DES and their recent generalizations to stochastic hybrid systems (SHS), especially for the simplest class of sensitivity estimators known as infinitesimal perturbation analysis (IPA).

Keywords

Gradient estimation · Sample-path techniques · Sensitivity analysis · Discrete event systems · Hybrid dynamical systems

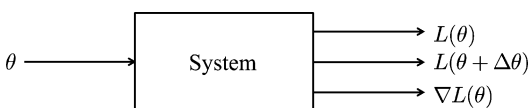
Introduction

Sensitivity analysis is an essential component of the system design process in a wide variety of application areas. In essence, it provides

quantitative variations of performance metrics resulting from possible perturbations in design set points and hence can be used in *optimization and control* as well as provide measures of performance robustness. *Perturbation analysis* (PA) is a systematic technique for computing sample-based sensitivity estimators of performance metrics in discrete event systems (DES) by using the special properties of their sample realizations. The effectiveness of such estimators (e.g., unbiasedness) depends on the characteristics of the DES to which PA is applied and on the specific performance metric of interest. The purpose of this entry is to present and explain some of the main ideas and fundamental techniques of PA.

Figure 1 depicts an abstract schematic where the operation of a stochastic system depends on a parameter θ that is chosen from a given set Θ . Let $J(\theta)$ be an expected-value performance function of the system, and suppose that $J(\theta) = E[L(\theta)]$, where $E[\cdot]$ denotes expectation and $L(\theta)$ is a sample realization computable from a sample path of the system. In many situations $J(\theta)$ lacks a closed-form expression, and its sample realization, $L(\theta)$, provides the most practical way for its estimation. Applications of sensitivity analysis often concern the effects of perturbations in the parameter θ on the sample realization $L(\theta)$. Denoting such perturbations by $\Delta\theta$, their effects can be characterized by the difference term $L(\theta + \Delta\theta) - L(\theta)$. PA provides such difference terms from the *same* sample path that was used for computing $L(\theta)$. Furthermore, if $\theta \in R^n$ and the function $L(\cdot)$ is differentiable, then PA can compute the gradient-term $\nabla L(\theta)$ from the same sample path. These sample-path-based sensitivities can be used, under suitable conditions, to estimate the quantities $J(\theta + \Delta\theta) - J(\theta)$ and $\nabla J(\theta)$, respectively.

The PA theory was pioneered by Yu-Chi Ho who led its eventual development by his group



Perturbation Analysis of Discrete Event Systems, Fig. 1 Framework for perturbation analysis (PA)

and other researchers over two decades. The early works were motivated by optimal resource management problems in manufacturing and concerned the effects of buffer allocation on throughput in transfer lines (Ho et al. 1979; Ho and Cassandras 1983). Subsequently PA was developed in the setting of queueing networks by virtue of their wide use as models in applications. In this setting, typically θ is a set-point parameter affecting service times, inter-arrival times, routing fractions, buffer sizes, and various flow-control laws; $J(\theta)$ is an expected-value performance metric like average delay, throughput, and loss; and $L(\theta)$ is a sample realization of $J(\theta)$. The special structure of sample paths of queueing networks often yields simple PA algorithms for the difference terms $L(\theta + \Delta\theta) - L(\theta)$, as well as the gradient term $\nabla L(\theta)$, from the common sample path. The PA techniques for computing $L(\theta + \Delta\theta) - L(\theta)$ are collectively referred to as *finite perturbation analysis* (FPA), while those for computing $\nabla L(\theta)$ are called *infinitesimal perturbation analysis* (IPA) (Ho et al. 1983). Much of the development of PA has focused on IPA, rather than FPA, due to its greater simplicity and natural use in optimization, and hence it will be the focal point of this entry. For comprehensive expositions of PA and its various techniques, please see Ho and Cao (1991), Glasserman (1991), and Cassandras and Lafortune (2008).

The purpose of the IPA gradient, $\nabla L(\theta)$, is to estimate $\nabla J(\theta)$. This, however, can be useful only as long as $\nabla L(\theta)$ is an unbiased realization of $\nabla J(\theta)$, namely,

$$E[\nabla L(\theta)] = \nabla E[L(\theta)] = \nabla J(\theta),$$

and in this case, it is said that *IPA is unbiased* (Cao 1985). Since $J(\theta) = E[L(\theta)]$, unbiasedness means the commutativity of the operators of differentiation with respect to θ and integration (expectation) in the probability space, and this is closely related to the condition that, with probability one (w.p.1), the random function $L(\theta)$ is continuous throughout Θ . As a matter of fact, the two conditions are practically synonymous. However, shortly after the emergence of IPA, it became apparent that in many

queueing models of interest, $L(\theta)$ is not continuous, and hence IPA is not unbiased (Heidelberger et al. 1988). Subsequently various techniques to overcome this problem were explored, including so-called cut-and-paste of the sample paths and re-parametrization of the underlying probability space via statistical conditioning. For a more comprehensive coverage of such techniques, please see Cassandras and Lafortune (2008) and references therein. These techniques can yield unbiased gradient estimators in principle but often at the expense of prohibitive computing costs. Recently an alternative approach has emerged, based on *stochastic flow models* (SFM) that are comprised of fluid queues (Cassandras et al. 2002). It extends, significantly, the class of models and problems where IPA is unbiased and has the added advantage of yielding very simple gradient estimators.

The following sections of this entry present IPA in the general setting of DES, explain the limits of its scope in queueing models, describe alternative PA techniques for extending those limits, present the SFM approach, and conclude with some thoughts on current research directions.

DES Setting for IPA

IPA can be applied to any DES modeled as a stochastic timed automaton, defined in Cassandras and Lafortune (2008) and discussed in “► [Models for Discrete Event Systems: An Overview](#).” Briefly, a stochastic timed automaton is a six-tuple $(\mathcal{E}, \mathcal{X}, \Gamma, p, p_0, G)$, where $\mathcal{E}$ is an event set, $\mathcal{X}$ is a state space, and $\Gamma(x) \subseteq \mathcal{E}$ is the set of feasible events when the state is x , defined for all $x \in \mathcal{X}$. The initial state is drawn from $p_0(x) = P[X_0 = x]$. Subsequently, given that the current state is x , with each feasible event $i \in \Gamma(x)$, we associate a clock value Y_i , which represents the time until event i is to occur. Thus, comparing all such clock values, we identify the triggering event $E' = \arg \min_{i \in \Gamma(x)} \{Y_i\}$ where $Y^* = \min_{i \in \Gamma(x)} \{Y_i\}$ is the interevent time (the time elapsed since the last event occurrence). To simplify the notation, we define $e' := E'$. Thus, with e' determined, the state transition

probabilities $p(x'; x, e')$ are used to specify the next state x' . Finally, the clock values are updated: Y_i is decremented by Y^* for all i (other than the triggering event) which remain feasible in x' , while the triggering event (and all other events which are activated upon entering x') are assigned a new lifetime sampled from a distribution G_i . The set $G = \{G_i : i \in \mathcal{E}\}$ defines the stochastic clock structure of the automaton.

Let $T_{\alpha,n}$ denote the n th occurrence time of event $\alpha \in \mathcal{E}$, and let $V_{\alpha,n}$ denote a realization of the lifetime event distribution G_α such that $V_{\alpha,n} = T_{\alpha,n} - T_{\beta,m}$ for some (any) event $\beta \in \mathcal{E}$ and $m \in \{1, 2, \dots\}$. One can then always write $T_{\alpha,n} = V_{\beta_1,k_1} + \dots + V_{\beta_s,k_s}$ for some s . Let us now consider a parameter $\theta \in R$ which can only affect one or more of the event lifetime distributions $G_\alpha(x; \theta)$; in particular, θ does not affect the state transition mechanism. The case where $\theta \in R^n$ can be handled in similar ways, but the one-dimensional case permits us to use the derivative notation rather than the gradient symbol, which simplifies the presentation. Viewing lifetimes as functions of θ , $V_{\alpha,k}(\theta)$, it can be shown (under mild technical conditions, see Glasserman 1991) that

$$\frac{dV_{\alpha,k}}{d\theta} = - \frac{[\partial G_\alpha(x; \theta)/\partial \theta]_{(V_{\alpha,k}, \theta)}}{[\partial G_\alpha(x; \theta)/\partial x]_{(V_{\alpha,k}, \theta)}}, \quad (1)$$

where the subscript $(V_{\alpha,k}, \theta)$ indicates that the corresponding derivative of G_α is evaluated at the point $(V_{\alpha,k}, \theta)$. This describes how a perturbation in θ *generates* a perturbation in the associated event lifetime $V_{\alpha,k}$. Such a perturbation can now *propagate* through the DES to affect various event occurrence times according to the dynamics prescribed by the stochastic timed automaton. Event time derivatives $dT_{\alpha,n}(\theta)/d\theta$ are given by

$$\frac{dT_{\alpha,n}}{d\theta} = \sum_{\beta,m} \frac{dV_{\beta,m}}{d\theta} \eta(\alpha, n; \beta, m) \quad (2)$$

where $\eta(\alpha, n; \beta, m)$ is a triggering indicator taking values in $\{0, 1\}$ so that $\eta(\alpha, n; \beta, m) = 1$ if the n th occurrence of event α is triggered by the m th occurrence of β ; $\eta(\alpha, n; \alpha, n) = 1$ for all $\alpha \in \mathcal{E}$; $\eta(\alpha, n; \beta', m') = 1$ if $\eta(\alpha, n; \beta, m) = 1$

and $\eta(\beta, m; \beta', m') = 1$; and $\eta(\alpha, n; \beta, m) = 0$ otherwise.

This leads to a general-purpose algorithm for evaluating event time derivatives along an observed sample path (see Algorithm 1) of a DES modeled as a stochastic timed automaton. In particular, we define a perturbation accumulator, Δ_α , for every event $\alpha \in \mathcal{E}$. The accumulator Δ_α is updated at event occurrences in two ways: (i) it is incremented by $dV_\alpha/d\theta$ whenever an event α occurs, and (ii) it is coupled to an accumulator Δ_β whenever an event β (possibly $\beta = \alpha$) occurs that activates an event α . No particular stopping condition is specified, since this may vary depending on the problem of interest.

Algorithm 1 General-purpose IPA algorithm for stochastic timed automata

1. Initialization

If event α is feasible at x_0 : $\Delta_\alpha := dV_{\alpha,1}/d\theta$
 Else, for all other $\alpha \in \mathcal{E}$: $\Delta_\alpha := 0$
 2. Whenever event β is observed

If event α is activated with new lifetime V_α :

 - 2.1. Compute $dV_\alpha/d\theta$ through (1)
 - 2.2. $\Delta_\alpha := \Delta_\beta + dV_\alpha/d\theta$
-

Sample Function Derivatives Since many sample performance functions $L(\theta)$ of interest can be expressed in terms of event times $T_{\alpha,n}$, we can use (2) and Algorithm 1 in order to obtain derivatives of the form $dL(\theta)/d\theta$. As an example, a large class of such functions is of the form

$$L_T(\theta) = \int_0^T C(x(t, \theta)) dt$$

where $C(x(t, \theta))$ is a bounded cost associated with operating the system at state $x(t, \theta)$. Then,

$$\frac{dL_T}{d\theta} = \sum_{k=1}^{N(T)} \frac{dT_k}{d\theta} [C(x_{k-1}) - C(x_k)]$$

where $N(T)$ counts the total number of events observed in $[0, T]$ and x_k is the state remaining

fixed in any interval $(T_k, T_{k+1}]$ with $T_k = T_{\alpha,n}$ for some $\alpha \in \mathcal{E}, n = 1, 2, \dots$

Estimation of Performance Measure Derivatives Using $dL_T(\theta)/d\theta$ from above, we can obtain unbiased estimates of $dJ/d\theta$ if the following condition holds:

$$\frac{dJ(\theta)}{d\theta} = E \left[\frac{dL_T(\theta)}{d\theta} \right].$$

As mentioned earlier, this key condition is closely related to the continuity of the sample performance functions. A discontinuity often is caused by a swap in the order of two events that results from small variations in θ and yields different future state trajectories. However, it is possible that the future state trajectory following the occurrence of the two events is invariant under their order, and in this case, the two events are said to *commute*. This commuting condition, defined by Glasserman, was shown to be identical, under broad assumptions, to the continuity of the sample functions $L(\theta)$ and hence to the unbiasedness of IPA (Glasserman 1991).

The main ideas discussed in this section will next be illustrated on a simple queue.

Queueing Example

Consider the IPA gradient (derivative) of the expected sojourn time (delay) in a GI/G/1 queue with respect to a real-valued parameter of its arrival process. Assume that the queue is empty at time $t = 0$, and it serves its customers according to the order of their arrivals. Let us denote by $a_k, k = 1, 2, \dots$, the arrival times, and by $s_k, k = 1, 2, \dots$, the service times of consecutive customers. Furthermore, we denote by $v_k, k = 1, 2, \dots$, the k th inter-arrival time, namely, $v_k = a_k - a_{k-1}$, where $a_0 := 0$. Observe that the queue is a stochastic timed automaton as defined earlier, with event space $\{arrival, departure\}$, state space $\{0, 1, \dots\}$ representing the queue's occupancy, and feasible event set $\{arrival\}$ when $x = 0$ and $\{arrival, departure\}$ when $x > 0$.

Let $\theta \in \mathcal{R}$ is a parameter of the distribution of the inter-arrival times, and hence the realizations

of v depend on θ in a functional manner. For example, if the arrival process is Poisson and θ is its rate, then a realization of the inter-arrival times has the form $v = -\theta \ln(1 - \omega)$, where $\omega \in [0, 1]$ is a uniform variate. To emphasize the dependence of v on θ , we denote it by $v(\theta)$, while its dependence on ω is implicit. Similarly, the arrival times depend on θ and hence denoted by $a_k(\theta)$, but the service times s_k do not depend on θ . The departure time of the k th customer and its delay depend on θ and are denoted by $d_k(\theta)$ and $D_k(\theta)$, respectively. The forthcoming paragraphs discuss the derivatives of these functions with respect to θ ; we use the prime symbol to indicate such derivatives in order to simplify the notation.

Define the sample performance function

$$L_N(\theta) := N^{-1} \sum_{k=1}^N D_k(\theta)$$

for a given $N > 0$. Under stability conditions, $\lim_{N \rightarrow \infty} L_N(\theta) = J(\theta)$ w.p.1, where $J(\theta)$ denotes the mean of the delay's stationary distribution. The role of IPA is to estimate $J'(\theta)$ via the sample derivative $L'_N(\theta)$, and this is justified as long as $\lim_{N \rightarrow \infty} L'_N(\theta) = J'(\theta)$ w.p.1. In this case, IPA is said to be *strongly consistent*. In contrast, unbiasedness pertains to the performance function $J_N(\theta) := E[L_N(\theta)]$ and means that $E[L'_N(\theta)] = J'(\theta)$. The concepts of strong consistency and unbiasedness are closely related except that the latter concerns finite-horizon processes while the former pertains to stationary distributions in steady state. Both concepts have been extensively investigated in recent years: strong consistency in the setting of Markov chains and Markov decision processes ([► Perturbation Analysis of Steady-State Performance and Relative Optimization](#), Cao 2007) and unbiasedness in the context of stochastic hybrid systems, as will be described in the sequel. We will focus the rest of the discussion on the issue of unbiasedness.

Since $L_N(\theta) = N^{-1} \sum_{k=1}^N D_k(\theta)$, its IPA derivative is $L'_N(\theta) = N^{-1} \sum_{k=1}^N D'_k(\theta)$, and since $D_k(\theta) = a_k(\theta) - d_k(\theta)$, it follows that $D'_k(\theta) = a'_k(\theta) - d'_k(\theta)$. The last two derivative

terms are computable via the following recursive procedures. First, $a_k(\theta) = a_{k-1}(\theta) + v_k(\theta)$, and hence

$$a'_k(\theta) = a'_{k-1}(\theta) + v'_k(\theta).$$

The term $v'_k(\theta)$ has to be obtained directly from the realization of v , and this often is possible since such realizations depend functionally on θ ; for instance, in the previous example, $v = -\theta \ln(1 - \omega)$, and hence $v'(\theta) = -\ln(1 - \omega)$. Next, $d_k(\theta)$ is given by the Lindley equation

$$d_k(\theta) = \max \{a_k(\theta), d_{k-1}(\theta)\} + s_k,$$

and therefore, denoting by ℓ_k the index of the customer that started the busy period containing customer k , we have that

$$d'_k(\theta) = a'_{\ell_k}(\theta).$$

From these recursive relations, it follows that $D'_k(\theta) = 0$ if customer k starts a busy period and $D'_k(\theta) = -\sum_{i=\ell_k+1}^k v'_i(\theta)$ if customer k does not start a busy period.

These equations shed light on the structure of IPA in a general class of queueing networks. First there is the *perturbation generation*, namely, the sampling of derivative (gradient) terms directly from the sample sequence of variates (ω) which define the sample path; that was $v'(\theta)$ in the above example. These terms drive the recursive equations that yield the IPA derivatives. The recursive equations often are based on the tracking of certain events such as the start of busy periods or idle periods, and the process of tracking the derivatives through them is referred to as *perturbation propagation*. In the above example, it is obvious how the perturbation propagation tracks the busy periods at the queue. Furthermore, in a network setting, the perturbations can propagate from one queue to the next in a natural fashion. For instance, suppose that customers departing from the queue analyzed above enter a second queue. Then the derivative terms $d'_k(\theta)$ of the upstream queue act as the derivative terms $a'_k(\theta)$ of the downstream queue. This structure of perturbations' generation and propagation through the tracking of busy periods and other events



Perturbation Analysis of Discrete Event Systems,
Fig. 2 Queue with two customer classes

often yields simple recursive algorithms for computing the IPA derivatives.

Concerning the issue of unbiasedness, it is clear that in the above example, the sample function $L_N(\theta)$ is continuous in θ , and hence IPA is unbiased. However, in many systems of interest, the IPA derivative is biased. For example, consider the two-input, single-server queue shown in Fig. 2, where customers are served according to their arrival order regardless of source. Suppose that θ is a parameter of the upper arrival process but not of the lower arrival process and denote the respective inter-arrival times of the input streams by $v_{1,k}(\theta)$, $k = 1, 2, \dots$, and $v_{2,m}$, $m = 1, 2, \dots$, as indicated in the figure. Furthermore, let $d_{1,k}(\theta)$ denote the departure time from the queue of the k th customer that came from the upper source, and let $d_{2,m}(\theta)$ denote the departure time from the queue of the m th customer that came from the lower source. Similarly, let $D_{1,k}(\theta)$ and $D_{2,m}(\theta)$ be the delays of the k th customer from the upper source and the m th customer from the lower source, respectively. Lastly, in analogy with the previous example, consider the sample performance functions $L_{1,N}(\theta) := N^{-1} \sum_{k=1}^N D_{1,k}(\theta)$ and $L_{2,N}(\theta) := N^{-1} \sum_{m=1}^N D_{2,m}(\theta)$.

The IPA derivatives $L'_{1,N}(\theta)$ and $L'_{2,N}(\theta)$ have quite similar expressions to those derived earlier, but they are not unbiased. To see this point, suppose that $v_{1,k}(\cdot)$ is a monotone increasing function of θ , and consider the functions $a_{1,k}(\theta)$, $d_{1,k}(\theta)$, and $d_{2,k}(\theta)$ for a common sample path. Suppose that at some point $\bar{\theta}$ the order of arrivals of the n th customer from the upper source and the m th customer from the lower source is swapped. Then the service order of these customers will be swapped as well, inducing discontinuities in $d_{1,k}(\theta)$ and $d_{2,m}(\theta)$ at the point $\theta = \bar{\theta}$. Consequently, the sample performance functions $L_{1,N}(\theta)$ and $L_{2,N}(\theta)$ also will be

discontinuous at $\theta = \bar{\theta}$, and hence their IPA derivatives are biased. Furthermore, if the queue is a part of a network and its output process directs customers to other queues, then the discontinuities in the various traffic processes will propagate downstream.

The causes of bias of IPA in queueing networks include multiple customer classes, non-Markovian routing, and loss (spillover) due to finite buffers. This leaves a limited class of networks where IPA can be unbiased and hence useful in applications. The following sections describe various approaches developed for enlarging the class of systems where IPA is unbiased.

IPA Extensions

When IPA fails (because the commuting condition is violated or a sample function exhibits discontinuities in θ), one can still use the PA approach to derive unbiased performance sensitivity estimates. There are two ways to accomplish this: (i) by modifying the stochastic timed automaton model so that IPA is “made to work” and (ii) by paying the price of more information collected from the observed sample path, in which case the same essential PA philosophy can lead to unbiased and strongly consistent estimators, but these are no longer as simple as IPA ones. Regarding (i), the main idea here is that there may be more than one way to construct (statistically equivalent) sample paths of a stochastic DES, and while one way leads to discontinuous sample functions $L(\theta)$, another does not; a variety of such ways is provided in Cassandras and Lafortune (2008). Regarding (ii), the methodology of *smoothed perturbation analysis* (SPA) (Gong and Ho 1987) provides a generalization of IPA in which more information is extracted from a DES sample path in order to gain some knowledge about the magnitude of jumps in $L(\theta)$.

The main idea of SPA lies in the “smoothing property” of conditional expectation. If we are willing to extract information from a sample path and denote it by Z , called the *characterization* of

the sample path, then we can evaluate, not just the sample function $L(\theta)$ but also the conditional expectation $E[L(\theta) \mid \mathcal{Z}]$ (provided we have some distributional knowledge based on which this expectation can be evaluated). This can result in a smoother function of θ than $L(\theta)$. Thus, starting with the condition for an IPA estimator to be unbiased,

$$\nabla J(\theta) = E[\nabla L(\theta)],$$

we rewrite the right-hand side above as shown below, replacing $E[L(\theta)]$ by the expectation of a conditional expectation:

$$\nabla J(\theta) = \nabla E[L(\theta)] = \nabla E[E[L(\theta) \mid \mathcal{Z}]] \quad (3)$$

where the inner expectation is a conditional one, and the conditioning is on the characterization $\mathcal{Z}$. Treating $E[L(\theta) \mid \mathcal{Z}]$ as the new sample function, we expect it to be “smoother” than $L(\theta)$, and, in particular, continuous in θ . Then, under some additional conditions (comparable to those made in the development of IPA) the interchange of differentiation and expectation in (3) can be justified:

$$\nabla J(\theta) = E[\nabla E[L(\theta) \mid \mathcal{Z}]].$$

Letting $L_{\mathcal{Z}}(\theta) := E[L(\theta) \mid \mathcal{Z}]$, the SPA estimator of $\nabla J(\theta)$ is

$$[\nabla J(\theta)]_{SPA} = \nabla L_{\mathcal{Z}}(\theta). \quad (4)$$

Naturally, the idea is to minimize the amount of added information represented by $\mathcal{Z}$, since this incurs added costs we would like to avoid. The choice of the characterization $\mathcal{Z}$ generally depends on the sample function considered and the system under study.

A number of other extensions to IPA have also been developed (see Cassandras and Lafortune 2008). It is also worth mentioning that the PA approach can be applied to a parameter θ taking values from a finite set $\Theta = \{\theta_0, \theta_1, \dots, \theta_M\}$. The theory of *concurrent estimation* and *sample path constructability* provides techniques to estimate performance measures of the DES through

the process of constructing sample paths under each of $\theta_0, \theta_1, \dots, \theta_M$; for details see Cassandras and Lafortune (2008).

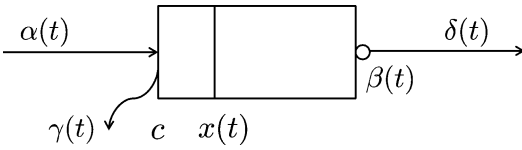
SFM Framework for IPA

The stochastic flow model (SFM) framework essentially consists of fluid queues which forego the notion of the individual customer and focus instead on the aggregate flow. In such a fluid queue, traffic and service processes are characterized by instantaneous flow rates as opposed to the arrival, departure, and service times of discrete customers. The SFM qualifies as a *stochastic hybrid system* with bilayer dynamics: discrete event dynamics at the upper layer and time-driven dynamics at the lower layer. The discrete events are associated with abrupt (discontinuous) changes in traffic-flow processes, such as the boundaries of busy periods at the queues. In contrast, the time-driven dynamics describe the continuous evolution of flow rates between successive discrete events, usually by differential equations or explicit functional terms. Performance metrics that are natural to SFMs typically reflect quantitative measures of flow rates, like average throughput, buffer workload, and loss.

Due to the smoothing effects of SFMs, they appear to provide a far more natural setting for IPA than their analogous discrete-queueing counterparts. Furthermore, their IPA gradients often are computable via extremely simple algorithms that are based entirely on the observed sample path. Consequently SFMs could, in principle, be implemented on the sample paths generated by an actual system rather than simulations thereof and thus be used in real-time optimization.

All of this will be explained via a concrete example of a queue which, though simple, captures the salient features of the SFM setting for IPA. For a more comprehensive discussion, please see (Cassandras et al. 2010; Wardi et al. 2010; Yao and Cassandras 2013).

Consider the fluid queue depicted in Fig. 3 whose input and output flow rate processes are denoted, respectively, by $\alpha(t)$ and $\delta(t)$. The output-flow process depends on the input-flow



Perturbation Analysis of Discrete Event Systems, Fig. 3 Basic SFM

process via the action of the server as well as the buffer size. The server is characterized by an instantaneous processing rate, denoted by $\beta(t)$, and the buffer size, namely, the maximum amount of fluid the buffer can hold, is denoted by c . Fluid overflow occurs when the inflow (arrival) rate exceeds the service rate while the buffer is full, and the overflow (loss) rate is denoted by $\gamma(t)$.

Suppose that $\alpha(t)$ and $\beta(t)$ are random functions defined on a suitable probability space, and assume that they are piecewise continuous and of bounded variation w.p.1. In order to describe their functional relations to $\delta(t)$ and $\gamma(t)$, we define the state variable to be the amount of fluid in the buffer (workload), and denote it by $x(t)$. The dynamics of the system evolve according to the following one-sided differential equation:

$$\frac{dx}{dt^+} = \begin{cases} 0, & \text{if } x(t) = 0 \text{ and } \alpha(t) \leq \beta(t) \\ 0, & \text{if } x(t) = c \text{ and } \alpha(t) \geq \beta(t) \\ \alpha(t) - \beta(t), & \text{otherwise,} \end{cases}$$

and $\delta(t)$ and $\beta(t)$ are related to $\alpha(t)$ and $\gamma(t)$ via

$$\delta(t) = \begin{cases} \beta(t), & \text{if } x(t) > 0 \\ \alpha(t), & \text{if } x(t) = 0, \end{cases}$$

and

$$\gamma(t) = \begin{cases} \alpha(t) - \beta(t), & \text{if } x(t) = c \\ 0, & \text{if } x(t) < c. \end{cases}$$

Network arrangements of such fluid queues, with specified routing and control schemes, provide a rich class of SFMs.

Let $\theta \in \mathcal{R}$ be a parameter of the inflow rate, the service rate, or a network control law; for instance, θ can be the *on* time of the flow from an off/on source, a uniform rate of the server, or the threshold level in a threshold-based flow control.

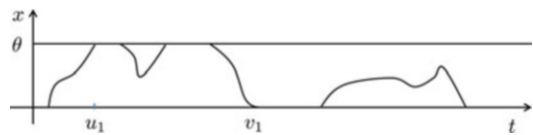
Then the aforementioned traffic processes are functions of θ and t and hence are denoted by $\alpha(\theta; t)$, $\beta(\theta; t)$, $x(\theta; t)$, etc. Fix the parameter θ , and consider the evolution of the system over a given time horizon, $[0, T]$. Performance measures of interest in applications include the average loss rate over the horizon $[0, T]$ and the average workload there which is related to the delay by Little's Law. Related to them are the sample performance functions $L_{\gamma, T}(\theta) := \int_0^T \gamma(\theta, t) dt$ and $L_{x, T}(\theta) := \int_0^T x(\theta, t) dt$; the former is called the *loss volume* and the latter the *cumulative workload*.

To illustrate the forms of their IPA derivatives, consider the basic SFM shown in Fig. 3, and let θ be its buffer size, namely, $\theta = c$. We say that a busy period of the queue is *lossy* if it incurs some loss at any amount. Let us denote by N_T the number of lossy busy periods in the horizon $[0, T]$. Then (see Cassandras et al. 2002), the IPA derivative of the loss volume has the following form:

$$L'_{\gamma, T}(\theta) = -N_T,$$

where again we use the prime symbol to denote derivative with respect to θ . This formula amounts to a counting process and indeed it is very simple. As an example, Fig. 4 depicts a typical state trajectory derived from a sample path. It is readily seen that the first busy period is lossy, while the second one is not, and therefore, $L'_{\gamma, T}(\theta) = -1$.

Concerning the cumulative workload, suppose that the queue has M lossy busy periods in the time interval $[0, T]$, and let us enumerate them by the counter $m = 1, \dots, M$. Moreover, denote by u_m the first time the buffer becomes full in its m th lossy busy period and by v_m the end time of that busy period. Then (see Cassandras et al. 2002),



Perturbation Analysis of Discrete Event Systems, Fig. 4 State trajectory of the SFM

$$L'_{x,T}(\theta) = \sum_{m=1}^M (v_m - u_m).$$

In the example provided by Fig. 4, $L'_{x,T}(\theta) = v_1 - u_1$.

These equations for the IPA derivatives not only are very simple but require no knowledge of the specific form of the processes $\{\alpha(t)\}$ or $\{\beta(t)\}$, their realizations, or underlying probability law. They depend only on limited information which is directly observed from the sample path and hence are said to be *nonparametric* or *model-free*. Furthermore, they have been shown to possess considerable robustness to modeling variations that do not cause significant alterations of the busy periods of the queue. A case of interest is when the SFM formalism is used as an abstraction of a queue. Then the above IPA formulas that were derived from an analysis of the SFM can be successfully applied to sample paths that are generated from the discrete queue. This is in contrast to the IPA formulas that are derived from the discrete queue, which generally are highly biased.

All of these properties of IPA in the SFM setting, including its unbiasedness, simplicity, the nonparametric nature of its algorithms, and its robustness to model variations, have had extensions to SFM networks and systems beyond the basic model (see Cassandras et al. 2010; Wardi et al. 2010; Yao and Cassandras 2013). As mentioned above, this suggests the potential application of IPA not only in system optimization via off-line simulation, but also in real-time control where the sample paths are generated from the actual system.

Summary and Current Research Directions

This paper narrates the evolution of fundamental ideas underscoring the development of perturbation analysis since its inception in 1979 (Ho et al. 1979). Introduced to solve a specific design-optimization problem in transfer

lines, PA has evolved into a framework for sample-path sensitivity analysis of stochastic hybrid dynamical systems. Its main focus has been on infinitesimal perturbation analysis, which gives gradient estimators of expected-value performance functions that can be naturally used in optimization. During the 1980s and 1990s, the main theoretical problem regarding IPA concerned its unbiasedness and strong consistency, thought to be essential conditions for its effective use in optimization. Various extensions of the basic IPA technique have been developed, including different sample-path representations of a given performance function and alternative approximate-modeling frameworks for DES by SFM and their generalizations to stochastic hybrid systems.

In the past 15 years, the focus of research on PA has shifted to large-scale, complex systems. They include large-scale Markov systems with applications to learning, Markov decision processes, and other complex stochastic optimization problems (e.g., Cao 2007, 2015) and decentralized control and event-based control systems with applications to multi-agent networks and cyber-physical systems (e.g., Zhong and Cassandras 2010; Cassandras et al. 2013). A recent historical survey of PA (Wardi et al. 2018) provides a detailed description of the evolution of the IPA technique and a more-comprehensive exposition of current research directions.

Cross-References

- [Models for Discrete Event Systems: An Overview](#)
- [Perturbation Analysis of Steady-State Performance and Relative Optimization](#)

Bibliography

- Cao X-R (1985) Convergence of parameter sensitivity estimates in a stochastic experiment. *IEEE Trans Autom Control* 30:834–843

- Cao X-R (2007) Stochastic learning and optimization: a sensitivity-based approach. Springer, Boston
- Cao X-R (2015) Optimization of average rewards of time nonhomogeneous Markov chains. *IEEE Trans Autom Control* 60:1841–1865
- Cassandras CG, Lafortune S (2008) Introduction to discrete event systems, 2nd edn. Springer, New York
- Cassandras CG, Wardi Y, Melamed B, Sun G, Panayiotou CG (2002) Perturbation analysis for on-line control and optimization of stochastic fluid models. *IEEE Trans Autom Control* 47:1234–1248
- Cassandras CG, Wardi Y, Panayiotou CG, Yao C (2010) Perturbation analysis and optimization of stochastic hybrid systems. *Eur J Control* 16: 642–664
- Cassandras, CG, Lin, X, Ding XC (2013) An optimal control approach to the multi-agent persistent monitoring problem. *IEEE Trans Autom Control* 58: 947–961
- Glasserman P (1991) Gradient estimation via perturbation analysis. Kluwer, Boston
- Gong WB, Ho YC (1987) Smoothed perturbation analysis of discrete event systems. *IEEE Trans Autom Control* 32:858–866
- Heidelberger P, Cao X-R, Zazanis M, Suri R (1988) Convergence properties of infinitesimal perturbation analysis estimates. *Manag Sci* 34: 1281–1302
- Ho YC, Cao, X-R (1991) Perturbation analysis of discrete event dynamical systems. Kluwer, Boston
- Ho YC, Cassandras CG (1983) A new approach to the analysis of discrete event dynamic systems. *Automatica* 19:149–167
- Ho YC, Eyster MA, Chien DT (1979) A gradient technique for general buffer storage design in a serial production line. *Int J Prod Res* 17:557–580
- Ho YC, Cao X-R, Cassandras CG (1983) Infinitesimal and finite perturbation analysis for queueing networks. *Automatica* 19:439–445
- Wardi Y, Adams R, Melamed B (2010) A unified approach to infinitesimal perturbation analysis in stochastic flow models: the single-stage case. *IEEE Trans Autom Control* 55:89–103
- Wardi Y, Cassandras CG, Cao XR (2018) Perturbation analysis: a framework for data-driven control and optimization of discrete event and hybrid systems. *Annu Rev Control* 45:257–266. Special Issue on History of Discrete Event Systems, Ed. M. Silva
- Yao C, Cassandras CG (2013) Perturbation analysis and optimization of multiclass multiobjective stochastic flow models. *Discret Event Dyn Syst* 21: 219–256
- Zhong M, Cassandras CG (2010) Asynchronous distributed optimization with event-driven communication. *IEEE Trans Autom Control* 55: 2735–2750

Perturbation Analysis of Steady-State Performance and Relative Optimization

Xi-Ren Cao

Department of Automation, Shanghai Jiao Tong University, Shanghai, China

Institute of Advanced Study, Hong Kong University of Science and Technology, Hong Kong, China

Abstract

We introduce some special theories and methodologies for perturbation analysis (PA) and optimization of steady-state performance and their extensions. Such theories and methodologies utilize the special features of a dynamic systems and usually take different perspectives from the traditional optimization approaches, and therefore they may lead to new insights, new results, and efficient algorithms. The topics discussed include the gradient-based optimization for system with continuous parameters and the direct-comparison-based optimization for systems with discrete policies. Both constitute the relative optimization approach, an alternative to dynamic programming. This approach also applies to continuous-time and continuous-state dynamic systems, leading to a new paradigm of stochastic control.

Keywords

Gradient estimation · Sample-path techniques · Sensitivity analysis · Queueing networks · Relative optimization · Markov decision processes · Stochastic control

Introduction

In this entry, we introduce some special theories and methodologies for perturbation analysis (PA) and optimization of steady-state performance and their extensions. Such theories and methodologies utilize the special

features of a dynamic systems and usually take different perspectives from the traditional optimization approaches, and therefore they may lead to new insights, new results, and efficient algorithms. The topics discussed include the gradient-based optimization for system with continuous parameters and the direct-comparison-based optimization for systems with discrete policies. Both constitute the so-called relative optimization approach, an alternative to dynamic programming. This approach also applies to continuous-time and continuous-state dynamic systems, leading to a new paradigm of stochastic control.

As discussed in ▶ “[Perturbation Analysis of Discrete Event Systems](#),” perturbation analysis (PA) can be applied to both finite-period and steady-state performance. This chapter will mainly focus on the latter and a related topic, the relative optimization, which includes the gradient-based optimization for systems with continuous parameters and the comparison-based optimization for systems with discrete parameters or policies.

Gradient-Based Approaches

Basic Ideas

The gradient-based performance optimization of discrete event dynamic systems (DEDSs) consists of three steps:

1. Developing efficient algorithms to estimate the performance gradients using the special features of a DEDS
2. Studying the properties of the gradient estimates, including investigating whether they are unbiased and/or strongly consistent
3. With the gradient estimates, developing efficient optimization algorithms

Steps 1 and 2 are referred to as PA, and Step 3 is usually done together with standard gradient-based optimization approaches, such as hill-climbing type of approaches and stochastic approximation approaches such as the Robbins-Monro algorithm (Robbins and Monro [1951](#)).

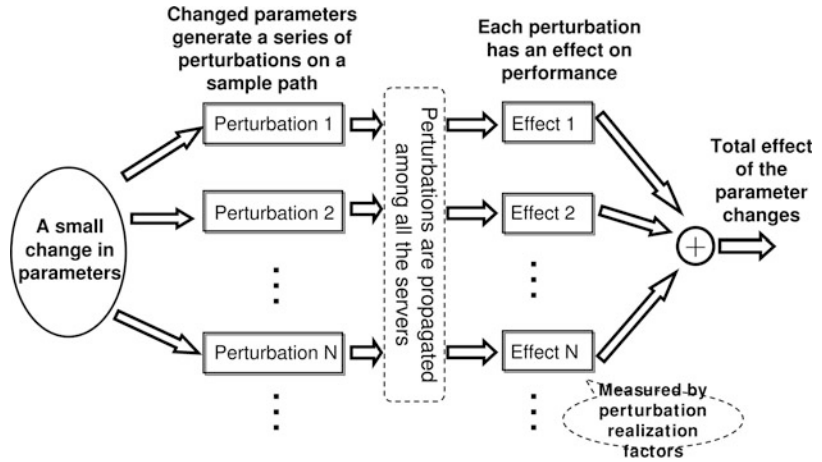
Our focus here is on PA for steady-state performance. The main principle for estimating the gradients of steady-state performance is decomposition. In a DEDS, a small change in the value of a system parameter induces a series of changes on the system’s sample path; each of such changes is called a *perturbation*. A single perturbation alone will affect the sample path and therefore affect the system performance. Such an effect is typically small, and therefore, the linear superposition usually holds. Thus, the effect of a small parameter change on the steady-state performance can be determined by summing up the effects of all the perturbations induced (or generated) by the parameter change. This principle is illustrated by Fig. 1, and it applies to many different systems with different performance criteria. Following the principle, efficient algorithms can be developed, and their strong consistency can be proved (Cao [2007](#)). Its application to queueing systems and Markov systems will be discussed in the following two subsections.

Queueing Systems

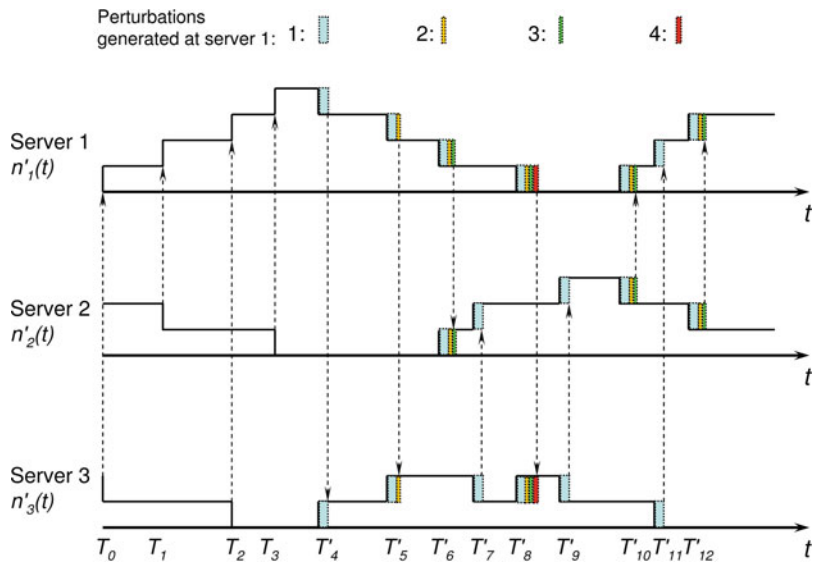
The gradient-based approach for DEDSs starts with queueing systems, known as infinitesimal perturbation analysis (IPA), or simply PA, ▶ “[Perturbation Analysis of Discrete Event Systems](#).” Queueing systems are widely used as a model for many DEDSs in literature and have very unique structural features, and PA utilizes such special features to develop fast algorithms to estimate the performance gradients.

We first give a brief explanation for the simple rules of PA of queueing systems (Ho and Cao [1983, 1991](#)). Consider a closed Jackson network with M servers and N customers. The service times of customers at server i are exponentially distributed with service rate μ_i , $i = 1, 2, \dots, M$. After a customer completes its service at server i , it goes to server j with a routing probability q_{ij} , $i, j = 1, 2, \dots, M$. The service discipline is first come first served. The number of customers at server i is denoted as n_i , and the system state is denoted as $\mathbf{n} = (n_1, n_2, \dots, n_M)$. The state process is $\mathbf{n}(t) = (n_1(t), n_2(t), \dots, n_M(t))$ with $n_i(t)$ being the number of customers at

Perturbation Analysis of Steady-State Performance and Relative Optimization,
Fig. 1 The decomposition of performance changes



Perturbation Analysis of Steady-State Performance and Relative Optimization,
Fig. 2 Perturbation generation and propagation in a queueing network



server i at time t , $i = 1, 2, \dots, M$, $t \geq 0$. Define T_l as the l th state transition time of the process $\mathbf{n}(t)$.

Figure 2 illustrates an example of a sample path where each stair-style line represents a trajectory of the number of customers at one server, and the customer transitions among servers are indicated by dotted arrows.

An exponentially distributed service time with rate μ can be generated according to $s = -\frac{1}{\mu} \ln \zeta$, where ζ is a uniformly distributed random number in $[0, 1]$. Now let the service rate of a server, say server k , change from μ_k to $\mu_k + \Delta\mu_k$, $\Delta\mu_k < \mu_k$. Then the service time s will change to $s' = -\frac{1}{\mu_k + \Delta\mu_k} \ln \zeta$ (the same ζ is

used for both s and s'). Thus, the perturbation of the service time induced by the change in μ_k is

$$\Delta s = s' - s \approx -\frac{\Delta\mu_k}{\mu_k} s. \quad (1)$$

In summary, because of a small (infinitesimal) change of service rate $\Delta\mu_k$, every customer's service completion time at server k will be delayed by a small amount Δs that is proportional to its original service time with a multiplier $-\frac{\Delta\mu_k}{\mu_k}$; or, we say that server k obtains a perturbation of Δs . This is the rule of *perturbation generation*.

Next, we observe that a perturbation will affect the service starting and completion times of other customers in the same server and in other

servers in the network. We say a perturbation will be *propagated* through the network. First, the perturbations generated at a server will accumulate until this server is idle. When the server enters an idle period, all its perturbations will be lost. (After the idle period, the service starting time is determined by the customer that terminates the idle period, which carries the perturbation of another server.) Second, when a customer finishes its service at server i and enters server j after an idle period of server j , server j will obtain the same amount of perturbations as server i . If server j is not idle at this time, the perturbation at server i will only affect the arrival time of this customer and will not affect any other customers/servers.

Figure 2 illustrates the perturbation generation and propagation process of a network in which the service rate of server 1 is decreased with an infinitesimal amount. Given a system sample path obtained by simulation or observation, we can record the perturbations of all servers as they are generated and propagated along the sample path. From the perturbations, we can get a perturbed sample path and finally get the performance changes caused by the change in the service rate. Specifically, we have the following algorithm for the perturbations of the service completion times (if server k 's service rate is perturbed).

Algorithm 1 Perturbation analysis

1. *Initialization:* Set variables $\Delta_i = 0, i = 1, 2, \dots, M$.
2. *Perturbation generation:* At the l th service completion time of server k , set $\Delta_k = \Delta_k + s_{k,l}, l = 1, 2, \dots$, where $s_{k,l}$ is the service time of the l th customer at server k .
3. *Perturbation propagation:* If a customer from server i terminates an idle period of server j , set $\Delta_j = \Delta_i, i, j = 1, 2, \dots, M$.

In Step 2, we add $s_{k,l}$ as the perturbation generated, instead of $-\frac{\Delta\mu_k}{\mu_k}s_{k,l}$ as indicated by (1). The factor $-\frac{\Delta\mu_k}{\mu_k}$ will be canceled when estimating performance derivatives with respect to $\Delta\mu_k$. The algorithm yields a perturbed sample path. With the original path and the perturbed path, we can estimate the original and perturbed (steady-state) throughput, and then the

estimates of its derivative with respect to μ_k can be obtained, and it is proved that the estimate is strongly consistent. Only a few lines need to be added in the simulation code to obtain the derivatives. Experiments show that the results are very accurate (error is around 5 %, compared with analytical results, Ho and Cao 1983).

Markov Systems

The decomposition principle shown in Fig. 1 has been applied to Markov systems (Cao 2007).

Consider an irreducible and aperiodic Markov chain $\mathbf{X} = \{X_n : n \geq 0\}$ on a finite state space $\mathcal{S} = \{1, 2, \dots, M\}$ with transition probability matrix $P = [p(j|i)]_{i,j=1}^{M,M} \in [0, 1]^{M \times M}$, with $p(j|i)$ being the transition probability from state i to state j . Let $\pi = (\pi_1, \dots, \pi_M)$ be the vector representing its steady-state probabilities and $f = (f_1, f_2, \dots, f_M)^T$ be the reward (or cost) vector, where “ T ” represents transpose. We have $Pe = e$, where $e = (1, 1, \dots, 1)^T$ is an M -dimensional vector whose all components equal 1, and we have $\pi = \pi P$. We consider the long-term average (steady-state) performance defined as

$$\begin{aligned} \eta &= E_\pi(f) = \sum_{i=1}^M \pi_i f_i \\ &= \pi f = \lim_{L \rightarrow \infty} \frac{F_L}{L}, \quad w.p.1, \end{aligned} \quad (2)$$

where

$$F_L = \sum_{l=0}^{L-1} f(X_l).$$

Let P' be another ergodic transition probability matrix on the same state space. Suppose P changes to $P(\delta) = P + \delta Q = \delta P' + (1 - \delta)P$, with $\delta > 0$, $Q = P' - P = [q(j|i)]$, and the reward function f keeps the same. We have $Qe = 0$. The performance measure will change to $\eta(\delta) = \eta + \Delta\eta(\delta)$. The derivative of η in the direction of Q is defined as $\frac{d\eta(\delta)}{d\delta} = \lim_{\delta \rightarrow 0} \frac{\Delta\eta(\delta)}{\delta}$.

In this discrete-state Markov system, a perturbation means that the system is perturbed from one state i to another state j . For example, consider the case where $q(i|k) = \frac{1}{2}, q(j|k) = \frac{1}{2}$, and

$q(l|k) = 0$ for all $l \neq i, j$. Suppose that these probabilities change to $q(i|k) = \frac{1}{2} + \delta$, $q(j|k) = \frac{1}{2} - \delta$, and $q'(l|k) = 0$ for all $l \neq i, j$. Then it may happen that at some time in the original sample path, the system transits from state k to state i , but in the perturbed path, it transits from state k to state j instead. In this case, we say that the change in transition probabilities induces a perturbation from i to j at this time. To study the effect of such a perturbation, we consider two independent sample paths $\mathbf{X} = \{X_n; n \geq 0\}$ and $\mathbf{X}' = \{X'_n; n \geq 0\}$ with $X_0 = i$ and $X'_0 = j$; both of them have the same transition matrix P . The average effect of a perturbation from i to j , $i, j = 1, \dots, M$, on F_L can be measured by the *perturbation realization factor* defined as

$$d(i, j) = \lim_{L \rightarrow \infty} E \left[\sum_{l=0}^{L-1} (f(X'_l) - f(X_l)) | X_0 = i, X'_0 = j \right]. \quad (3)$$

The matrix $D \in \mathcal{R}^{M \times M}$, with $d(i, j)$ as its (i, j) th element, is called a *realization matrix*. From (3), it is easy to prove that D satisfies the equation (Cao 2007)

$$D - PDP^T = F, \quad (4)$$

where $F = fe^T - ef^T$. Because $d(i, j) = -d(j, i)$, for any i, j , and $d(i, k) = d(i, j) + d(j, k)$ for any i, j , and k , we may define a vector $g = (g(1), \dots, g(M))^T$ such that

$$d(i, j) = g(j) - g(i), \quad (5)$$

and $D = ge^T - eg^T$. g is called a *performance potential*, which can be estimated with many sample path-based algorithms, and it satisfies the Poisson equation (Cao 2007)

$$(I - P + e\pi)g = f. \quad (6)$$

Intuitively, (3) and (5) indicate that every visit to state i contributes to F_L on the average by the amount of $g(i)$, so the effect of a perturbation from i to j is $d(i, j) = g(j) - g(i)$. Now, we consider a sample path consisting of L

transitions. Among these transitions, on the average there are $L\pi_i$ transitions at which the system is at state i . After being at state i , the system jumps to state j on the average $L\pi_i p(j|i)$ times. If the transition probability matrix P changes to $P(\delta) = P + \delta Q$, then the change in the number of visits to state j after being at state i is $L\pi_i q(j|i)\delta = L\pi_i [p'(j|i) - p(j|i)]\delta$. This contributes a change of $\{L\pi_i [p'(j|i) - p(j|i)]\delta\}g(i)$ to F_L . Thus, the total change in F_L due to the change of P to $P(\delta)$ is

$$\begin{aligned} \Delta F_L &= \sum_{i,j=1}^M L\pi_i [p'(j|i) - p(j|i)]\delta g(i) \\ &= L(\pi Qg)\delta. \end{aligned}$$

Finally, we have

$$\frac{d\eta}{d\delta} = \lim_{\delta \rightarrow 0} \frac{1}{\delta} \frac{\Delta F_L}{L} = \pi Qg = \pi(P' - P)g. \quad (7)$$

This equation can be easily proved by the Poisson equation (6). The above reasoning gives an intuitive explanation. If the reward function also changes from f to $f(\delta) = f + \delta(f' - f)$, then (7) becomes

$$\frac{d\eta}{d\delta} = \pi[(P'g + f') - (Pg + f)]. \quad (8)$$

Further Works

The ideas described in the previous subsections may stimulate new research topics in theoretical analysis, estimation algorithms, and applications. Here we can only give a very brief review for some of them.

1. There is a large literature on whether the PA-based derivative estimates are unbiased (finite period) and/or strongly consistent (steady state). This was first formulated in Cao (1985) and was further discussed in Heidelberger et al. (1988). By now, there have been extensive studies in this direction: proving the unbiasedness or consistency for various systems and modifying the approach for system when the IPA estimates are not unbiased (e.g., Cassandras and Lafortune

1999; Glasserman 1991; Fu and Hu 1997). This also includes the recently proposed fluid model; see ▶ “[Perturbation Analysis of Discrete Event Systems](#).”

2. Another research topic is how to develop fast and efficient algorithms for estimating the performance gradients, especially in the case of Markov systems; see, e.g., Cao and Wan (1998), Baxter and Bartlett (2001), and Cao (2005). This is called *policy gradients* in the reinforcement learning literature.
3. There are also research works on how the gradient estimates and policy iteration (see section “[Direct Comparison and Policy Iteration](#)”) can be combined with stochastic approximation approaches to develop fast convergent optimization algorithms; see Marbach and Tsitsiklis (2001) for gradient-based approach and Fang and Cao (2004) for policy iteration-based approach.

Direct Comparison and Policy Iteration

The sensitivity-based view has been extended to optimization in discrete spaces of policies. With this view, we can develop a new approach to performance optimization based on a direct comparison of the performance of any two policies. This provides an alternative to the standard dynamic programming to solving the Markov decision processes (MDP) types of problems; it also has been applied to solve some problems when the standard MDP fails (Cao 2007; Cao and Wan 2017).

In an MDP, there is an action space denoted as $\mathcal{A}$. For simplicity, we only consider the discrete case. When the system is at any state $i \in \mathcal{S}$, an action $\alpha = d(i)$ is taken, which controls the system transition probability, denoted as $p^\alpha(j|i)$, $i, j \in \mathcal{S}$, and the reward function, denoted as $f(i, \alpha)$. The mapping $d : \mathcal{S} \rightarrow \mathcal{A}$ is called a *policy*. Since a policy corresponds to a transition probability matrix $P^d = [p^{d(i)}(j|i)]_{i,j=1}^M$ and the reward vector $f^d = (f(1, d(1)), \dots, f(M, d(M)))^T$, we also call the pair (P, f) a policy.

Consider two policies (P, f) and (P', f') , and assume that the Markov chain under both policies is ergodic. We use prime “'” to denote the quantities associated with (P', f') . First, multiplying both sides of the Poisson equation (6) with π' on the left and after some calculations, we get

$$\eta' - \eta = \pi' \{ (P'g + f') - (Pg + f) \}. \quad (9)$$

This is called a *performance difference formula*, and many optimization results can be derived from it in an intuitive way.

Policy Iteration and the Optimality Equation

The difference formula (9) has a nice decomposition structure: it contains two factors, the first one is π' , which does not depend on P , and the second one is $(P'g + f') - (Pg + f)$, in which all the parameters are known except the performance potential g , which can be obtained by only analyzing system with P . This nice feature makes the difference formula the basis of performance optimization.

For two M -dimensional vectors a and b , we define $a = b$ if $a(i) = b(i)$ for all $i = 1, 2, \dots, M$; $a \leq b$ if $a(i) \leq b(i)$ for all $i = 1, 2, \dots, M$; $a < b$ if $a(i) < b(i)$ for all $i = 1, 2, \dots, M$; and $a \leq b$ if $a(i) < b(i)$ for at least one i and $a(j) = b(j)$ for other components. The relation $\leq$ includes $=$, $\leq$, and $<$. Similar definitions are used for the relations $>$, $\geq$, and $\geq$.

Next, we note that $\pi'(i) > 0$ for all $i = 1, 2, \dots, M$. Thus, from (9), we know that if $(P' - P)g + (f' - f) \geq 0$, then $\eta' - \eta > 0$. From (9) and the fact $\pi' > 0$, the proof of the following lemma is straightforward.

Lemma 1 *If $Pg + f \leq (\leq) P'g + f'$, then $\eta < (\leq) \eta'$.*

It is interesting to note that in the lemma, we use only the potentials with one Markov chain, i.e., g . Thus, because of the special structure of the performance difference formula (9), if the condition in Lemma 1 holds, to compare the performance measures under two policies, we may only need the potentials with one policy.

Policy iteration and the optimality equation can be easily derived from (9) and Lemma 1. A

policy iteration algorithm is illustrated in Algorithm 2.

Algorithm 2 Policy iteration

1. Guess an initial policy d_0 , and set $k = 0$.
2. (Policy evaluation) Obtain the potential g^{d_k} by solving the Poisson equation $(I - P^{d_k})g^{d_k} + \eta^{d_k}e = f^{d_k}$ or estimating it on a sample path. (The superscript “ d_k ” is added to quantities associated with policy d_k .)
3. (Policy improvement) Choose

$$d_{k+1} \in \arg \left\{ \max_{d \in \mathcal{D}} [f^d + P^d g^{d_k}] \right\}, \quad (10)$$

component-wise (i.e., to determine an action for each state). If in state i , action $d_k(i)$ attains the maximum, and set $d_{k+1}(i) = d_k(i)$.

4. If $d_{k+1} = d_k$, stop; otherwise, set $k := k + 1$ and go to Step 2.
-

It follows directly from Lemma 1 that when the iteration does not stop, the performance improves at each iteration. It can be proved easily by construction that the iteration stops at an optimal policy. Again, from Lemma 1, when the iteration stops at a policy $\hat{d}$, it holds

$$\eta^{\hat{d}}e + g^{\hat{d}} = \max_{d \in \mathcal{D}} \{f^d + P^d g^{\hat{d}}\}. \quad (11)$$

This is the Hamilton-Jacobi-Bellman (HJB) optimality equation. If we further define the Q-factor,

$$Q^d(i, \alpha) = f(i, \alpha) + \sum_j p^\alpha(j|i)g^d(j). \quad (12)$$

Then the policy iteration equation (10) and the HJB equation (11) become

$$d_{k+1}(i) := \arg \max_{\alpha \in \mathcal{A}} \{Q^{d_k}(i, \alpha)\} \quad (13)$$

and

$$Q^{\hat{d}}(i, \alpha) = \max_{\beta \in \mathcal{A}} \{Q^{\hat{d}}(i, \beta)\}, \quad \alpha = \hat{d}(i). \quad (14)$$

The only difference between (8) and (9) is that π in (8) is replaced by π' in (9). This leads to an interesting observation: *policy iteration in*

MDPs in fact chooses the policy with the steepest directional derivative as the policy in the next iteration. Therefore, policy iteration in fact can be viewed as the “gradient-based” optimization in a discrete space.

In our approach, the HJB equation and policy iteration are obtained from the performance difference equation (9), which compares the performance of any two policies. It is hence called a *direct-comparison*-based approach, which, together with the gradient-based approach, constitutes the *relative optimization* approach. This approach also applies to more general problems, such as multichain Markov systems, systems with absorbing states, and problems with other performance criteria such as the discounted performance and the bias, etc.

The relative optimization approach has been successfully applied to the n th-bias optimality problem (Cao 2007). Essentially, starting with the performance difference formulas (those similar to (9) for different performance), we can develop a simple and direct approach to derive the results that are equivalent to the sensitive discount optimality for multichain Markov systems with long-run average criteria (Puterman 1994), and no discounting is needed and no dynamic programming is used. The approach, motivated by the development for discrete event dynamic systems, provides a clear overall picture for the area of MDP.

The relative optimization approach also applies to performance optimization of dynamic systems with continuous-time and continuous-states and hence provides a new framework to stochastic control and leads to some new results. The approach can also be applied to some problems where dynamic programming fails, including the event-based optimization problems, where the sequence of events may not be Markovian; see the next two sections.

Event-Based Optimization

It is well-known that for most systems modeled by Markov processes, the state spaces are too large, and it is not computationally feasible

to implement policy iteration or to solve the HJB equations. On the other hand, in many practical problems in engineering, finance, and social sciences, control actions are only taken when certain events occur. For example, in traffic control of a network of subnetworks, oftentimes one cannot control the traffic in the same subnetwork, and control actions are only applied when there are packets transferring among different subnetworks. In a portfolio management problem, the investor sells or buys stocks when the price history experiences some predetermined patterns (e.g., reaches some level). In a sensor network, actions are taken only when one of the sensors detects some abnormal situations. In a material handling problem, actions are taken when the inventory level falls below certain threshold.

Conceptually, anything happened in the past can be chosen as an event. However, the number of such events is too big (much bigger than the number of states), and studying all these events makes analysis infeasible and defeats our original purpose. Therefore, to properly define events, we need to strike a balance between the generality on the one hand and the applicability on the other hand.

In the event-based setting, an event e is defined as a set of state transitions with certain common properties. That is, $e := \{(i, j) : i, j \in \mathcal{S} \text{ and } \langle i, j \rangle \text{ has common properties}\}$, where $\langle i, j \rangle$ denotes a state transition from i to j . This definition can also be easily generalized to represent a finite sequence of state transitions. We shall see that in many real problems, the number of events requiring control actions is usually much smaller than that of the states.

An event-based policy d is defined as a mapping from $\mathcal{E}$ to $\mathcal{A}$, with $\mathcal{E}$ being the space of all events. That is, $d : \mathcal{E} \rightarrow \mathcal{A}$. When an event $e \in \mathcal{E}$ happens, we choose an action $a = d(e)$ according to a policy d . Let $\mathcal{D}_e$ denote the set of all the stationary and deterministic event-based policies. The reward function under policy d is denoted as $f^d = f(i, d(e))$, and the associated long-run average performance is denoted as η^d . Our goal is to find an optimal policy $\hat{d}$ which maximizes the long-run average performance as follows:

$$\hat{d} = \arg \max_{d \in \mathcal{D}_e} \left\{ \eta^d \right\} \quad (15)$$

The main difficulty in developing the event-based optimization theories and algorithms lies in the fact that the sequence of events is usually not Markovian. The standard optimization approach such as dynamic programming does not apply to such problems. However, we can apply the relative optimization approach to this event-based optimization problem. Here we give a very brief discussion (Cao 2007; Xia et al. 2014).

Consider an event-based policy d , $d \in \mathcal{D}_e$. When event e occurs, the conditional transition probability is denoted as $p^{d(e)}(j|i, e)$. Let $\pi^d(i|e)$ be the conditional steady-state probability of state i when event e occurs under policy d . Define the aggregated Q-factor

$$Q^d(e, \alpha) = \sum_i \pi^d(i|e) \times \left[f(i, \alpha) + \sum_j p^\alpha(j|i, e) g^d(j) \right]. \quad (16)$$

We may use these aggregated Q-factors to develop an event-based policy iteration algorithms as Algorithm 2 and obtain the policy iteration equation (cf. (13))

$$d_{k+1}(e) := \arg \max_{\alpha \in \mathcal{A}} \{ Q^{d_k}(e, \alpha) \}, \quad e \in \mathcal{E}, \quad (17)$$

and the event-based HJB optimality equation (cf. (14)) is

$$Q^{\hat{d}}(e, \alpha) = \max_{\beta \in \mathcal{A}} \{ Q^{\hat{d}}(e, \beta) \}, \quad \alpha = \hat{d}(e). \quad (18)$$

It can be proved that if the conditional probability $\pi^h(i|e)$ does not depend on the policy, i.e.,

$$\pi^h(i|e) = \pi^d(i|e), \quad \forall i, e, \quad \text{and } h, d, \quad (19)$$

then policy iteration (17) indeed leads to a sequence of increasing performance and (18) specifies an event-based optimal policy.

The event-based aggregated Q-factor (16) can be estimated on a sample path in the same way as for potentials (Cao 2007; Xia et al. 2014). The number of events requiring actions is usually

much smaller than the number of states, and in some cases such as the network admission control problem, it is linear to the system size.

The crucial condition for the above event-based optimization is (19). There are many problems, such as the control of the networks of networks and the portfolio management problem, for which the condition holds; there are also many problems for which the condition does not hold. The error in applying (18) and policy iteration (17) comes from the difference between $\pi^h(i|e)$ and $\pi^d(i|e)$ at each iteration. Further research is needed.

We use a computer network as an example, in which each computer or router can be modeled as a queueing subnetwork and the computer network is then a network of such subnetworks. Assume that there are M subnetworks, and subnetwork m , $m = 1, 2, \dots, M$, consists of k_m servers. The number of customers at the j th server of subnetwork m is denoted as $n_{m,j}$, $j = 1, 2, \dots, k_m$, and the number of customers in all the servers in subnetwork m is denoted as $N_m = \sum_{j=1}^{k_m} n_{m,j}$. Suppose the service time is exponentially distributed, then the system state is $\mathbf{n} := (n_{1,1}, \dots, n_{1,k_1}; \dots; n_{M,1}, \dots, n_{M,k_M})$, and define the aggregated state as $\mathbf{N} := (N_1, \dots, N_M)$. Suppose that the transition probabilities among the servers in the same subnetwork are fixed and we can only control the transition probabilities among the subnetworks, and furthermore, we can only observe $\mathbf{N}$. Then the problem can be modeled as an event-based optimization with an event being a customer transition among subnetworks, and meanwhile, the aggregated state is $\mathbf{N}$. It can be proved that the condition (19) holds, and therefore, we may apply policy iteration (17) and HJB equation (18) (Xia et al. 2014).

Many existing problems fit the event-based framework. For example, in a partially observable Markov decision process (POMDP), we may define an observation, or a sequence of observations, as an event. Other examples include state and time aggregations, hierarchical control (hybrid systems), and options. Different events can be defined to capture the special features in these different problems. In this sense, the

event-based approach may provide a unified view to these different problems (Cao 2007; Xia et al. 2014).

The Relative Optimization Approach, Stochastic Control, and Others

We refer to the approaches discussed in the previous sections the *relative optimization* approach. The name comes from the fact that only the relative values of the performance measures are used in optimization. When the parameters are continuous, it is the gradient-based approach, and the focus is on developing efficient algorithms that utilize the particular system structure to estimate the gradients (to justify the algorithms, the unbiasedness or the consistency of the estimates should be considered). When the parameters are discrete, it is the direct-comparison approach that leads to policy iteration and the HJB equations, and policy iteration can be viewed as the gradient-based method in discrete spaces. This approach is different from the conventional dynamic programming, and it has been successfully applied to MDP with different criteria, the n -bias optimization problems, and time nonhomogeneous Markov chains (Cao 2015) and some other problems that dynamic programming may not work well.

The relative optimization approach was motivated by the study of discrete event dynamic systems (DEDS). It has been realized that the principles and methodologies developed for DEDSs also apply to the optimization of continuous-time and continuous-state (CTCS) systems, and therefore, the approach provides to a new framework for the area of stochastic control, and some new results can be obtained.

In CTCS systems, the dynamic is driven by Brownian motions or Levy processes. A typical CTCS Markov process is the Ito diffusion process defined by the stochastic differential equation. The transition probability matrix in a DEDS should be replaced by the infinitesimal generator in a CTCS system, which is an operator on the space of continuous functions. With the perfor-

mance difference formulas, we can redevelop the stochastic optimal control theory for many performance measures, including the long-run average, discounted performance, and finite horizon problems, with no dynamic programming. In addition, new results can be obtained. For example, explicit optimality conditions can be derived at non-smooth points of a value function, and viscosity solution to the HJB equation is not needed; multi-class CTCS processes can be defined, and optimality conditions can be derived for such processes. See Cao (2017, 2019) for more details.

Other applications of the relative optimization approach include the *time-inconsistent optimization problems*. In behavioral finance, people's preference is modeled with a distorted probability. For example, a risk-taking person buys lotteries, because in her/his mind, she/he enlarges the possibility of winning a large sum, and a risk-averse person buys insurance, because she/he is afraid of a big loss and therefore enlarges its probability.

The optimization problem with a distorted probability suffers from the time-inconsistent issue; i.e., an optimal policy for the problem in period $[t, T)$, $0 < t < T$, is not optimal in the same period for the problem in $[0, T]$. Thus, the standard dynamic programming fails.

The gradient-based approach has been applied to the portfolio management problem with probability distortion. With this approach, we discovered that the performance with distorted probability maintains some sort of linearity called *monolinearity*. This property shed new insights to the portfolio management problem and the nonlinear expected utility theory (Cao and Wan 2017).

Conclusion

The relative optimization approach has been developed to the performance optimization of both DEDSs and CTCS systems. The approach utilizes the dynamic structure of a system. For systems with continuous parameters, it is the gradient-based optimization, in which the special feature of the system helps in developing

efficient algorithms to estimate the performance derivatives; for systems with discrete policies, it is the direct-comparison-based approach, with which policy iteration and HJB equations can be derived intuitively by using the performance difference formulas. Policy iteration can be viewed as the gradient method in a discrete space. The estimation of gradients and the implementation of policy iteration can be carried out on a given sample path, and efficient online learning algorithms can be developed (Cao 2007). The approach provides an alternative to the traditional dynamic programming and therefore can be applied to some problems where dynamic programming does not work.

Cross-References

- [Models for Discrete Event Systems: An Overview](#)
- [Perturbation Analysis of Discrete Event Systems](#)

Acknowledgments This research was supported in part by the Collaborative Research Fund of the Research Grants Council, Hong Kong Special Administrative Region, China, under Grant No. HKUST11/CRF/10 and 610809.

Bibliography

- Baxter J, Bartlett PL (2001) Infinite-horizon policy-gradient estimation. *J Artif Intell Res* 15:319–350
- Cao XR (1985) Convergence of parameter sensitivity estimates in a stochastic experiment. *IEEE Trans Autom Control* 30:834–843
- Cao XR (2005) A basic formula for online policy gradient algorithms. *IEEE Trans Autom Control* 50(5):696–699
- Cao XR (2007) *Stochastic learning and optimization – a sensitivity-based approach*. Springer, New York
- Cao XR (2015) Optimization of average rewards of time nonhomogeneous Markov chains. *IEEE Trans Autom Control* 60:1841–1856
- Cao XR (2017) Optimality conditions for long-run average rewards with under selectivity and non-smooth features. *IEEE Trans Autom Control* 62:4318–4332
- Cao XR (2019) State classification and multi-class optimization of continuous-time and continuous-state Markov processes. *IEEE Trans Autom Control* 64:3632–3646

- Cao XR, Wan YW (1998) Algorithms for sensitivity analysis of Markov systems through potentials and perturbation realization. *IEEE Trans Control Syst Technol* 6:482–494
- Cao XR, Wan XW (2017) Sensitivity analysis of nonlinear behavior with distorted probability. *Math Financ* 27:115–150
- Cassandras CG, Lafortune S (1999) Introduction to discrete event systems. Kluwer Academic Publishers, Boston
- Fang HT, Cao XR (2004) Potential-based on-line policy iteration algorithms for Markov decision processes. *IEEE Trans Autom Control* 49:493–505
- Fu MC, Hu JQ (1997) Conditional Monte Carlo: gradient estimation and optimization applications. Kluwer Academic Publishers, Boston
- Glasserman P (1991) Gradient estimation via perturbation analysis. Kluwer Academic Publishers, Boston
- Heidelberger P, Cao XR, Zazanis M, Suri R (1988) Convergence properties of infinitesimal perturbation analysis estimates. *Manag Sci* 34:1281–1302
- Ho YC, Cao XR (1983) Perturbation analysis and optimization of queueing networks. *J Optim Theory Appl* 40:559–582
- Ho YC, Cao XR (1991) Perturbation analysis of discrete-event dynamic systems. Kluwer Academic Publisher, Boston
- Marbach P, Tsitsiklis JN (2001) Simulation-based optimization of Markov reward processes. *IEEE Trans Autom Control* 46:191–209
- Puterman ML (1994) Markov decision processes: discrete stochastic dynamic programming. Wiley, New York
- Robbins H, Monro S (1951) A stochastic approximation method. *Ann Math Stat* 22:400–407
- Xia L, Jia QS, Cao XR (2014) A tutorial on event-based optimization – a new optimization framework. *Discret Event Dyn Syst Theory Appl Invited Paper* 24:103–132

can physically interact with humans to collaborate with them in the execution of several tasks in different application domains, from manufacturing to healthcare, from inspection to maintenance. This chapter reviews all such technologies, which range from innovative actuation systems, such as series elastic actuators or highly integrated torque controlled joints, to sensing systems, such as joint torque sensors and force/tactile devices or 3D vision systems for tracking of humans acting in the robot workspace. Another major technological advancement that enabled safe physical human-robot interaction (pHRI) is the compliance control of robotic manipulators in conjunction with smart collision detection and reaction strategies to be adopted in case of accidental human-robot contacts. Programming of robots that are allowed to physically interact with humans is by far more intuitive than standard programming procedures and, again, this is enabled by smart technologies like haptic gestures recognition in combination with advanced torque control or tactile skin devices.

Keywords

Safety · Soft robotics · Cobots · Collision detection and reaction

Physical Human-Robot Interaction

Ciro Natale

Dipartimento di Ingegneria, Università degli Studi della Campania Luigi Vanvitelli, Aversa, Italy

Abstract

Robots and humans can nowadays share their workspace safely. This possibility has been enabled by a combination of technologies leading to the so-called cobots, robots that

Introduction

Robots have been conceived since the dawn of robotics as machines capable of physical interaction with the environment. They are designed to respond to external stimuli, with different autonomy levels, just as a computer or a mobile phone does, but with a great difference: they are capable to physically change the world they are interacting with by exchanging mechanical energy. First industrial robots were designed to interact only with unanimated objects mainly to perform only repetitive tasks, and they were not allowed to interact with humans, because of severe safety requirements.

Nowadays, many robots are designed to perform their task not only on the shop floor of a manufacturing industry but also in scenarios where they have to share their workspace with humans, e.g., in service or personal robotics applications. This new generation of robots are called *cobots*, that stands for *collaborative robots*, since they are specifically designed to assist humans during task execution through direct physical interaction.

The cobot technology is a real breakthrough in robotics, and to achieve such result, engineers had to face a number of challenges from many points of view in the development process, from mechanical design and actuation to sensing and control architectures. The primary objective they had to pursue is the safety of such products during the physical interaction with human operators, both in case of intentional and unintentional contacts.

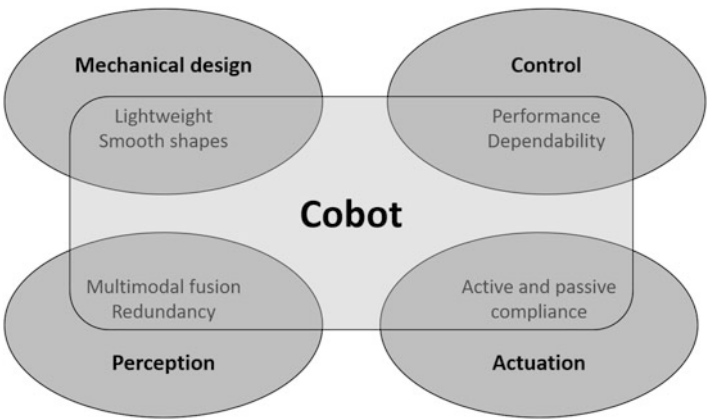
Figure 1 lists the main requirements that should be satisfied and the most relevant design objectives of the mechanical design, actuation and sensing system, as well as control architecture of a cobot. Most modern commercial cobots (some of them are depicted in Fig.2) adopt lightweight materials, and their links have smooth shapes without any sharp edge. The joint actuators have an intrinsic compliance or can be controlled in such a way to exhibit a programmable compliance. Usually, apart from the standard joint position sensors, these arms are endowed with redundant sensors and with joint torque sensors. Availability of direct

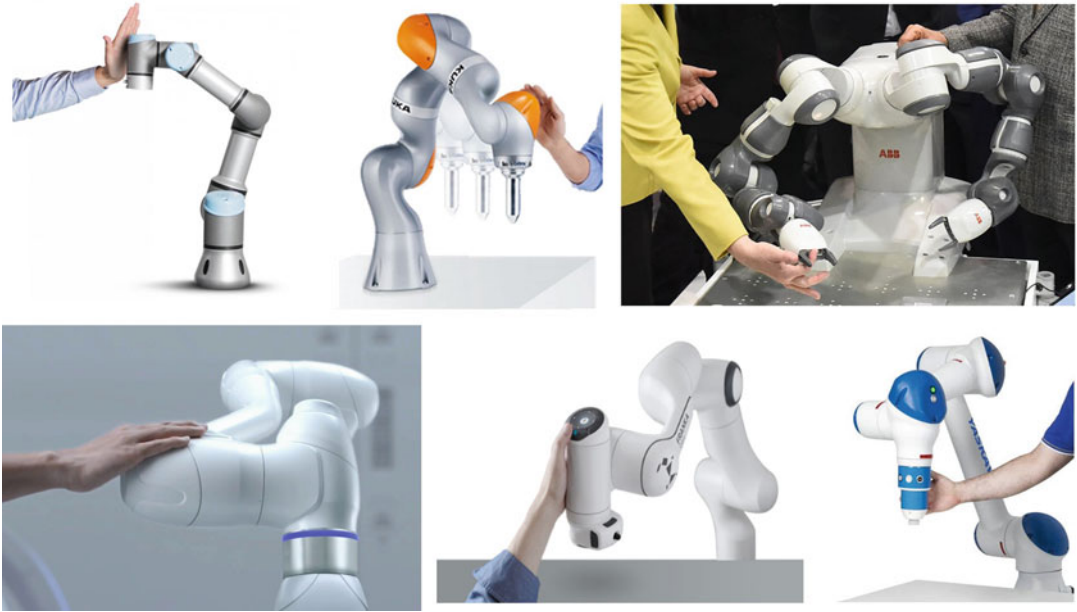
torque measurements to the control system enables advanced features such as programmable compliant motion (impedance control, force control) that, on one hand, increases the safety level during the interaction with human operators and, on the other hand, allows the robot to adapt to external uncertain contact constraints. All cobots give the user the possibility to program the robot by using hand guidance, usually through the aid of a smart human-machine interface (HMI) on board of the last link of the arm or even through haptic gestures that can be recognized based on external contact estimation. Reliable detection of contact with the environment and, most importantly, with the human operator is of paramount importance. The ability to distinguish between intentional contact and accidental collisions is also a distinguishing feature of some cobots. As better explained in the section dedicated to sensing systems, the detection can be based either on direct measurement of external forces or through model-based estimation algorithms.

Human-Robot Interaction Technology

Design of robots that are allowed to physically interact with humans in a safe manner is characterized by different approaches which involve design of actuation, sensing, and control subsystems of the robot.

Physical Human-Robot Interaction, Fig. 1 Cobot technology characteristics





Physical Human-Robot Interaction, Fig. 2 Some commercial cobots (from left to right): UR3, LBR iiwa, YuMi, M1013, Panda, and HC10. (Courtesy of Universal Robots,

Kuka Robotics, ABB Robotics, Doosan Robotics, Franka Emika and Yaskawa, respectively)

Actuation System

To satisfy the design requirements of a cobot discussed in Fig. 1, e.g., the lightweight design and compliant behavior, distinct approaches are generally adopted for the actuation system, i.e., the *mechatronic approach*, the *tendon-driven approach*, and the *intrinsically flexible approach*.

In the mechatronic approach, power and control electronics is integrated into the joints to achieve high modularity so as to obtain different kinematic designs while maintaining joint characteristics. Joint modules are usually based on high-torque motors with high transmission ratios in combination with joint torque sensors to get accurate joint torque tracking capabilities, besides classical joint position sensing. Compliance is then ensured by proper control algorithms (see “[Control System](#)” section).

Dually, tendon-driven robots are typically actuated by remote actuators (usually placed in the robot base) with the aim to reduce the moving inertia and in turn to limit the mechanical energy of any possible collision during the robot motion. Low reduction ratios are usually adopted in the

tendon mechanisms so as to obtain backdrivable systems.

Another method to achieve high dynamic performances, still ensuring safety (including that of the robot) in case of collisions, consists in the adoption of intrinsically compliant mechanisms in the robot joints. This is the case of the well-known series elastic actuator (SEA) technology mostly adopted to actuate robot legs or powered orthoses. The main SEA characteristics are their intrinsic resistance to impacts and vibrations and the low output mechanical impedance, which makes them backdrivable. However, their impedance is fixed, while it might be useful, depending on the application, to have actuators with adjustable impedance or at least compliance. Variable Stiffness Actuators (VSA) and Variable Impedance Actuators (VIA) were originally conceived to make robots safer during unforeseen collisions by dynamic decoupling of the motor inertia from the link-side inertia, but then they have been adopted to improve robot performance as well depending on the type of task being carried out, e.g., pure positioning (high stiffness) or physical contact (low stiffness). An additional

advantage of actuators including elastic elements is the possibility to store mechanical energy that can be suitably output to the load to improve dynamic performance.

Few examples of all discussed joint actuation technologies are reported in Fig. 3.

Sensing System

The perception system of cobots is usually more rich than the one of a standard robot, for two main reasons. Both implementation of a compliant behavior and detection and reaction to collisions with the environment require sensing systems able to get contact information of the robot with external agents. In standard industrial robots, compliant motion is usually achieved by resorting to wrist force/torque sensors, which are able to measure the contact wrench applied by the end effector to the environment. However, physical interaction of the robot with a human operator, both deliberate and accidental, can take place on every part of the robot body. To this aim, joint torque sensors can be adopted to estimate external forces applied to any link of the robot body. Torque sensing is usually done via strain gauge-based devices and their measurement is used also to implement low-level torque control with a high degree of accuracy, mainly to reject torque disturbances such as friction in gear boxes. Of course joint torque measurements include not only contribution of the external contact forces but also dynamic effects such as gravity, which need to be compensated for. A number of algorithms have been proposed in the literature to solve this problem. The most common approach is the so-called angular momentum-based observer. The most commonly adopted dynamic model of a robot manipulator with flexible joints is the so-called Spong's model, i.e.,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{ext}} \quad (1)$$

$$\mathbf{B}\ddot{\boldsymbol{\theta}} + \boldsymbol{\tau} = \boldsymbol{\tau}_m - \boldsymbol{\tau}_F \quad (2)$$

$$\boldsymbol{\tau} = \mathbf{K}_j(\boldsymbol{\theta} - \mathbf{q}) \quad (3)$$

where $\mathbf{q}$ and $\boldsymbol{\theta}$ are the joint position vectors at link and motor sides, respectively, $\mathbf{M}(\mathbf{q})$ is the inertia matrix of the rigid body dynamics, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the

matrix of Coriolis and centrifugal terms, $\mathbf{g}(\mathbf{q})$ is the vector of gravitational torques, $\boldsymbol{\tau}$ is the vector of joint torques, $\boldsymbol{\tau}_{\text{ext}}$ is the vector of external torques (those corresponding to external contact forces applied to the links), $\mathbf{B}$ is the inertia matrix of the joint motors, $\boldsymbol{\tau}_m$ is the vector of motor driving torques, $\boldsymbol{\tau}_F$ is the vector of friction torques on the motor side, and, finally, $\mathbf{K}_j$ is the joint stiffness matrix.

The model-based observer is based on the equation governing the estimate $\hat{\mathbf{p}}$ of the angular momentum vector $\mathbf{p} = \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}$, namely

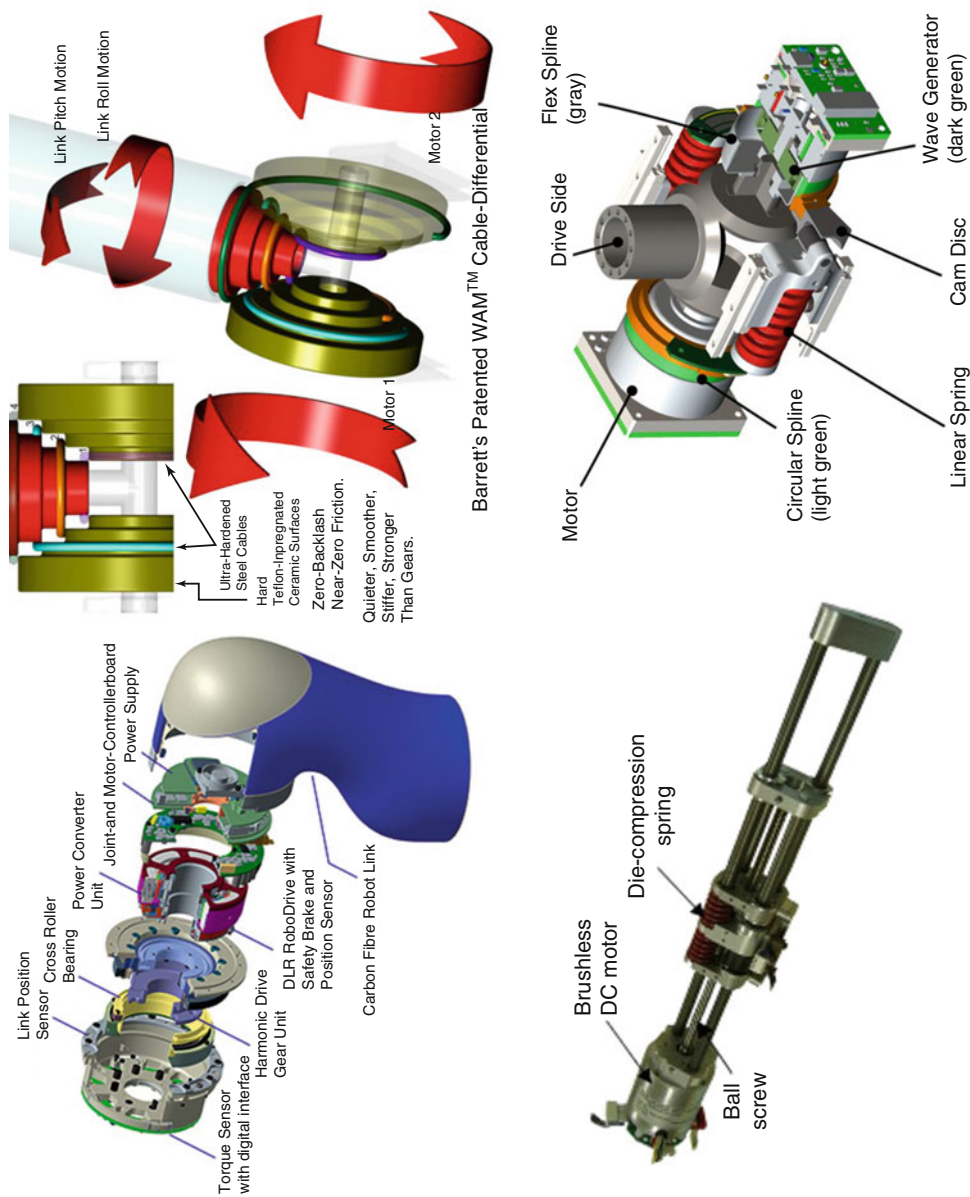
$$\dot{\hat{\mathbf{p}}} = -k_O(\hat{\mathbf{p}} - \mathbf{p}) + \boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) - \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}}. \quad (4)$$

It is straightforward to verify that the residual $\mathbf{r} = \hat{\mathbf{p}} - \mathbf{p}$ asymptotically converges to the external torque $\boldsymbol{\tau}_{\text{ext}}$ with a time constant proportional to $1/k_O$. Such observer requires full-state feedback as well as joint torque measurements, and it is more suitable to be adopted when the arm is used under impedance or compliance control. Whereas, in case the arm is position controlled, the joint position can be considered very close to the reference $\mathbf{q}_d$, and thus a computationally less expensive algorithm can be used to estimate external joint torques, i.e.,

$$\boldsymbol{\tau}_{\text{ext}} \simeq \boldsymbol{\tau} - \left(\mathbf{M}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}_d, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}_d) \right). \quad (5)$$

Note how this method is affected only by the noise on the measured joint torque, while algorithm (4) requires differentiation of joint position measurements.

Both the methods discussed above require accurate knowledge of the dynamic model of the robot; moreover, they can be used to detect the components of the contact forces applied to the links and the corresponding contact points only in special cases. On the other hand, direct measurement of contact location and external forces applied to the robot body can be achieved by proper distributed sensing devices, like artificial tactile skins. Most existing tactile skins are able to detect only the contact point and the normal pressure on it, while only few prototypes are able to perform measurement of the complete



Physical Human-Robot Interaction, Fig. 3 (continued)

contact force vector in multiple points. The most adopted technologies are piezoresistive, capacitive, and optoelectronic ones. Some prototypes are reported in Fig. 4 together with joint torque sensors, wrist force sensors, as well as depth cameras used to monitor robot workspace.

In fact, great relevance is taken on by visual perception systems adopted to monitor the robot workspace with the aim to ensure safety of humans that share their workspace with the robot. A real breakthrough in robotics happened at the arrival on the market of low-cost depth cameras, while high-end stereo vision has become widespread only after the adoption of computationally powerful devices such as GPUs (Graphical Processing Units). These perception systems are nowadays capable of accurate tracking of human bodies even in presence of partial occlusions.

Control System

Physical interaction of a robot with the environment, including humans, is certainly more safe when the robot is controlled with a suitable compliance control algorithm. In this way the arm configuration tries to adapt to contact forces rather than following the programmed path as close as possible despite any contact. During the decades, basic compliance control schemes of the early 1980s evolved to more sophisticated algorithms that not only provide the robot with suitable programmable compliant behaviors but also optimize robot configuration to minimize the *effective mass* at the contact point, i.e., the equivalent inertia in the direction of the contact, by resorting to multitask with priority compliance control algorithms. Of course, multiple objectives can be achieved only if the robot has enough degrees of freedom to exploit.

Collision Reaction

During operation of a cobot, an algorithm to detect collisions is always active, and it usefully exploits the external torque estimated as described in the previous section. By comparing the components of the residual $\mathbf{r}$ with suitably selected thresholds (e.g., 10% of the maximum torque of each joint), a collision reaction algorithm is triggered. Different strategies can be adopted, i.e.:

1. the robot is *stopped* at the position reached at the time of collision;
2. the robot control is switched to *gravity compensation mode* so as to obtain a very compliant behavior, i.e., $\boldsymbol{\tau} = \mathbf{g}(\mathbf{q})$;
3. the robot control is switched to the so-called torque reflex mode, i.e., $\boldsymbol{\tau} = \mathbf{g}(\mathbf{q}) + (\mathbf{I} - \mathbf{K}_r)\boldsymbol{\tau}_{\text{ext}}$, such that the robot inertia is scaled by $\mathbf{K}_r^{-1}$;
4. the robot control is switched to *admittance control mode* so as to move the contact point in the direction of the estimated contact force vector but away from the collision area.

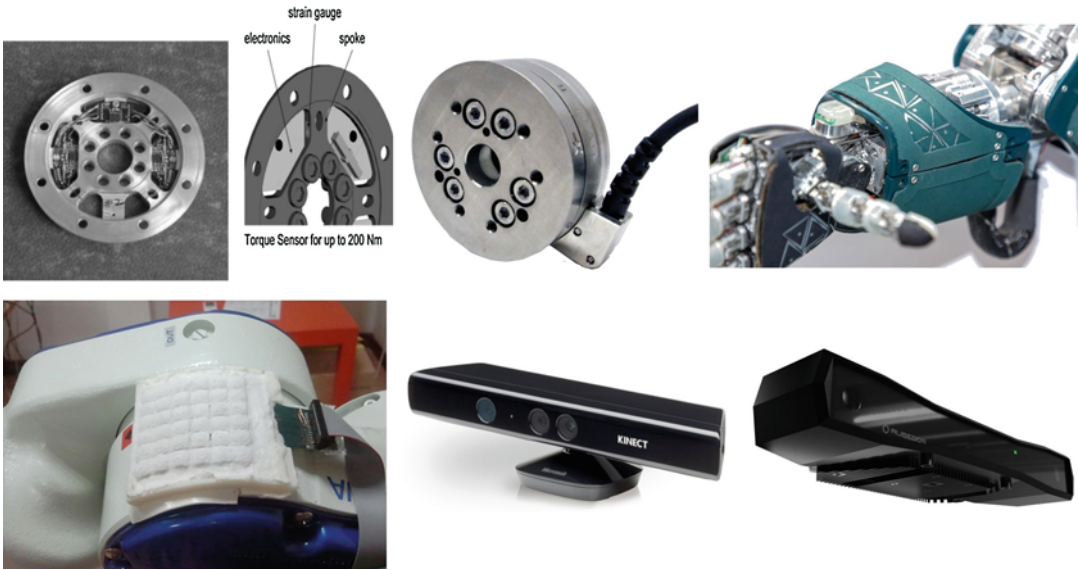
In the last case, availability of direct contact force measurements is very useful and safe to ensure a behavior of the robot not influenced by modeling errors but only by sensor noise.

Manual Guidance

One of the features that distinguishes a cobot from a standard industrial robot is the possibility to program the motion of the machine by using the so-called manual guidance or walk-through programming. The basic approach is to control the robot in gravity compensation mode so that it is fully compliant along all directions and the user can manually adjust any link in the desired configuration. Another approach is to use admittance control by measuring the contact force and translating such force into a velocity to move

Physical Human-Robot Interaction, Fig. 3 Joint actuation systems (from left to right): joint mechatronics of the LWR III, cable-differential transmission of the WAM arm, SEA23-23 actuator, and floating spring joint of the

David. (Courtesy of DLR, Barrett Technologies, Yobotics Inc. (with permission from Elsevier Garcia et al. 2011), and DLR, respectively)



Physical Human-Robot Interaction, Fig. 4 Perception systems for pHRI (from left to right): joint torque sensor of the LWR III, wrist f/t sensor, tactile skin of iCub, SUN

tactile skin, Kinect v1 RGB-D camera, and Viper stereo camera. (Courtesy of DLR, ATI, IIT, Univ. of Campania, Microsoft and Rubedos, respectively)

the arm, e.g., through tactile skins, which can be used also as HMI. If the arm is redundant, redundancy can be exploited in combination with online adaptation of admittance parameters with the aim to enhance the comfort perceived by humans during manual guidance.

Summary and Future Directions

Even though the cobot technology originally imagined by researcher in the late 1990s can now be considered a reality, still a lot of research is being done toward even more powerful and safe robots.

An interesting area of research concerns the combination of compliance control strategies with machine learning methods with the aim to automatically find the right parametrization of compliance controllers for service robotics applications (Leidner et al. 2016) or cognitive reasoning for compliant manipulation (Leidner 2019).

Another research trend is the one toward the investigation of methods to estimate human intention in collaborative workspaces

through human motion prediction (Mainprice and Berenson 2013) as well as human action prediction (Hawkins et al. 2013), e.g., in assembly collaborative tasks in the automotive industry (Kinugawa et al. 2017).

Thinking of human-robot teams carrying out common collaborative tasks is nowadays no longer a dream; recent studies on mutual adaptation explore methods to allow robots to adapt their behavior with the aim to improve the performance of the human-robot team (Nikolaidis et al. 2017).

Cross-References

- [Force Control in Robotics](#)
- [Trust Models for Human-Robot Interaction and Control](#)

Recommended Reading

Many surveys can be found in the literature summarizing the main technical achievements that led to the cobot technology (Haddadin and

Croft 2016; Villani et al. 2018; Bauer et al. 2008; Goodrich and Schultz 2007). The first patent citing the term cobot is Colgate and Peshkin (1999). One of the first road maps for the cobot technology development is De Santis et al. (2008), which introduced suggestions to draft metrics for evaluating safety and dependability in pHRI. Those suggestions became the scientific basis of the technical specification ISO/TS 15066 (ISO/TC 299 Robotics 2016), still under review, to be included in the standard ISO 10128 (ISO/TC 299 Robotics 2011).

Literature on actuators suitably designed for pHRI is vast, the integrated joint technology of DLR (Hirzinger et al. 2002) was the basis of more than a commercial cobot, and SEA were first introduced by Pratt et al. (1997) and were later combined with magneto-rheological fluids for controlling the damping of a locomotion system in Garcia et al. (2011). VSA of the David robot was presented in Wolf et al. (2011), while the cable-driven actuation system of the WAM arm appeared in Gordon and Townsend (1989).

Force/tactile sensing systems are nowadays among the hot topics in many robotics conferences as they are the primary source of information on the physical interaction of a robot with its environment. Joint torque sensors of DLR (Hirzinger et al. 2003) are the basis of the mechatronics approach to perform torque control of modern cobots. A number of tactile skins have been developed throughout the years, and the one equipping the iCub in Fig. 4 was first presented in Cannata et al. (2008), while the only artificial skin sensitive to both normal and shear loads is the one presented in Cirillo et al. (2014) and later used in Cirillo et al. (2016) for admittance control and in Cirillo et al. (2017) for manual guidance and intuitive programming via haptic gestures. The manual guidance based on adaptive impedance control was instead proposed in Ficuciello et al. (2015).

Control design of robots with dynamic characteristics typical of a cobot is mostly based on the famous dynamic model of robots with elastic joints proposed by Spong (1987). Such model is at the base of compliance and impedance control algorithms

of lightweight robots (Albu-Schäffer et al. 2007). The same research group has also contributed to the study of hierarchical multitask control algorithms for service robots (Dietrich et al. 2017, 2018), mainly based on the dynamically consistent approach of Featherstone and Khatib (1997) and Sentis and Khatib (2005).

The model-based estimation algorithms for collision detection and reaction based on the generalized momentum date back to De Luca et al. (2006) and Haddadin et al. (2008), but the reader interested in examining in depth the topic can start from the recent survey (Haddadin et al. 2017).

Bibliography

- Albu-Schäffer A, Ott C, Hirzinger G (2007) A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *Int J Robot Res* 26(1):23–39
- Bauer A, Wollherr D, Buss M (2008) Human-robot collaboration: a survey. *Int J Humanoid Robot* 5(1):47–66
- Cannata G, Maggiali M, Metta G, Sandini G (2008) An embedded artificial skin for humanoid robots. In: *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, pp 434–438
- Cirillo A, Cirillo P, De Maria G, Natale C, Pirozzi S (2014) An artificial skin based on optoelectronic technology. *Sensors Actuators A Phys* 212:110–122
- Cirillo A, Ficuciello F, Natale C, Pirozzi S, Villani L (2016) A conformable force/tactile skin for physical human-robot interaction. *IEEE Robot Autom Lett* 1(1):41–48
- Cirillo A, Cirillo P, De Maria G, Natale C, Pirozzi S (2017) A distributed tactile sensor for intuitive human-robot interfacing. *J Sens* 2017:14
- Colgate J, Peshkin M (1999) Cobots. U.S. Patent 5,952,796
- De Luca A, Albu-Schäffer A, Haddadin S, Hirzinger G (2006) Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 1623–1630
- De Santis A, Siciliano B, De Luca A, Bicchi A (2008) An atlas of physical human-robot interaction. *Mech Mach Theory* 43(3):253–270
- Dietrich A, Wu X, Bussmann K, Ott C, Albu-Schäffer A, Stramigioli S (2017) Passive hierarchical impedance control via energy tanks. *IEEE Robot Autom Lett* 2(2):522–529

- Dietrich A, Ott C, Park J (2018) The hierarchical operational space formulation: stability analysis for the regulation case. *IEEE Robot Autom Lett* 3(2):1120–1127
- Featherstone R, Khatib O (1997) Load independence of the dynamically consistent inverse of the jacobian matrix. *Int J Robot Res* 16(2):168–170
- Ficuciello F, Villani L, Siciliano B (2015) Variable impedance control of redundant manipulators for intuitive human-robot physical interaction. *IEEE Trans Robot* 31(4):850–863
- Garcia E, Arevalo J, Muñoz G, de Santos PG (2011) Combining series elastic actuation and magneto-rheological damping for the control of agile locomotion. *Robot Auton Syst* 59(10):827–839
- Goodrich MA, Schultz AC (2007) Human-robot interaction: a survey. now Publisher Inc., Hanover, MA
- Gordon SJ, Townsend WT (1989) Integration of tactile force and joint torque information in a whole-arm manipulator. In: *Proceedings, 1989 International Conference on Robotics and Automation*, vol 1, pp 464–469
- Haddadin S, Croft E (2016) *Physical human-robot interaction*, 2nd edn. Springer, Berlin Heidelberg, pp 1835–1874
- Haddadin S, Albu-Schäffer A, De Luca A, Hirzinger G (2008) Collision detection and reaction: a contribution to safe physical human-robot interaction. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 3356–3363
- Haddadin S, De Luca A, Albu-Schäffer A (2017) Robot collisions: a survey on detection, isolation, and identification. *IEEE Trans Robot* 33(6):1292–1312
- Hawkins KP, Nam Vo, Bansal S, Bobick AF (2013) Probabilistic human action prediction and wait-sensitive planning for responsive human-robot collaboration. In: *2013 13th IEEE-RAS International Conference on Humanoid Robots*, pp 499–506
- Hirzinger G, Sporer N, Albu-Schäffer A, Hahnle M, Krenn R, Pascucci A, Schedl M (2002) Dlr's torque-controlled light weight robot iii: are we reaching the technological limits now? In: *Proceedings 2002 IEEE International Conference on Robotics and Automation* (Cat. No.02CH37292), vol 2, pp 1710–1716
- Hirzinger G, Brunner B, Landzettel K, Sporer N, Butterfaß J, Schedl M (2003) Space robotics—dlr's telerobotic concepts, lightweight arms and articulated hands. *Auton Robot* 14(2):127–145
- ISO/TC 299 Robotics (2011) ISO 10218-1:2011 robots and robotic devices – safety requirements for industrial robots – part 1: robots. International Organization for Standardization
- ISO/TC 299 Robotics (2016) ISO/TS 15066:2016 – robots and robotic devices – collaborative robots. International Organization for Standardization
- Kinugawa J, Kanazawa A, Arai S, Kosuge K (2017) Adaptive task scheduling for an assembly task coworker robot based on incremental learning of human's motion patterns. *IEEE Robot Autom Lett* 2(2):856–863
- Leidner DS (2019) *Cognitive reasoning for compliant robot manipulation*. Springer, Cham
- Leidner DS, Dietrich A, Beetz M, Albu-Schäffer A (2016) Knowledge-enabled parameterization of whole-body control strategies for compliant service robots. *Auton Robot* 40(3):519–536
- Mainprice J, Berenson D (2013) Human-robot collaborative manipulation planning using early prediction of human motion. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 299–306
- Nikolaidis S, Hsu D, Srinivasa S (2017) Human-robot mutual adaptation in collaborative tasks: models and experiments. *Int J Robot Res* 36(5–7):618–634
- Pratt G, Williamson M, Dillworth P, Pratt J, Wright A (1997) Stiffness isn't everything. *Lecture notes in control and information sciences*, vol 223. Springer, London/Berlin, pp 253–262
- Sentis L, Khatib O (2005) Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *Int J Humanoid Robot* 02(04):505–518
- Spong MW (1987) Modeling and control of elastic joint robots. *J Dyn Syst Meas Control* 109(4):310–318
- Villani V, Pini F, Leali F, Secchi C (2018) Survey on human-robot collaboration in industrial settings: safety, intuitive interfaces and applications. *Mechatronics* 55:248–266
- Wolf S, Eiberger O, Hirzinger G (2011) The dlr fsj: energy based design of a variable stiffness joint. In: *2011 IEEE International Conference on Robotics and Automation*, pp 5082–5089

PID Control

Sebastian Dormido¹ and Antonio Visioli²

¹Departamento de Informatica y Automatica, UNED, Madrid, Spain

²Dipartimento di Ingegneria Meccanica e Industriale, University of Brescia, Brescia, Italy

Abstract

Since their introduction in industry a century ago, proportional–integral–derivative (PID) controllers have become the de facto standard for the process industry. In this entry, fundamentals of PID control are outlined, starting from the basic control law. Additional functionalities and the tuning and automatic tuning of the parameters are then considered.

Keywords

Anti-windup · Autotuning · Controller tuning · Derivative action · PI control · Proportional control · Proportional-integral-derivative control · Ziegler-Nichols

Synonyms

Proportional-Integral-Derivative Control

Introduction

A proportional–integral–derivative (PID) controller is a three-term controller that has a long history in the automatic control field, starting from the beginning of the last century. Owing to its intuitiveness and relative simplicity, in addition to the satisfactory performance that it is able to provide with a wide range of processes, it has become the de facto standard controller in industry. It has been evolving along with the progress of technology, and nowadays it is very often implemented in digital form rather than with pneumatic or electrical components. It can be found in virtually all kinds of control equipments, either as a stand-alone (single-station) controller or as a functional block in Programmable Logic Controllers (PLCs) and Distributed Control Systems (DCSs). Actually, the new potentialities offered by the development of the digital technology and of the software packages have led to a significant growth of the research in the PID control field: new effective tools have been devised for the improvement of the analysis and design methods of the basic algorithm as well as for the improvement of the additional functionalities that are implemented with the basic algorithm in order to increase its performance and its ease of use.

The success of the PID controllers is also enhanced by the fact that they often represent the fundamental component for more sophisticated control schemes that can be implemented when the basic control law is not sufficient to achieve the required performance or when a more complicated control task is of concern.

Basics

Using a PID controller means applying a feedback controller that consists of the sum of three types of control actions: a proportional action, an integral action, and a derivative action.

The proportional control action is proportional to the current control error, according to the expression

$$u(t) = K_p e(t) = K_p (r(t) - y(t)), \quad (1)$$

where u is the controller output, K_p is the proportional gain, r is the reference signal, and y is the process output. Its meaning is straightforward, since it implements the typical operation of increasing the control variable when the control error is large (with appropriate sign). The transfer function of a proportional controller can be trivially derived as

$$C(s) = K_p. \quad (2)$$

The main drawback of using a pure proportional controller is that, in general, it cannot set to zero the steady-state error. This motivates the addition of a bias (or reset) term u_b , namely,

$$u(t) = K_p e(t) + u_b. \quad (3)$$

The value of u_b can then be adjusted manually until the steady-state error is reduced to zero.

In commercial products, the proportional gain is often replaced by the proportional band PB , which is the range of error that causes a full-range change of the control variable, i.e.,

$$PB = \frac{100}{K_p}. \quad (4)$$

The integral action is proportional to the integral of the control error, i.e.,

$$u(t) = K_i \int_0^t e(\tau) d\tau, \quad (5)$$

where K_i is the integral gain. It appears that the integral action is related to the past values of the control error. The corresponding transfer

function is

$$C(s) = \frac{K_i}{s}. \quad (6)$$

The presence of an integral action allows to reduce the steady-state error to zero when a step reference signal is applied or a step load disturbance occurs. In other words, the integral action is able to set automatically the correct value of u_b in (3) so that the steady-state error is zero. For this reason, the integral action is also often called *automatic reset*.

While the proportional action is based on the current value of the control error and the integral action is based on the past values of the control error, the derivative action is based on the predicted future values of the control error. An ideal derivative control law can be expressed as

$$u(t) = K_d \frac{de(t)}{dt}, \quad (7)$$

where K_d is the derivative gain. The corresponding controller transfer function is

$$C(s) = K_d s. \quad (8)$$

The meaning of the derivative action can be better understood by considering the first two terms of the Taylor series expansion of the control error at time T_d ahead:

$$e(t + T_d) \simeq e(t) + T_d \frac{de(t)}{dt}. \quad (9)$$

If a control law proportional to this expression is considered, i.e.,

$$u(t) = K_p \left(e(t) + T_d \frac{de(t)}{dt} \right), \quad (10)$$

this naturally results in a PD controller. The control variable at time t is therefore based on the predicted value of the control error at time $t + T_d$. For this reason, the derivative action is also called *anticipatory control*, or *rate action*, or *pre-act*.

The combination of the proportional, integral, and derivative actions can be done in different ways. In the so-called *ideal* or *non-interacting* form, the PID controller is described by the

following transfer function:

$$C_i(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right), \quad (11)$$

where K_p is the proportional gain, T_i is the integral time constant, and T_d is the derivative time constant. An alternative representation is the *series* or *interacting* form:

$$\begin{aligned} C_s(s) &= K'_p \left(1 + \frac{1}{T'_i s} \right) (T'_d s + 1) \\ &= K'_p \left(\frac{T'_i s + 1}{T'_i s} \right) (T'_d s + 1), \end{aligned} \quad (12)$$

where the fact that a modification of the value of the derivative time constant T'_d affects also the proportional action justifies the nomenclature adopted. Suitable conversion formulae can be applied to obtain an ideal PID controller equivalent to a series one. Obtaining an equivalent PID controller in series form starting from an ideal one is possible only if the zeros of the ideal PID controller are real.

Additional Functionalities

The expression (11) or (12) of a PID controller is actually not employed in practical cases because of a few problems that can be solved with suitable modifications of the basic control law.

Modifications of the Derivative Action

From Expressions (11) and (12), it appears that the controller transfer function is not proper, because of the derivative action, and therefore, it cannot be implemented in practice. Indeed, the high-frequency gain of the pure derivative action is responsible for the amplification of the measurement noise in the manipulated variable. This problem can be solved by filtering the derivative action with (at least) a first-order low-pass filter. The filter time constant should be selected in order to suitably filter the noise and to avoid a significant influence on the dominant dynamics of the PID controller. Thus, it can be selected

as T_d/N , where N generally assumes a value between 1 and 33, although in the majority of the practical cases its setting falls between 8 and 16. Alternatively, the overall control variable can be filtered.

Another issue related to the derivative action that has to be considered is the so-called derivative kick. In fact, when an abrupt (stepwise) change of the set-point signal occurs, the derivative action is very large, and this results in a spike in the control variable signal, which is undesirable. This problem could be simply avoided by applying the derivative term to the process output only instead of the control error. In this case, the ideal (not filtered) derivative action becomes

$$u(t) = -K_p T_d \frac{dy(t)}{dt}. \quad (13)$$

Obviously, when the set-point signal is constant, applying the derivative term to the control error or to the process variable is equivalent. Thus, the load disturbance rejection performance is the same in both cases.

Set-Point Weighting for Proportional Action

A typical problem with the design of a feedback controller is to achieve a high performance both in the set-point following task and in the load disturbance rejection task at the same time. For example, for stable processes, a fast load disturbance rejection is achieved with a high-gain (aggressive) controller, which gives an oscillatory set-point step response on the other side. This problem can be approached by using a two-degree-of-freedom control architecture, where a feedback controller is designed to achieve a high bandwidth and therefore a satisfactory load disturbance rejection performance, and then the set-point signal is filtered before applying it to the closed-loop system.

In the context of PID control, this can be achieved by weighting the set-point signal for the proportional action, that is, to define the proportional action as follows:

$$u(t) = K_p(\beta r(t) - y(t)), \quad (14)$$

where the value of β is between 0 and 1.

In this way, the control scheme has a feedback controller (11), and the set-point signal is filtered by the system

$$F(s) = \frac{1 + \beta T_i s + T_i T_d s^2}{1 + T_i s + T_i T_d s^2}. \quad (15)$$

The load disturbance rejection task is decoupled from the set-point following task, and obviously it does not depend on the weight β , which can be employed to smooth the (step) set-point signal in order to damp the response to a set-point change. The smaller the value of β , the smaller the overshoot and the higher the rise time.

Anti-windup

One of the most well-known possible sources of performance degradation is the so-called integrator windup phenomenon, which occurs when the controller output saturates (typically when a large set-point change occurs). In this case, the system operates as in the open-loop case, since the actuator is at its maximum (or minimum) limit, regardless of the process output value. The control error decreases more slowly than in the ideal case (where there are no saturation limits), and therefore, the integral term becomes large (it *winds up*). Thus, even when the value of the process variable attains that of the reference signal, the controller still saturates due to the integral term, and this generally yields large overshoots and settling times.

In order to cope with this problem, an additional functionality designed for this purpose can be conveniently used. This can be done in different ways. For example, in the conditional integration approach, the integration is stopped when the control variable saturates and the control error and the control variable have the same sign. Alternatively, in the back-calculation approach, the integral term is recomputed when the controller saturates by feeding back the difference of the saturated and unsaturated control signal.

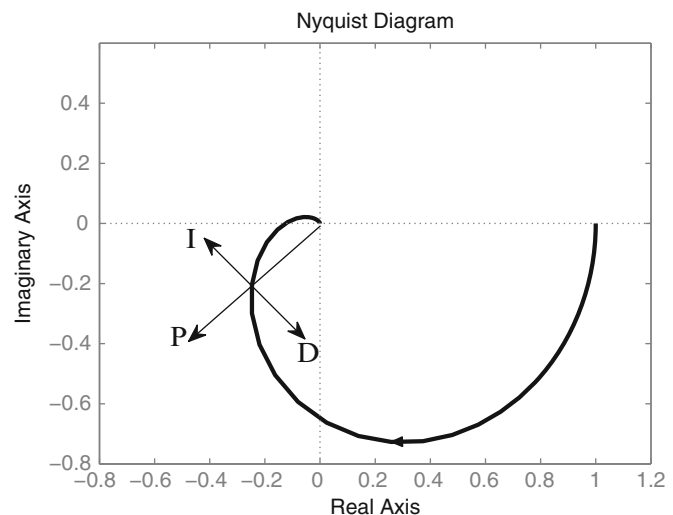
Tuning

The selection of the PID parameters, i.e., the tuning of the PID controller, is obviously the crucial issue in the overall controller design. This operation should be performed in accordance with the control specifications (which should take into account the set-point following, the load disturbance rejection, the control effort, and the robustness of the system). A major advantage of the PID controller is that its parameters have a clear physical meaning, and therefore, manual tuning is relatively simple. For example, for stable processes, increasing the proportional gain leads, in general, to a faster but more oscillatory response. In fact, by increasing K_p for the same value of the control error, the proportional control action increases, and so does the aggressiveness of the controller. Then, increasing the integral time constant (i.e., decreasing the effect of the integral action) results, in general, in a slower response but in a more damped system. This is because a larger value of T_i implies a smaller value of the control action at a given time instant of the transient response (assuming the same values of the past control errors). Finally, increasing the derivative time constant gives a damping effect. Indeed, if the set point is constant, the derivative action is proportional to the derivative of the process variable with a negative sign, and therefore, the

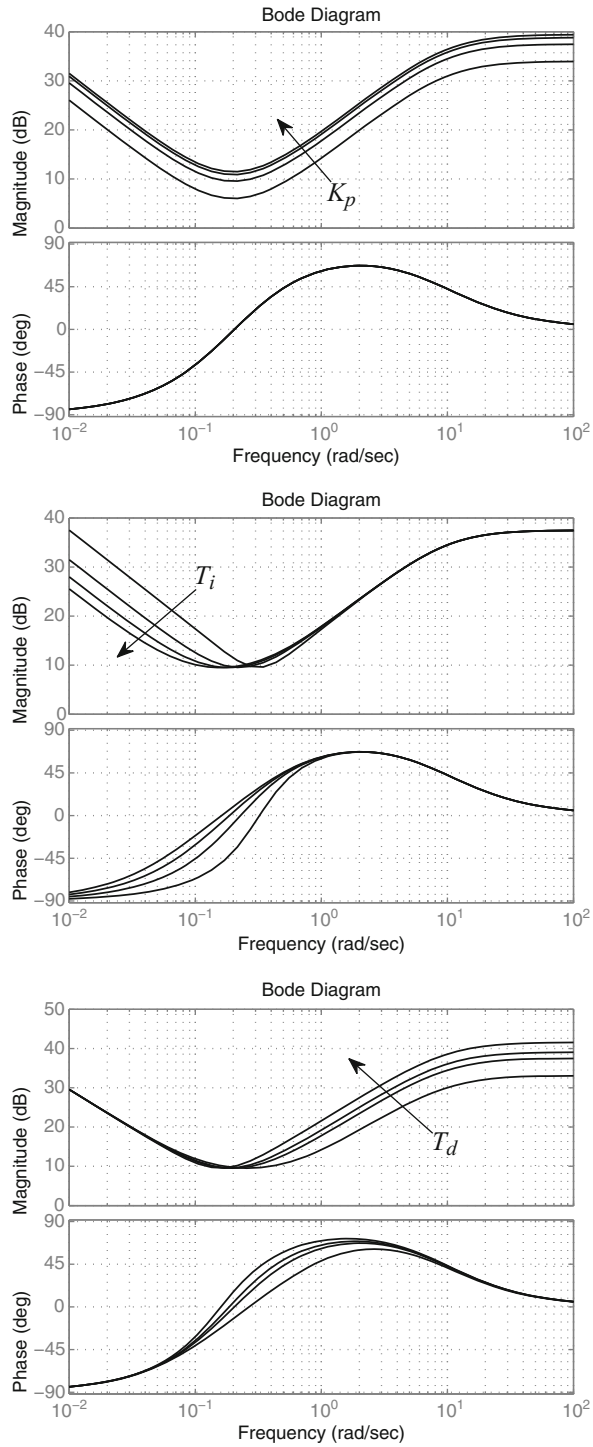
derivative action increases (with a negative sign) when the slope of the transient response increases (so that a big overshoot is avoided). However, in this context, much care should be taken to avoid increasing the derivative time constant too much as an opposite effect might occur in this case and an unstable system could eventually result. This is because the prediction over a too long time interval might be wrong.

The above considerations can be understood by considering a transient response in the time domain but also by considering the frequency response of the system and how it changes by modifying the PID parameters. For example, the effect of increasing the three controller actions can be seen as translating any point of the Nyquist plot in each of the directions shown in Fig. 1. From another point of view, analogous considerations can be done by considering the Bode plot. For example, considering a PID controller in ideal form with a filter on the overall control action, the effect of modifying the three parameters in the controller Bode plot is shown in Fig. 2. The effects of the parameter modification in the achieved performance can be better ascertained by plotting the frequency response of the loop transfer function for different cases. As an example, consider the process $P(s) = 1/(10s + 1)e^{-4s}$ and the PID controller $C(s) = K_p(1 + \frac{1}{T_i s} + T_d s)\frac{1}{T_f s + 1}$

PID Control, Fig. 1 Effect of an increment of the three PID control actions on a point of the Nyquist plot



PID Control, Fig. 2 Effect of an increment of three PID parameters on the controller Bode plot. The modifications are considered separately, namely, two parameters are fixed, while the other is modified



where the parameters are in the following ranges: $K_p \in [2, 15/4]$, $T_i \in [4, 16]$, $T_d \in [1.5, 4]$, being always $T_f = 0.1$. From Fig. 3, it appears that an increment of K_p yields an increment of the bandwidth and a decrement of the phase margin. Conversely, an increment of T_i has an opposite effect. Finally, it can be seen that the increment of T_d initially yields to an increment of the bandwidth and of the phase margin at the same time, but then a sudden increment of the bandwidth occurs, and this corresponds to a sudden decrement of the phase margin (because of the dead time of the process) with a possible loss of stability.

In any case, in order to ease the procedure, a large number of tuning rules have been proposed in the last century, starting from the well-known Ziegler–Nichols ones (Nichols and Ziegler 1942). Different approaches in this context are analyzed hereafter.

Empirical Tuning

Empirical tuning methods (like the Ziegler–Nichols and its refinements, Cohen–Coon, or Chien–Hrones–Reswick ones (O'Dwyer 2006)) consist in selecting the parameters of the PID controllers by using some empirical formulae which give the PID gains based on parameters of the process. Usually, the process parameters are those of a first-order-plus-dead-time (FOPDT) model or the ultimate gain and frequency of the process itself (the ultimate gain is the largest value of a proportional-only control that produces a sustained oscillation of the process variable, that is, that results in a marginally stable closed-loop system, while the ultimate frequency is the frequency of the corresponding sustained oscillation). These parameters can be obtained by means of a simple open-loop (step response) or closed-loop (relay feedback) experiment.

Model-Based Tuning

In model-based tuning (like the Dahlin's and Haalman's methods and the Internal Model Control one (O'Dwyer 2006)), the PID control law is determined analytically starting from a process model and by selecting an appropriate (closed-loop) target transfer function. The user

generally selects the desired closed-loop time constant as a tuning parameter which allows the handling of the trade-off between aggressiveness and robustness (and control effort). In this context, according to the well-known SIMC (Simplified Internal Model Control) tuning rules (Skogestad 2003), the closed-loop time constant should be selected equal to the dead time of the process.

Optimal Tuning

Optimal tuning rules aim at minimizing a given objective function. Usually, an integral function of the control error is selected for this purpose, for example,

$$J = \int_0^\infty t^n e^2(t) dt \quad (16)$$

where $n = 0, 1, 2$ or the Integrated Absolute Error

$$IAE = \int_0^\infty |e(t)| dt. \quad (17)$$

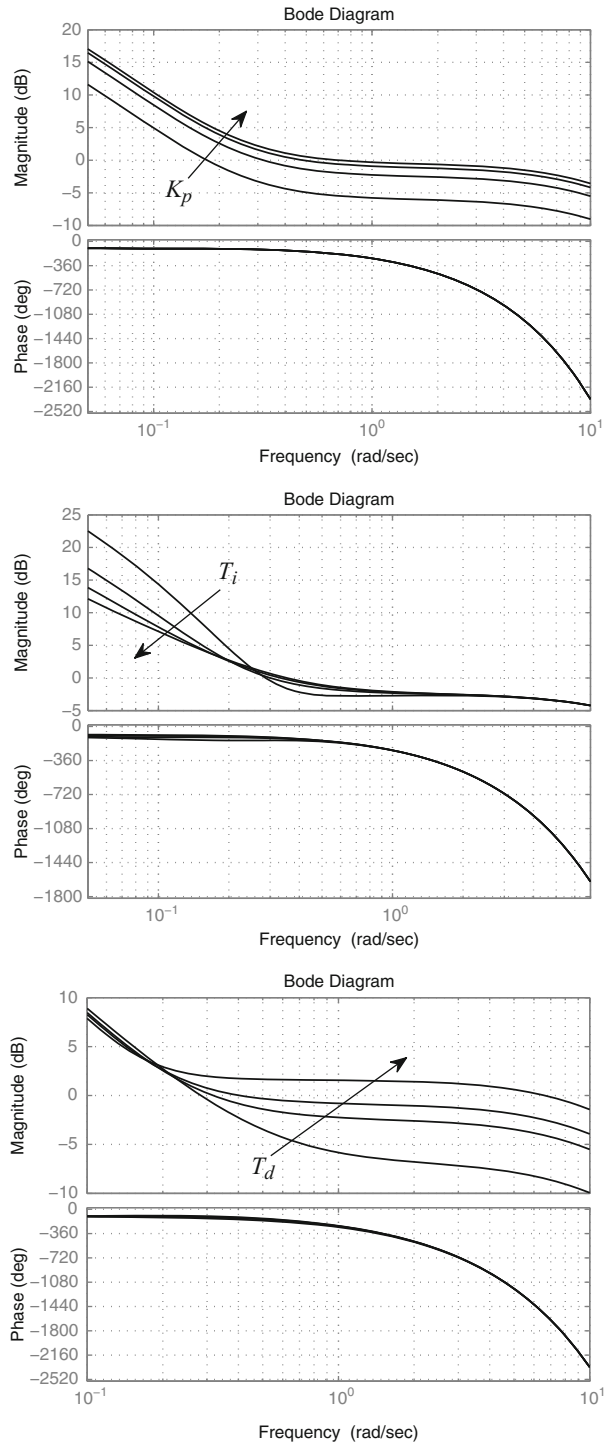
By solving the optimization problem for different kinds of (normalized) processes and by interpolating the results, it has been possible to obtain tuning formulae that give the PID gains based on the process parameters. Actually, it should be noted that as no (robustness) constraints are considered in the optimization procedure, a poor robustness may eventually result in the control system.

Robust Tuning

Recently, tuning rules which explicitly consider the robustness issue have been devised. In particular, the maximum of the sensitivity function is often considered as robustness index. A selected value of M_s is then employed as a constraint in finding the optimal PID parameters which minimize a given performance index. An additional constraint on the maximum complementary sensitivity function can also be considered. For example, the AMIGO tuning rules (Åström and Hägglund 2004) have been devised by applying this approach, where the integral gain is maximized in order to obtain the best reduction of load disturbances.

PID Control, Fig. 3

Example of the effect of an increment of three PID parameters on the loop transfer function Bode plot. The modifications are considered separately, namely, two parameters are fixed, while the other is modified



Automatic Tuning

The functionality of automatically identifying the process model and tuning the controller based on that model is called automatic tuning (or, simply, auto-tuning) (► [Autotuning](#)). In particular, an identification experiment is performed after an explicit request of the operator, and the values of the PID parameters are updated at the end of it (for this reason, the overall procedure is also called *one-shot automatic tuning* or *tuning-on-demand*). The design of an automatic tuning procedure involves many critical issues, such as the choice of the identification procedure (usually based on an open-loop step response or on a relay feedback experiment), of the a priori selected (parametric or non parametric) process model, and of the tuning rule.

The one-shot automatic tuning functionality is available in practically all the single-station controllers available on the market. More advanced control units might provide a *self-tuning* functionality, where the identification procedure is continuously performed during routine process operation in order to track possible changes of the system dynamics and the PID parameters values are adaptively modified. In this case, all the issues related to adaptive control have to be taken into account. In particular, performance assessment methodologies, which are capable to evaluate if the PID design can be improved, are of significant relevance in this context (► [Controller Performance Monitoring](#)).

Design Tools

Although one of the major advantages of PID controllers is their relative simplicity, Computer-Aided Control System Design tools (► [Computer-Aided Control Systems Design: Introduction and Historical Overview](#)) have been developed in order to help the user in their design (starting from the identification of the process) by taking into account the different control requirements in a given application (Guzman et al. 2008). In this context, all the additional functionalities can be considered, as well as more complex control

architectures where, in any case, the PID control is still the basic element (► [Control Structure Selection](#), ► [Control Hierarchy of Large Processing Plants: An Overview](#)).

Summary and Future Directions

PID controllers are the most employed controllers in industry, and the knowledge about their use is well established, with the presence of many effective tuning and automatic tuning techniques. Despite this, PID controllers are still being developed under many points of view. For example, design methodologies for more complex control schemes (like cascade control or control of multivariable systems with or without the use of a decoupling strategy) can be improved. Further, the advancement of the technologies poses new problems that need to be addressed. For example, the use of wireless sensors and actuators calls for event-based PID controllers whose design should take into account the asynchronous sampling. The availability of faster and faster microprocessors has also stimulated an increasing interest in fractional-order PID controllers which allows a more flexible design at the expense of an increment of the complexity.

Cross-References

- [Autotuning](#)
- [Computer-Aided Control Systems Design: Introduction and Historical Overview](#)
- [Control Hierarchy of Large Processing Plants: An Overview](#)
- [Controller Performance Monitoring](#)
- [Control Structure Selection](#)

Recommended Reading

Basic concepts of PID controllers can be found in almost every book on process control. For a detailed treatment, see (Åström and Hägglund 2006) where all the methodological as well as technological aspects are covered. An excellent collection of tuning rules can be found in

O'Dwyer (2006). More advanced topics can be found in Tan et al. (1999), Yu (2006), Johnson and Moradi (2005), Knospe (2006), Visioli (2006), Wang et al. (2008), Visioli and Zhong (2010), and Vilanova and Visioli (2012).

Bibliography

- Åström KJ, Hägglund T (2004) Revisiting the Ziegler–Nichols step response method for PID control. *J Process Control* 14:635–650
- Åström KJ, Hägglund T (2006) Advanced PID control. ISA Press, Research Triangle Park
- Guzman JL, Åström KJ, Dormido S, Hägglund T, Berenguel M, Pigué Y (2008) Interactive learning modules for PID control. *IEEE Control Syst Mag* 28:118–134
- Johnson MA, Moradi MH (eds) (2005) PID control – new identification and design methods. Springer, London
- Knospe C (ed) (2006) PID control. *IEEE Control Syst Mag* 26(1):30–31. Special section
- Nichols NB, Ziegler JG (1942) Optimum settings for automatic controllers. *Trans ASME* 64:759–768
- O'Dwyer A (2006) Handbook of PI and PID tuning rules. Imperial College Press, London
- Skogestad S (2003) Simple analytic rules for model reduction and PID controller tuning. *J Process Control* 13:291–309
- Tan KK, Wang Q-G, Hang CC, Hägglund T (1999) Advances in PID control. Springer, London
- Vilanova R, Visioli A (eds) (2012) PID control in the third millennium: lessons learned and new approaches. Springer, London
- Visioli A (2006) Practical PID control. Springer, London
- Visioli A, Zhong Q-C (2010) Control of integral processes with dead time. Springer, London
- Wang Q-G, Ye Z, Cai WJ, Hang CC (2008) PID control for multivariable processes. Springer, London
- Yu CC (2006) Autotuning of PID controllers: a relay feedback approach. Springer, London

PIDs and Biquads: Practical Control of Mechatronic Systems

Daniel Abramovitch
Mass Spec Division, Agilent Technologies,
Santa Clara, CA, USA

Abstract

The Proportional plus Integral plus Derivative (PID) controller is the most ubiquitous

feedback controller design, in industry, in consumer products, and even in first collegiate controls classes. While PID controllers are the preferred “Brand-X”/whipping boy of numerous doctoral theses on control, most of these comparisons go no further than simulation. In practice, it takes a lot of extra effort to justify replacing a PID controller. This entry discusses the main points of the PID + filter architecture (Abramovitch (2015c) Trying to keep it real: 25 years of trying to get the stuff I learned in grad school to work on mechatronic systems. In: Proceedings of the 2015 Multi-Conference on Systems and Control. IEEE, Sydney, pp 223–250) that is the mainstay of mechatronic control systems.

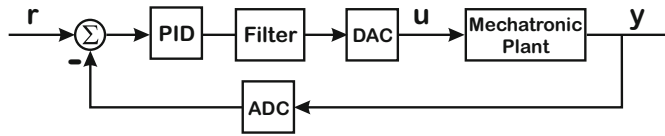
Keywords

PID control · Mechatronics · Filtering · Biquads · System ID

Introduction

Part of the popularity of PID controllers in practice is that many of the systems to which they are applied, whether they be temperature control, pressure control, or chemical process control (CPC), are reasonably modeled as first- or second-order systems. Complexity comes in the form of time delay, physical system changes, and/or nonlinearities. PID controllers are typically more than adequate for most systems modeled by a first- or second-order transfer function (Fig. 1).

Mechatronic systems are defined as systems with tight interactions between mechanical and electrical components. Such systems can have critical frequencies ranging from sub Hertz range up to hundreds of kiloHertz (or even megaHertz for nano-mechanical systems). Unlike the former systems, mechatronic systems often exhibit electrical or mechanical flexibilities that result in resonances and antiresonances when viewed in the frequency domain. While many of these systems have enough separation between the “rigid body” second-order system and the



PIDs and Biquads: Practical Control of Mechatronic Systems, Fig. 1 A practical digital control loop for a single-input, single output (SISO) mechatronic system. The digital controller is often implemented as a PID-

like controller in series with filtering to equalize the effect of high-frequency resonances and antiresonances (Abramovitch 2015c)

higher-frequency dynamics to be controlled using low-frequency PID controllers, they differ from non-mechatronic systems in that these dynamics (resonances and antiresonances) often show up in the desired control bandwidth and must be addressed.

PID Concepts

While PID controllers are the most standard controller used in practice, the PIDs themselves suffer from a lack of standardization. Virtually any control structure that has a proportional gain, an integrator with its own gain, and a differentiator with its own gain can be considered a PID controller, but the specification of gains can have a multitude of styles (Abramovitch 2015d). Consider the following three PID controller equations. The center term for each is the typical “three-parameter” form, while the rightmost term puts the PID in the form of an analog filter. For brevity, we have omitted the cases when a low-pass filter is added to the differentiator term (or on the entire PID), although this would be necessary if the PID were to be implemented in analog (continuous-time) form or discretized with a trapezoidal rule equivalent (Abramovitch 2015d). Still, these three forms are close to the style of specification of most continuous-time, unfiltered PID controllers that may be encountered, either in analysis or in commercial controllers. Recognizing one of these forms, we should be able to easily translate the control parameters into one of the other forms. This is extremely useful in getting an apples-to-apples

comparison of different commercial PID controllers.

$$\begin{aligned} C(s) &= K \left(1 + \frac{1}{T_I s} + T_D s \right) \\ &= K T_D \left(\frac{s^2 + \frac{1}{T_D} s + \frac{1}{T_I T_D}}{s} \right) \end{aligned} \quad (1)$$

$$\begin{aligned} C(s) &= K_P + \frac{K_I}{s} + K_D s \\ &= K_D \left(\frac{s^2 + \frac{K_P}{K_D} s + \frac{K_I}{K_D}}{s} \right) \end{aligned} \quad (2)$$

$$\begin{aligned} C(s) &= K_P + \frac{K_I}{T_I s} + K_D T_D s \\ &= K_D T_D \left(\frac{s^2 + \frac{K_P T_I}{K_D T_I T_D} s + \frac{K_I}{K_D T_I T_D}}{s} \right) \end{aligned} \quad (3)$$

The form of Eq. 1 is typically – but not always – associated with process control applications, temperature, and pressure control (Seborg et al. 2016). There is an overall controller gain, but the relative gains of the three parts are adjusted via the terms T_I and T_D , which nominally are meant to refer to the integration time (time over which the integral takes place) and differentiation time (time over which the derivative takes place), respectively. Even here, the terminology is confusing, since as written the integral and differentiator take place over all time. These terms take the place of the integrator and differentiator gains, although one might naturally assume that integration and differentiation times

would be characteristics of the device, rather than a tunable constant.

The form of Eq. 2 has three independent PID “knobs” with no hint of relation to the integration and differentiation time. The form of Eq. 3 breaks the integration and differentiation times out from the integrator and differentiator gains. While this

last form may seem a bit tedious due to the extra coefficients, it has a distinct advantage when discretized using a backward rectangular rule equivalent (Franklin et al. 1998) where $s \rightarrow \frac{z-1}{T_S z}$. If one sets $T_S = T_I = T_D$, which aligns with integration and differentiation over a single sample period, we get:

$$\begin{aligned} C_B(z) &= K_P + \frac{K_I}{1 - z^{-1}} + K_D(1 - z^{-1}) \\ &= (K_P + K_I + K_D) \left(\frac{1 - \frac{2K_D + K_P}{K_P + K_I + K_D} z^{-1} + \frac{K_D}{K_P + K_I + K_D} z^{-2}}{(1 - z^{-1})z^{-1}} \right) \end{aligned} \quad (4)$$

What is most revealing about this form is the similarity between the structures of Eqs. 3 and 4 as demonstrated in Fig. 2. The continuous and discrete parameters are related via $T_I = T_D = T_S$, while the structure remains essentially the same. Furthermore, the backward rectangular rule discretization has taken a non-proper analog PID form and made it a proper and therefore directly implementable discrete form.

While the denominator of Eq. 4 does not change with PID parameters, the numerator can produce anything from a lag filter (set $K_D = 0$) to a lead filter (set $K_I = 0$) to a notch filter (Abramovitch 2015d; Abramovitch et al. 2008). Figure 3 shows the ranges of a typical mechatronic control system. If the loop is closed well below the resonance, a PI controller may be used. If the loop is closed far above the resonance, a PD controller may be employed. (One might still use a small nonzero I gain to achieve zero steady-state error to a step.) It is when we try to close the loop in the vicinity of the resonance that we need to carefully use all the PID coefficients (Abramovitch et al. 2008), and so we need a far more accurate model than the previous two cases.

Another bridge between PID controllers and advanced methods is to realize that almost any linear SISO state space controller can be formulated as a PID plus a group of matched filters. Looking at it this way, the matched filters correspond to the estimator model, and neither of them works well if the modeling is inaccurate.

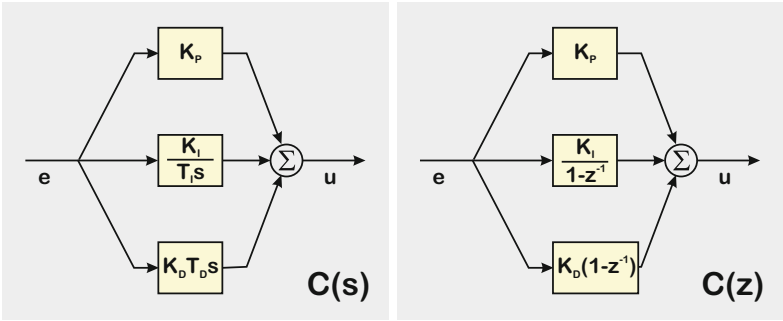
Generally, for systems beyond a double integrator, successful modeling involves a deep dive into measurements.

Filtering Higher-Order Dynamics

One of the distinguishing characteristics of mechatronic control systems is the commonality of flexibility in the physical system that manifests itself in the form of resonances and antiresonances. Generally, these dynamics do not lend themselves to identification via step response methods, relay methods, or most discrete-time, time domain methods (next section). Here we are more concerned with how to compensate for higher-frequency dynamics, and this involves filtering. The more accurate the filter model and (usually) the discrete implementation of that model, the more effectively we can control the system (Fig. 4).

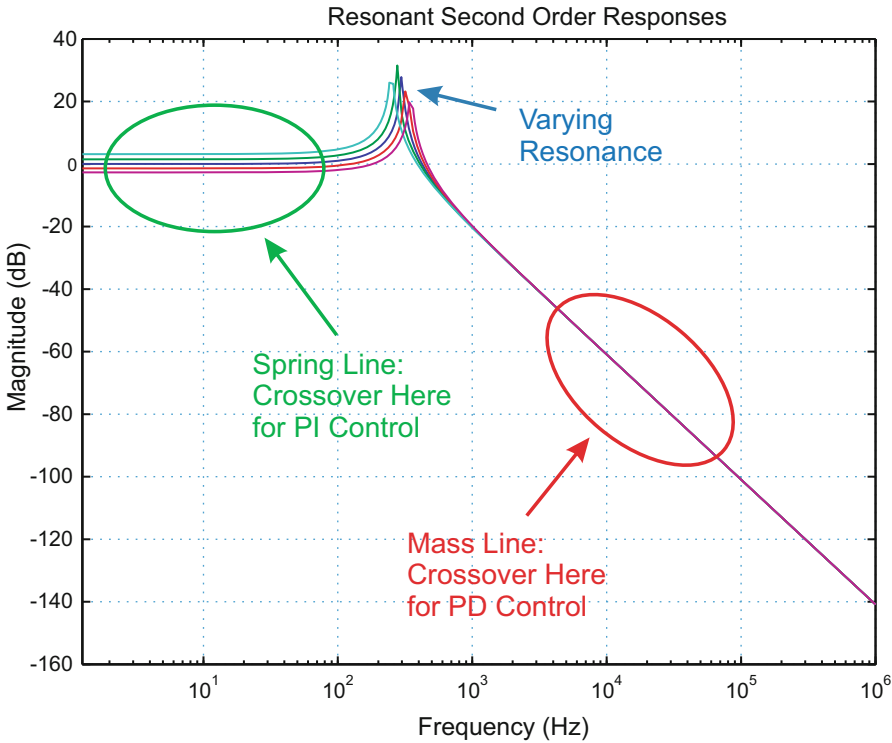
Identifying a Plant Model

The key issue in being able to push the performance of any control design is a deep understanding and accurate model of the physical system. Traditionally, PID controllers were attached to slow process systems, and the gains were manually adjusted until the closed-loop system oscillated and then reduced back down. A well-known



PIDs and Biquads: Practical Control of Mechatronic Systems, Fig. 2 A parallel form topology of a simple analog PID controller (left). The digital PID controller

(right) shows much of the same structure. The similarity between these is not coincidental, but owes to how most PID designs are discretized

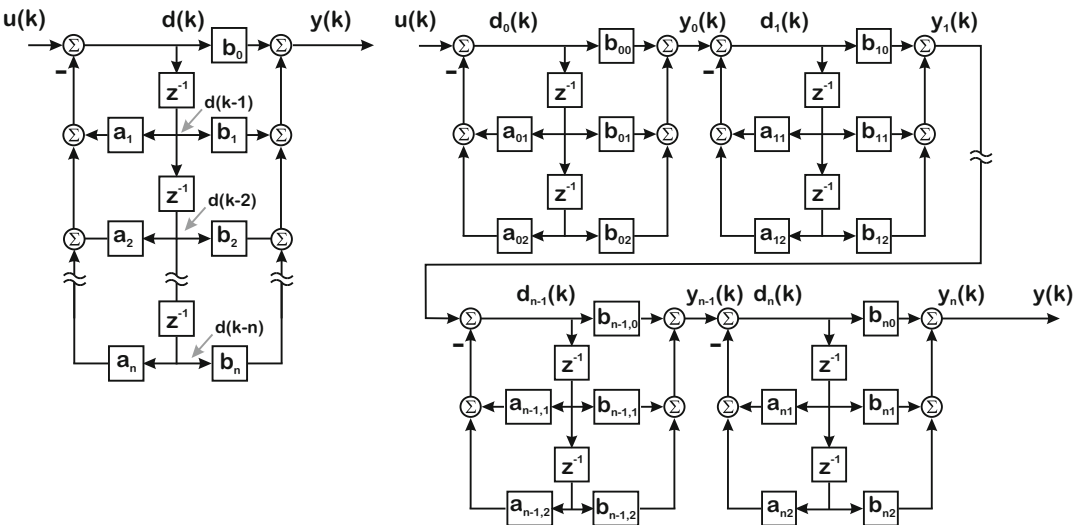


PIDs and Biquads: Practical Control of Mechatronic Systems, Fig. 3 The typical ranges in a mechatronic system and how they relate to PID control

procedure for this is Ziegler-Nichols, but other methods have supplanted it.

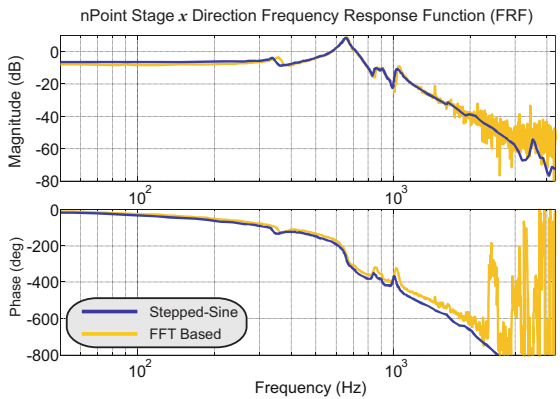
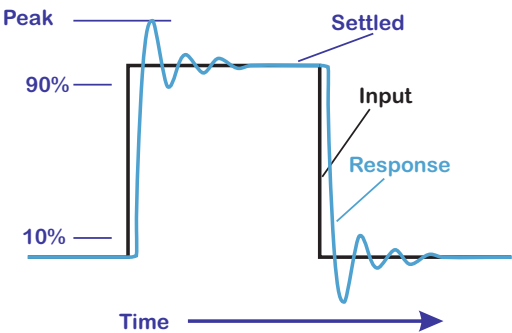
The first is to apply a step to the input of the physical system (left side of Fig. 5), observe the response, and extract dynamic parameters from these. In fact, step response methods are the main way of characterizing slow process systems that are not well suited for frequency response

methods. The issues with step response are that while they can be performed on any order system, one can only reasonably extract parameters from them if one assumes either a first- or second-order model ahead of time and then extracts the parameters associated with that model. The relay method replaces the controller in the loop by a signum function (the relay) to force an oscillation



PIDs and Biquads: Practical Control of Mechatronic Systems, Fig. 4 On the left: a high-order digital filter used in compensation implemented as a ratio of discrete polynomials. On the right: the same filter broken up into a

series of biquad filters which are more numerically stable and retain more of the original physical system dynamics structure



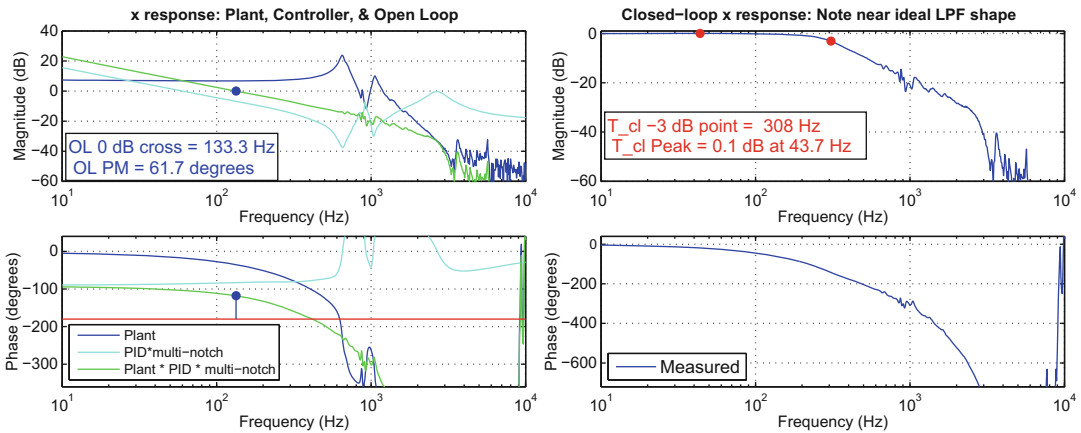
PIDs and Biquads: Practical Control of Mechatronic Systems, Fig. 5 On the left, zoomed in step response of stable second-order system with low damping and transport delay. On the right, a pair of frequency response func-

tions (FRFs) for the X axis of an AFM stage, computed via fast Fourier transform (FFT) and stepped-sine methods (Abramovitch 2015a). (Courtesy: Jeff Butterworth.)

of the closed-loop test system so as to back out the plant dynamics (Åström and Hägglund 2005). However, as with step response methods, it is only applicable when there are no significant resonances in the loop.

In this area, the most successful methods involve injecting signals across multiple frequencies and relating the output response to the input resulting in a frequency response function (FRF) (sometimes referred to in the academic literature as the empirical transfer function estimate). This

FRF may either be used in direct frequency domain design or to generate a parametric system model for control system design. The limits on these methods are often based on the sharpness of these extra dynamics (relating to their level of damping) and their proximity to one another. Different frequency domain methods have different levels of resolution, with the stepped-sine (known alternately as sine-dwell or in industry as swept-sine) method being the gold standard (Abramovitch 2015a). Frequency



PIDs and Biquads: Practical Control of Mechatronic Systems, Fig. 6 On the left: measured plant, compensator (PID + multinotch), and generated open-loop frequency response for AFM x-axis actuator. The PID and multinotch are automatically tuned to generate an open-loop response that looks like an integrator over the frequency range of interest, allowing the open-loop

crossover to be set subject to phase-margin constraints (Abramovitch 2015c). On the right: the measured closed-loop response. Note the nice, low-pass look of the closed-loop response and the minimal peaking. The closed-loop bandwidth can be adjusted from open-loop constraints or closed-loop peaking constraints

domain methods require that the input to the physical system have a rich spectral content and have a presumption that this will cause the output to also be spectrally rich. While methods based on the fast Fourier transform (FFT) are computationally faster than stepped-sine methods, the right side of Fig. 5 shows that when applicable, the stepped-sine produces much cleaner results. There are some drawbacks to any frequency domain method. The first is that the frequency response usually has to be converted to a parametric model in order to do design (Abramovitch et al. 2008). The second is that they are not really appropriate for heavily nonlinear systems or for systems with extremely slow time constants, such as those found in chemical process control or biological systems.

The critical point here is that the ability to push the performance of the control system design is directly tied to the confidence one has in the system model, and the latter depends very strongly on the types of measurements one can make on the system. Frequency response methods work very well in mechatronic systems, but they are less common in slow process applications where the time constants are on the orders of seconds or minutes.

Putting It All Together: An Example

Figure 6 shows an example that puts together the ideas of the previous sections. The system is identified using a stepped-sine method built into the controller (Abramovitch 2015a) resulting in the curve in the left plot. The clean response allows a controller to be generated (cyan curve) that equalizes the open-loop response (green curve) to look like an integrator up to the point at which the measurement gets too noisy. The controller is segmented into a PID to handle the lower-frequency behaviors and a series of biquads known as a multinotch (Abramovitch 2015b). The net result of this is that the open-loop crossover can be relatively close to the plant resonance and the closed-loop response (right plot) looks like a very clean low-pass filter with minimal peaking.

Summary

When something keeps working, over and over again across multiple situations, one can assume that that something is pretty effective. The class of PID controllers, despite being the whipping

boy of many a graduate student's doctoral thesis – is the most commonly used control law design because it can easily be applied to control second-order systems. Combining that with engineers' ability to beat many problems into second-order systems, we have a winning combination. For mechatronic systems, those percussions often involve adding filters to equalize out the open-loop dynamic response, allowing the PID to control the lower-frequency behavior of the physical system, while the filters (often implemented as string of biquads) compensate for the higher-frequency terms. The accuracy to which this can be done often depends on the knowledge of the physical system model, but even then a PID plus filters approach can often yield excellent results in practical systems.

Cross-References

- [Classical Frequency-Domain Design Methods](#)
- [Control Structure Selection](#)
- [Control Systems for Nanopositioning](#)
- [Control System Optimization Methods in the Frequency Domain](#)
- [PID Control](#)

Bibliography

- Abramovitch DY (2015a) Built-in stepped-sine measurements for digital control systems. In: Proceedings of the 2015 Multi-Conference on Systems and Control. IEEE, Sydney, pp 145–150
- Abramovitch DY (2015b) The Multinotch, part I: a low latency, high numerical fidelity filter for mechatronic control systems. In: Proceedings of the 2015 American Control Conference, AACC. IEEE, Chicago, pp 2161–2166
- Abramovitch DY (2015c) Trying to keep it real: 25 years of trying to get the stuff I learned in grad school to work on mechatronic systems. In: Proceedings of the 2015 Multi-Conference on Systems and Control. IEEE, Sydney, pp 223–250
- Abramovitch DY (2015d) A unified framework for analog and digital PID controllers. In: Proceedings of the 2015 Multi-Conference on Systems and Control. IEEE, Sydney, pp 1492–1497
- Abramovitch DY, Hoen S, Workman R (2008) Semi-automatic tuning of PID gains for atomic force microscopes. In: Proceedings of the 2008 American Control Conference, AACC. IEEE, Seattle
- Åström KJ, Hägglund T (2005) Advanced PID control. Oxford series on optical and imaging sciences. ISA Press, Research Triangle Park
- Franklin GF, Powell JD, Workman ML (1998) Digital control of dynamic systems, 3rd edn. Addison Wesley Longman, Menlo Park
- Seborg DE, Edgar TF, Mellichamp DA, Doyle FJ (2016) Process dynamics and control, 4th edn. Wiley, Hoboken

Pilot-Vehicle System Modeling

Aleksandr Efremov

Moscow Aviation Institute, Moscow, Russia

Abstract

The main types and variables of pilot-aircraft systems and pilot control response characteristics are considered. The basic regularities of pilot behavior exposed in closed-loop systems are briefly discussed. Different types of models of pilot behavior are reviewed including classical (McRuer), structural, and optimal control models.

Keywords

Pilot behavior · Manual control · Describing function · Remnant spectral density · Crossover pilot model · Structural model · Pilot optimal control model

Introduction

The effective use of piloted flight vehicles has always required a satisfactory match of vehicle characteristics (which include vehicle dynamics, flight control system, inceptors, displays) with the human pilot's characteristics as a flight controller. The provision of proper vehicle handling qualities by the flight control system and display and manipulator design has often posed serious

problems which the vehicle system engineer must solve.

The solutions of the problems require the knowledge of mutual interactions between the pilot and the vehicle. The understanding of such interactions requires a mathematical theory which can be used to explain known findings and to predict new ones. For handling qualities, such a theory should be based on the methods of control engineering and treat the pilot-vehicle system as a closed-loop (in general, a multiloop) entity. The sine qua non of the theory is a model of pilot dynamic characteristics in a form suitable for application using relatively conventional control engineering techniques. An adequate description of a pilot's dynamics response characteristics is not easily obtained because of the pilot's inherent adaptability and capacity for learning.

Main Variables of the Pilot-Aircraft System

The pilot-aircraft manual control system, shown in Fig. 1, is characterized by a number of variables. The main group of these variables is the so-called task variables, which comprise all the system inputs (command inputs i , disturbances d) and control system elements (display, manipulators, and controlled element dynamics, which is defined by the aircraft frame and flight control system dynamics).

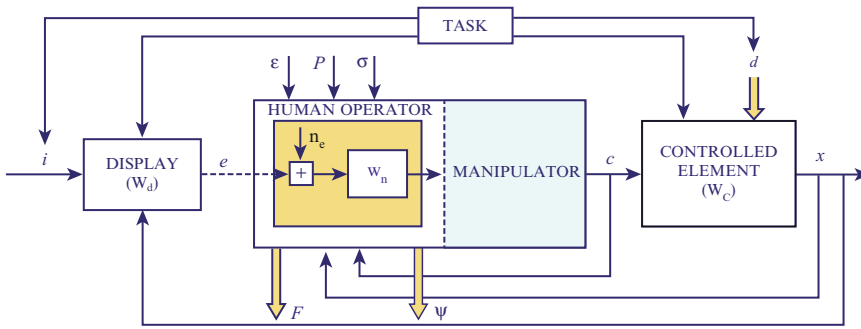
A specific feature of pilot-aircraft systems is the dependence of the piloting task on the task variables. For different piloting tasks, these variables or their parameters differ too. Stability of the closed-loop system is always a necessary, though not sufficient, criterion for the control strategy. Consequently, the pilot's dynamics are profoundly affected by the display and controlled element dynamics, because pilot response must be adapted to provide the necessary loop stability and accuracy. The characteristics of the other task variables (i , d), related to the mission and control strategy, also exert direct influence on the pilot dynamics, although their effects are more in the nature of adjustment and emphasis than of changes in fundamental regularities of pilot behavior.

These variables constitute an enormous range of possible conditions and piloting tasks. In addition to the task variables, the other groups of variables – procedural (p – instructions, training schedule order of presentation of trials, etc.), environmental (ε – illumination, vibration, temperature, and so forth), pilot centered (σ – physical condition, motivation, etc.) – have less influence on pilot-aircraft system features.

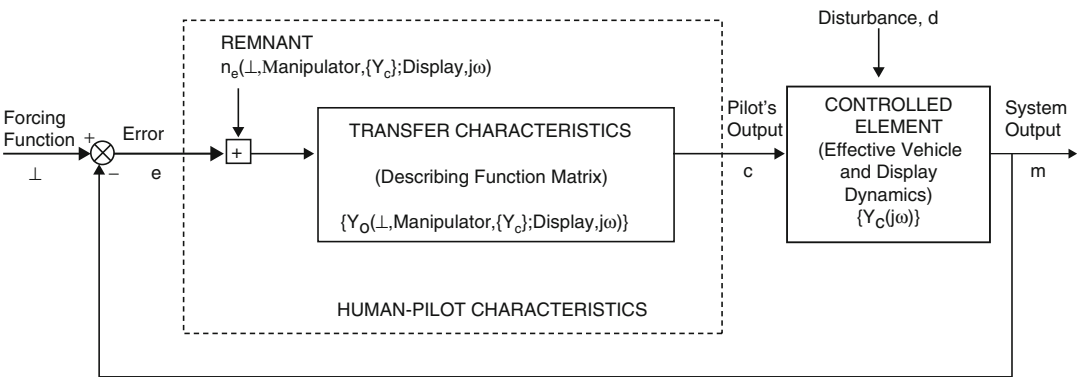
Types of Pilot-Aircraft Systems

The structure of the pilot-aircraft system depends on the piloting task. Some tasks (e.g., the pitch tracking task) can be interpreted with the help of the single loop compensatory block diagram (Fig. 1). In that case the pilot perceives only the error signal, $y(t) = e(t) = i(t) - x(t)$. The pitch angle tracking task can be interpreted in this way. The other tasks require more complicated descriptions. For example, the landing task is a multiloop compensatory task, where the inner loop closed by the pilot is the pitch control loop. Some piloting tasks are multichannel control tasks, in which the pilot perceives several visual stimuli (e.g., pitch angle and bank angle) and generates commands in several channels too. Pilots also perceive stimuli of different sensing modalities (visual, vestibular, proprioceptive). In cases where these influence the pilot's actions, the multimodality of the pilot-aircraft system has to be analyzed.

Many past experiments, in which human dynamic measurements were taken, have been conducted for investigation of compensatory tracking tasks. Some practical piloting tasks (e.g., aim-to-aim tracking in case when the target flies against a background of clouds) correspond to pursuit conditions. In that case, the pilot perceives the error signal $e(t)$ and the input signal $i(t)$. When flying through a mountain gorge or while driving a car, the operator sees the future trajectory $i(t + t^*)$ with preview over a certain time interval t^* . Previous studies of pilot operator behavior in preview tracking tasks demonstrated the considerable difference of the pilot and the pilot-aircraft system characteristics in comparison with the characteristics measured



Pilot-Vehicle System Modeling, Fig. 1 Pilot-aircraft system



Pilot-Vehicle System Modeling, Fig. 2 Quasi-linear paradigm for the human pilot

in compensatory tracking tasks. In particular, the tracking performance (variance of error) is lower 2.5 to 3.5 times when $t^* = 0.5 \div 1$ s.

In many piloting tasks the single loop compensation system defines the main features of more complicated types of pilot-aircraft systems and its flying qualities. Therefore, this type of the system has been investigated in more depth.

Pilot Control Response Characteristics

The most obvious aspect of human dynamic behavior in a manual control task is the pilot's control actions within that task. When the key variables are fixed and the signals in the control loop are approximately time invariant over an interval of interest, the pilot-vehicle system can be presented as a quasi-linear system. In that case, the pilot response can be presented by

two components: the pilot-describing function, $W_p(j\omega)$, taking into account the linear portion of pilot response on the stimulus $e(t)$, and remnant $n_e(t)$, which takes into account all nonlinear, nonstationary effects of pilot behavior (Fig. 2).

In the majority of piloting tasks, $n_e(t)$ is a stationary process characterizing the remnant spectral density $S_{n_en_e}(\omega)$ (McRuer and Krendel 1974). The pilot control response characteristics $W_p(j\omega)$ and $S_{n_en_e}(\omega)$ depend explicitly on the task variables (McRuer and Jex 1967; McRuer et al. 1968). In much experimental research, the technique for identification of these characteristics was based on the use of an input signal consisting of the sum of non-harmonically related sine waves with cut-off frequency ω_i at 1.5, 2.5, and 4 rad/s and different controlled element dynamics (Allen and Jex 1972; Magdaleno 1972; Shirley 1969).

In addition to control response, other types of responses also characterize pilot's behavior: physiological (F) and psychophysiological ψ responses (Fig. 1). For one of the psychophysiological response characteristics, the pilot opinion rating (PR) is widely used in experimental investigations as well as for the measurement of pilot control response. Pilot opinion ratings are defined by specialized scales (e.g., the Cooper-Harper scale Cooper and Harper 1969).

Modeling Pilot Behavior in Manual Control

Experimental investigations have demonstrated a specific regularity: for a variety of forcing functions and controlled elements the slope of the open-loop describing function $|W_{OL}(j\omega)|$ vs frequency was *unity*, that is, -20 dB/dec in the region of the crossover frequency ω_c (McRuer and Jex 1967). This observation has led to the conclusion that near ω_c , $W_{OL}(j\omega)$ can be presented by the “crossover model” (McRuer and Jex 1967)

$$W_{OL}(j\omega) = W_p(j\omega) \cdot W_C = \frac{\omega_c}{j\omega} e^{-j\omega\tau_e}$$

This model has two parameters:

$$\omega_c = \omega_{co}(\omega_c) + \Delta\omega(\omega_i)$$

$$\tau_e = \tau_o(\omega_c) + \Delta\tau(\omega_i)$$

For the controlled element dynamics $W_C = \frac{K}{s(Ts+1)}$, the increase of constant T leads to an increase of τ_0 and a decrease of ω_{C0} . The empirical dependences $\Delta\omega_c$ and $\Delta\tau_e$ on ω_i obtained for the rectangular form of input spectrum are the following: $\Delta\omega_c = 0.18\omega_i$, $\Delta\tau = -0.07\omega_i$.

McRuer proposed several modifications of the open-loop system crossover and pilot describing function models with different levels of accuracy of human neuromuscular system description (McRuer and Krendel 1974). One of the simplest ones (used widely by many researchers), which might be recommended for

description of pilot-aircraft system characteristics in the crossover frequency range, is the following

$$W_p(j\omega) = K_p \frac{T_L j\omega + 1}{T_1 j\omega + 1} e^{-j\omega\tau_e}$$

The selection of the parameters K_p , T_L , and T_1 is carried out by using “adjustment rules” so that the closed-loop system conforms to experimental frequency response characteristics. These adjustment rules reflect the main features of pilot behavior – adaptation and optimization.

A more complicated model of pilot describing function (“structural model”) was offered by R. Hess (1979, 1984). It takes into account the additional inner loop generated by the pilot as a result of his response to proprioceptive cues. The modification of this model (Efremov and Tjaglik 2011) (Fig. 3) demonstrated good agreement with the pilot describing function as measured in experiments. One of the features of this modified model is the criterion used for the parameter optimization: $I = \min[\sigma_e^2]$ or $I = \min[\sigma_e^2 + \beta\sigma_n^2]$. This procedure requires the knowledge of the pilot remnant spectral density. For the single loop system, such a model was developed by Levison et al. (1969).

$$S_{n_e n_e}(\omega) = 0.01\pi \frac{\sigma_e^2 + \sigma_e^2 T_L^2}{1 + T_L^2 \omega^2}$$

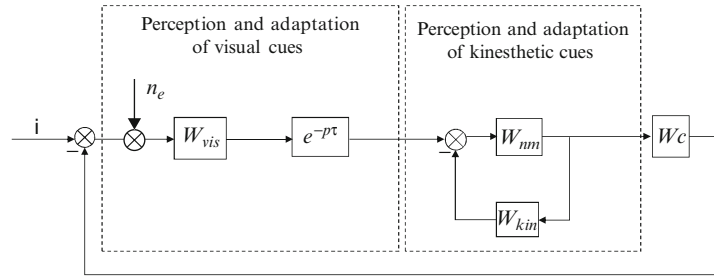
In a limited number of researches, the classic approach to pilot modeling considered above was used for more complicated types of the pilot-aircraft system, when the pilot perception of motion cues was taken into account (multimodality system Hess 1990) or for a case of the multi-loop pilot-aircraft system (Stapleford et al. 1967).

A different approach to pilot behavior modeling was developed by Kleiman et al. (1970). It is based on the modern optimal control theory and assumes that the pilot's goal is to minimize the cost function:

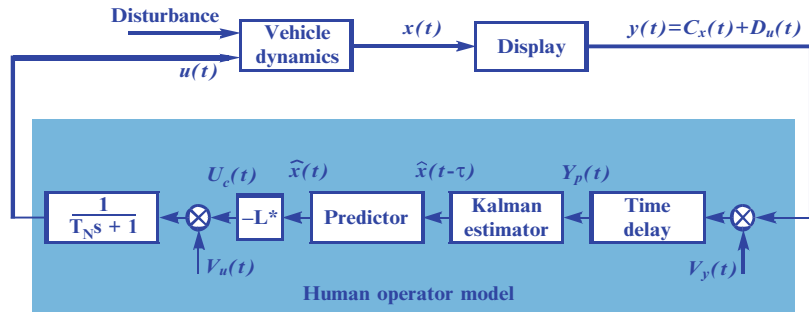
$$I = \lim_T \frac{1}{T} \int_0^T (x Q x^T + u Q_c u^T + \dot{u} G_c \dot{u}^T) dt$$

The model takes into account the main pilot limitation parameters: time delay in perception,

Pilot-Vehicle System Modeling, Fig. 3 Pilot structural model



Pilot-Vehicle System Modeling, Fig. 4 Optimal pilot model



the observation and motor noises, and the neuromuscular dynamics.

The predictive part of the model consists of the optimal controller ($-L^*$), Kalman filter, and predictor (Fig. 4). The software for definition of these elements allows the use of this model for the different types of structural the pilot-aircraft systems.

The classical, structural, and optimal pilot behavior models have been applied widely for different manual control tasks: the development of alternative criteria for flying qualities (Neal and Smith 1971; Efremov et al. 1998), the flight control system (Schmidt 1979) and display design (Klein and Clement 1973; Efremov and Tjaglik 2011; Mulder 1999; Grunwald 1985), exposition of the reasons of preview, and the analysis of reasons for pilot-induced oscillation (McRuer 1997) and the development of means for its suppression (Efremov 1995). A more precise pilot structural model taking into account the model of neuromuscular system and proprioceptive loop was used for the exposition of the reasons of biodynamic interaction effects, the influence of the type of inceptors and feel system dynamics on the pilot-aircraft system characteristics (Zaichik 2014).

In some of the researches, attempts have been made to find the relationship between the parameters of the closed-loop system, pilot control response characteristics, and pilot opinion ratings. The technique developed in these researches is called the “paper pilot technique” (Anderson 1970). The following modification of this technique has enabled a close match between the results of mathematical modeling (PR, T_L , variance of error, etc.) of the different types of the pilot-aircraft system and the results of experimental investigations (Efremov and Ogloblin 2006).

Summary and Future Directions

Pilot behavior has been studied extensively for single-loop stationary manual control tasks. Two approaches to the mathematical modeling of the pilot behavior have been developed: classical and optimal control. Both of them have produced good agreement with experimental results. The discussed models describe one of the main features of the pilot adaptation – “parameter adaptation,” when a change of any task variable causes a change of human operator control response characteristics. Only a limited number

of experimental investigations have been carried out for more complicated cases: multiloop and multimodality pilot-aircraft closed-loop systems. Broader investigations are necessary in the future to obtain accurate pilot mathematical models for these cases. Future investigation of pilot behavior modeling is also necessary for better formulations of other aspects of pilot adaptation:

- “Structural adaptation,” that is, when the pilot selects the loops and the best type of behavior (compensatory, pursuit, preview, etc.) appropriate for the different task variables and, in the case of the flight control system, changes in dynamics.
- “Goal adaptation,” that is, when a change of the piloting task or a failure in the controlled element dynamics is accompanied by a change of the goals.

Other future directions in pilot modeling are the development of models to predict the results in the case of sharp changes of controlled element dynamics, the optimization of the controlled element dynamics, the definition of the relationship between the pilot’s control response characteristics and his/her opinion rating in different piloting tasks, getting new criteria for the handling qualities, prediction of pilot-induced oscillations, and solving many other manual control problems.

Cross-References

- [Adaptive Human Pilot Models for Aircraft Flight Control](#)
- [Aircraft Flight Control](#)

Bibliography

- Allen RW, Jex H (1972) A simple Fourier analysis technique for measuring the dynamic response of manual control systems. *IEEE Trans Syst Man Cybern SMC* 2(5):638–643
- Anderson RO (1970) A new approach to the specification and evaluation of flying qualities. AFFDL-TR-69-120, Wright-Patterson AFB, Air Force Flight Dynamics Lab, June 1970
- Cooper GE, Harper RP (1969) The use of pilot rating in the evaluation of aircraft handling qualities. NASA TN-D-5153. Moffett Field, NASA Ames Research Center, Apr 1969
- Efremov AV (1995) Development and application of the methods for pilot aircraft system research to the manual control tasks of modern vehicles. In: AGARD conference proceedings No. 556 Dual usage in military and commercial technology in guidance and control, Oct 1995
- Efremov AV, Ogloblin AV (2006) Progress-in-the-loop investigations for flying qualities prediction and evaluation. ICAS Congress, Hamburg, Sept 2006
- Efremov AV, Tjaglik MS (2011) The development of perspective displays for highly precise tracking tasks. In: Holzapfel F, Theil S (eds) *Advances in aerospace guidance, navigation and control*. Springer, Berlin/Heidelberg, pp 163–174
- Efremov AV, Ogloblin AV, Predtechensky AN, Rodchenko VV (1992) Pilot as a dynamic system. *Mashinostroeniye*, Moscow, pp 1–343
- Efremov AV, Ogloblin AV, Koshelenko AV (1998) Evaluation and prediction of aircraft handling qualities. A collection of technical papers AIAA atmospheric flight mechanics conference and exhibit. AIAA – 98 – 4145, Boston, 10–12 Aug 1998
- Grunwald AI (1985) Predictor laws for pictorial displays. *J Guid Control Dyn* 8(5):545–552
- Hess R (1979) Structural model of the adaptive human behavior. *J Guid Control* 3(5):416–423
- Hess R (1984) The effects of time delay on systems subject to manual control. *J Guid Control Dyn* 7: 165–174
- Hess R (1990) A model of human use of motion cues. *J Guid Control Dyn* 13(3):476–486
- Kleiman D, Baron S, Levison W (1970) An optimal control model of human response, parts 1, 2. *Automatica* 6(3):357–369
- Klein R, Clement W (1973) Application of manual control display theory to the development of flight director systems for STOL aircraft AFFDL-72-152
- Levison W, Baron S, Kleiman D (1969) A model for controller remnant. *IEEE Trans MMS-10*(4): 101–108
- Magdaleno RE (1972) Serial segments method for measuring remnant. *IEEE Trans Syst Man Cybern SMC* 2(5):674–678
- McRuer DT (1997) Aviation safety and pilot control understanding and preventing unfavorable pilot-vehicle interactions. National Academy Press, Washington, DC
- McRuer DT, Jex HR (1967) A review of quasilinear pilot models. *IEEE Trans HFE-8*(3):231–249
- McRuer DT, Krendel ES (1974) Mathematical models of human pilot behavior. *AGARDograph* 188:1–72
- McRuer DT, Hofmann LG, Jex HR et al (1968) New approaches to human pilot/vehicle dynamic analysis. AFFDL-TR-67-150

- Mulder M (1999) Cybernetics of tunnel in the sky display. Ph.D. thesis, Delft
- Neal TP, Smith RE (1971) A flying qualities criterion for the design of a fighter flight control systems. *J Aircraft* 8(10):803–809
- Schmidt D (1979) Optimal flight control synthesis via pilot modeling. *J Guid Control Dyn* 4(2): 308–312
- Shirley R (1969) Application of modified fast Fourier transform to calculate human operator describing functions. *IEEE Trans Man-Machine Syst* MMS-10(4):140–144
- Stapleford R, Ashkenas J et al (1967) Analysis of several handling quality topics pertinent to advanced manned aircraft. AFFDL-TR-67-2, Wright-Patterson ASB, Air Force Flight Dynamics Lab, June 1967
- Zaichik LE et al (2014) Effect of manipulator type and feel system characteristics on high frequency biodynamic pilot-aircraft interaction. 28 ICAS Congress, Saint Petersburg, Sept 2014

Polynomial/Algebraic Design Methods

Vladimír Kučera

Faculty of Electrical Engineering, Czech
Technical University of Prague, Prague, Czech
Republic

Abstract

Polynomial techniques have made important contributions to systems and control theory. Algebraic formalism offers several useful tools for control system design. In most cases, control systems are designed to be stable and to meet additional performance specifications, such as optimality or robustness. The basic tool is a parameterization of all controllers that stabilize a given plant. Optimal or robust controllers are then obtained by an appropriate selection of the parameter. An alternative tool is a reduction of controller synthesis to a solution of a polynomial equation of specific type. These two polynomial/algebraic approaches will be presented as closely related rather than isolated alternatives.

Keywords

Controller synthesis · Linear systems · Polynomial equation approach to control system design · Youla-Kučera parameterization of stabilizing controllers

Stabilizing Controllers

The majority of control problems can be formulated using the diagram shown in Fig. 1. Given a plant S , determine a controller R such that the feedback control system is stable and satisfies some additional performance specifications, such as reference tracking, disturbance attenuation, optimality, or robustness.

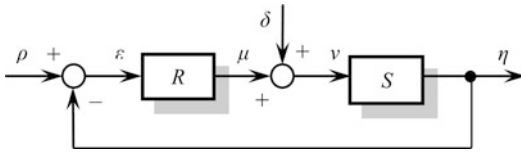
Suppose that the plant and the controller are linear time-invariant single-input single-output continuous-time systems with real *rational* transfer functions S and R , respectively. Stability is understood as the *input-output stability*, i.e., whenever the exogenous inputs δ and ρ are essentially bounded in amplitude, so too are the output signals μ and η (hence also ε and ν).

It is natural to separate the design task into two consecutive steps: (1) stabilization and (2) achievement of additional performance specifications. To do this, all solutions of the first step, i.e., *all controllers that stabilize the given plant*, must be found.

How can one characterize such controllers? Denote H_s the reference-to-error transfer function (sometimes called the sensitivity function) and H_c the disturbance-to-control transfer function (the so-called complementary sensitivity function) in the closed-loop control system, namely,

$$H_s = \frac{1}{1 + SR}, \quad H_c = \frac{SR}{1 + SR}.$$

Now suppose that S can be expressed as the ratio of two coprime polynomials, $S = b/a$, and that the controller has alike form, $R = q/p$. Then the two closed-loop transfer functions can be written as



Polynomial/Algebraic Design Methods, Fig. 1
Feedback control system

$$H_s = a \frac{p}{ap + bq} := aX,$$

$$H_c = b \frac{q}{ap + bq} := bY$$

Consequently, if R stabilizes S , then the rational functions X and Y are bound to be stable. These functions cannot be arbitrary, however, since $H_s + H_c = 1$. The stability equation follows as

$$aX + bY = 1.$$

Any stabilizing controller for S can be expressed as $R = Y/X (= q/p)$, where X and Y are a stable rational solution pair of the stability equation. This solution can be expressed in parametric form:

$$X = x + bW, \quad Y = y - aW,$$

furnishing in turn an explicit parameterization of the set of all stabilizing controllers for S :

$$R = \frac{y - aW}{x + bW},$$

known as the *Youla-Kučera parameterization*. Here x and y are any polynomials satisfying the Bézout equation $ax + by = 1$, while W is a free parameter ranging over the set of stable real rational functions such that $x + bW$ is not identically zero.

Example 1 Consider an integrator plant $S(s) = 1/s$. The Bézout equation admits a solution $x = 0$, $y = 1$ so that the set of all stabilizing controllers for S is given by

$$R(s) = \frac{1 - sW}{W}$$

for any stable real rational $W \neq 0$.

The parameter

$$W(s) = \frac{1}{s + 1}$$

yields $R = 1$, a proportional gain controller. The parameter

$$W(s) = \frac{s}{s^2 + s + 1}$$

results in a proportional-integral controller

$$R(s) = 1 + \frac{1}{s}.$$

Taking $W = 1$ leads to the stabilizing controller $R(s) = 1 - s$. The feedback system is stable, but it has a pole at $s = \infty$.

Additional Performance Specifications

There is a simple formula that generates all the stabilizing controllers for a given plant. Using this formula, we can obtain a parameterization of all stable closed-loop transfer functions that can be obtained by stabilizing a given plant. The bonus is that the parameterization is *affine* in the free parameter W . In contrast, the controller R appears in a nonlinear fashion:

$$\begin{aligned} \begin{bmatrix} v \\ y \end{bmatrix} &= \frac{1}{1 + SR} \begin{bmatrix} 1 & R \\ S & SR \end{bmatrix} \begin{bmatrix} d \\ r \end{bmatrix} \\ &= \begin{bmatrix} a(x + bW) & a(y - aW) \\ b(x + bW) & b(y - aW) \end{bmatrix} \begin{bmatrix} d \\ r \end{bmatrix}. \end{aligned}$$

As R and W are in a one-to-one correspondence, it is convenient to use W in lieu of R in the design process and calculate R subsequently. Thus, the parameterization of all stabilizing controllers makes it possible to separate the design process into two steps: the determination of all stabilizing controllers and the selection of the parameter that achieves the remaining design specifications. The extra benefit is that both tasks are linear.

Asymptotic Properties

Asymptotic properties of control systems can easily be accommodated in the sequential design procedure. These include the elimination of an offset due to step references, the ability of system output to follow a class of reference signals, or the asymptotic elimination of specific disturbances.

In Fig. 1, asymptotic *reference tracking* means that the output η follows the reference ρ as time approaches infinity, which is to say that the error ε approaches zero for large times. On the other hand, we speak of asymptotic *disturbance elimination* if the effect of the disturbance δ decreases at the output η for increasing time. In terms of Laplace transforms, $\varepsilon = H_s \rho$ and $\eta = SH_s \delta$ are to be *stable* rational functions.

Example 1 Consider the plant $S(s) = 1/(s+1)$. The Bézout equation admits a solution $x = 0$, $y = 1$. The set of all stabilizing controllers for S is

$$R(s) = \frac{1 - (s+1)W}{W}$$

for any stable real rational $W \neq 0$. The achievable sensitivity transfer functions are $H_s = (s+1)W$.

To track a step reference, $\rho = 1/s$, we must take $W = sW_1$ for any stable rational $W_1 \neq 0$. To eliminate a sinusoidal disturbance, $\delta = s/(s^2 + \omega^2)$, we constrain the parameter as $W = (s^2 + \omega^2)W_2$ for any stable rational $W_2 \neq 0$. To meet both requirements, we simply take $W = s(s^2 + \omega^2)W_3$ for any stable rational $W_3 \neq 0$, say $W = s(s^2 + \omega^2)/(s+1)^4$.

The resulting controller is

$$R(s) = \frac{3s^3 + (6 - \omega^2)s^2 + (4 - \omega^2)s + 1}{s(s^2 + \omega^2)}.$$

The controller obtained in Example 1 demonstrates the internal model principle: the unstable modes to be followed or eliminated must be generated by the controller unless they are present in the plant.

H₂ Optimal Control

The sequential design procedure will be further illustrated on the design of *linear-quadratic*

optimal controllers. Given a plant with transfer function $S = b/a$, the task is to find a controller that stabilizes the control system of Fig. 1 while minimizing the H_2 norm of some closed-loop transfer function, say of the complementary sensitivity function H_c .

The H_2 norm is defined for any strictly proper rational function G analytic on the imaginary axis as

$$\|G\|_2 = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} |G(j\omega)|^2 d\omega}.$$

The set of complementary sensitivity functions that can be achieved in the stabilized control system is

$$H_c = b(y - aW),$$

where W is a free stable rational parameter. The parameter will be selected so as to minimize the H_2 norm of H_c .

Let $\alpha\beta$ be a polynomial defined by keeping the stable (in $\operatorname{Re} s < 0$) zeros of ab while replacing the unstable (in $\operatorname{Re} s \geq 0$) ones with their negative values. Then $ab/\alpha\beta$ is inner (or all pass) and

$$\|H_c\|_2 = \left\| \frac{\alpha\beta}{ab} H_c \right\|_2 = \left\| \frac{\alpha y \beta}{a} - \alpha W \beta \right\|_2.$$

Consider the decomposition

$$\frac{\alpha y \beta}{a} = r + \frac{q}{a}$$

where r is a polynomial and q/a is strictly proper. With this decomposition,

$$\|H_c\|_2^2 = \left\| \frac{q}{a} \right\|_2^2 + \|r - \alpha W \beta\|_2^2$$

because q/a and $r - \alpha W \beta$ are orthogonal and thus the cross-terms contribute nothing to the norm. The last expression is a complete square whose first part is independent of W . Hence the minimizing parameter is $W = r/\alpha\beta$, and if it is indeed stable and admissible, it defines the unique optimal controller. Otherwise, no optimal controller exists.

The consequent minimum norm equals

$$\min_W \|H_c\|_2 = \left\| \frac{q}{a} \right\|_2.$$

Example 2 To illustrate, consider the plant $S(s) = 1/(s - 1)$. The class of all stabilizing controllers for S is found to be

$$R(s) = \frac{1 - (s - 1)W}{W}$$

for a free stable rational parameter $W \neq 0$. The complementary sensitivity transfer function is

$$H_c(s) = 1 - (s - 1)W.$$

Now $\alpha = s + 1$, $\beta = 1$ and the polynomial part of

$$\frac{\alpha\beta}{a} = \frac{s + 1}{s - 1} = 1 + \frac{2}{s - 1}$$

is $r = 1$. Thus H_c attains minimum H_2 norm for

$$W(s) = \frac{1}{s + 1}$$

and the corresponding optimal controller is $R(s) = 2$.

The optimal complementary sensitivity function is

$$H_c(s) = \frac{2}{s + 1}$$

and $\|H_c\|_2 = \sqrt{2}$.

Robust Stabilization

The notion of robust stability addresses stabilization of plants subject to modeling errors, when the actual plant may differ from the nominal model, using a fixed controller. The ultimate goal is to stabilize the actual plant. The actual plant is unknown, however, so the best one can do is to stabilize a large enough set of plants.

Thus the basis technique to model plant uncertainty is to model the plant as belonging to a set. Such a set can be either structured – for example, there is a finite number of uncertain parameters – or unstructured: the frequency response lies in a set in the complex plane for every frequency. The unstructured uncertainty model

is more important for several reasons. On the one hand, it is well suited to represent high-frequency modeling errors, which are generically present and caused by such effects as infinite-dimensional electromechanical resonance, transport delays, and diffusion processes. On the other hand, the unstructured model of uncertainty leads to a simple and useful design theory.

The unstructured set of plants is usually constructed as a neighborhood of the nominal plant, with the uncertainty represented by additive or multiplicative perturbations. The size of the neighborhood is measured by a suitable norm, most common being the H_∞ norm that is defined for any rational function G analytic on the imaginary axis as

$$\|G\|_\infty = \sup_\omega |G(j\omega)|.$$

Let us illustrate the design for *robust stability under unstructured norm-bounded multiplicative perturbations*. Consider a nominal plant with transfer function S and its neighborhood S_Δ defined by

$$S_\Delta := (1 + F\Delta)S$$

where F is a fixed stable rational function and Δ is a variable stable rational function such that $\|\Delta\|_\infty \leq 1$.

The idea behind this uncertainty model is that $F\Delta$ is the normalized plant perturbation away from 1:

$$\frac{S_\Delta}{S} - 1 = F\Delta.$$

Hence if $\|\Delta\|_\infty \leq 1$, then for all frequencies ω

$$\left| \frac{S_\Delta(j\omega)}{S(j\omega)} - 1 \right| = |F(j\omega)|$$

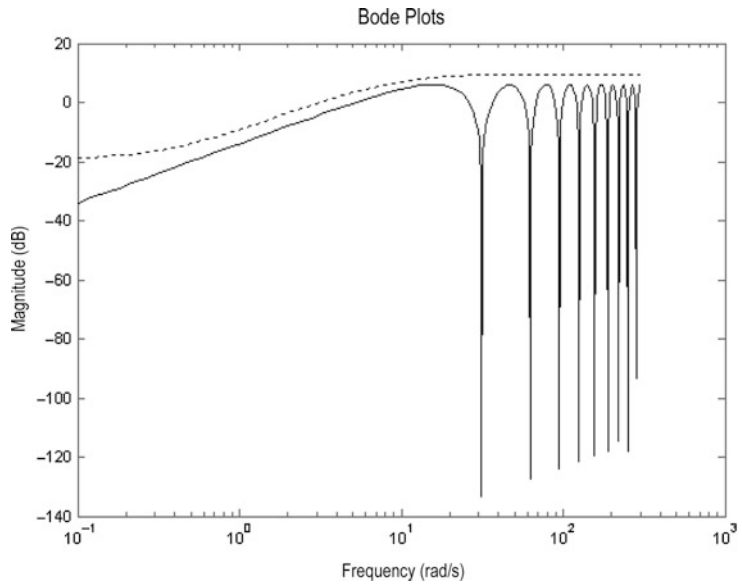
so that $|F(j\omega)|$ provides the uncertainty profile while Δ accounts for phase uncertainty.

Now suppose that R is a controller that stabilizes the nominal plant S . Applying the small gain theorem, R is seen to stabilize the entire family of plants S_Δ if and only if

$$\|H_c F\|_\infty < 1.$$

Polynomial/Algebraic Design Methods,

Fig. 2 Bode plots of F (dotted) and $e^{-0.2s} - 1$ (solid)



This is a necessary and sufficient condition for robust stabilization of the nominal plant S .

The set of all stabilizing controllers for $S = b/a$ is described by the formula

$$R = \frac{y - aW}{x + bW}$$

where $ax + by = 1$ and W is a free stable rational parameter. The robust stability condition then reads

$$\|b(y - aW)F\|_{\infty} < 1.$$

Any stable rational W that satisfies this inequality then defines a robustly stabilizing controller R for S . In case W actually minimizes the norm, one obtains the best robustly stabilizing controller.

Example 3 Consider a plant with the transfer function

$$S_{\tau}(s) = \frac{s+1}{s-1} e^{-\tau s}$$

where the time delay τ is known only to the extent that it lies in the interval $0 \leq \tau \leq 0.2$. The task is to find a controller that stabilizes the uncertain plant S_{τ} . The time-delay factor $e^{-\tau s}$ can be treated as a multiplicative perturbation of the nominal plant

$$S(s) = \frac{s+1}{s-1}$$

by embedding S_{τ} in the family

$$S_{\Delta} := (1 + F\Delta)S$$

where Δ ranges over the set of stable rational functions such that $\|\Delta\|_{\infty} \leq 1$. To do this, F should be chosen so that the normalized perturbation satisfies

$$\left| \frac{S_{\Delta}(j\omega)}{S(j\omega)} - 1 \right| = \left| e^{-j\omega\tau} - 1 \right| \leq |F(j\omega)|$$

for all ω and τ . A little time with the Bode magnitude plot shows that a suitable uncertainty profile is

$$F(s) = \frac{3s+1}{s+9}.$$

Figure 2 is the Bode magnitude plot of this F and $e^{-\tau s} - 1$ for $\tau = 0.2$, the worst value.

The task of stabilizing the uncertain plant S_{τ} is thus replaced by that of stabilizing every element in the set S_{Δ} , that is to say, by robustly stabilizing the nominal plant S with respect to the multiplicative perturbations defined by F .

The set of all stabilizing controllers for S is found to be

$$R(s) = \frac{0.5 - (s-1)W}{-0.5 + (s+1)W}$$

where $W \neq 0.5/(s+1)$ is any stable rational parameter. The robust stability condition reads

$$\|P - QW\|_{\infty} < 1$$

where

$$P(s) = 0.5(s+1) \frac{3s+1}{s+9},$$

$$Q(s) = (s-1)(s+1) \frac{3s+1}{s+9}.$$

Since Q has one unstable zero at $s = 1$, it follows from the maximum modulus theorem that the minimum of the H_{∞} norm taken over all stable rational functions W is $P(1) = 0.4 < 1$ and this minimum is achieved for

$$W(s) = \frac{P(s) - P(1)}{Q(s)} = \frac{1}{10} \frac{15s+31}{(s+1)(3s+1)}.$$

Thus, the robust stability condition is satisfied, and the corresponding best robustly stabilizing controller is

$$R(s) = \frac{2}{13} \frac{s+9}{s+1}.$$

Polynomial Equation Approach

In order to determine the set of all stabilizing controllers for a given plant, it is enough to determine one particular solution of the Bézout equation. It is therefore plausible that performance specifications in addition to stability can be met by selecting an appropriate solution of a polynomial equation that is related to the Bézout equation.

The reduction of controller synthesis to solving polynomial equations is referred to as the *polynomial equation approach* to control system design. The equations involved are Diophantine equations of the form

$$ap + bq = d$$

where a , b , and d are given polynomials and p , q are polynomials to be found. Such an equation is solvable for any d if and only if a and b are coprime polynomials. Then, the solution set is given by

$$p = p_0 + bt, \quad q = q_0 - at$$

where p_0 , q_0 is a particular solution and t is an arbitrary polynomial.

Such an equation is in fact the pole placement equation. Thus, pole placement is a prototype control problem.

Pole Placement

The requirement of stability places all closed-loop system poles within the left half-plane $\text{Re } s < 0$. Very often, however, we wish to allocate the poles to a specific region of the half-plane or to achieve specific pole positions.

Given a plant $S = b/a$, the set of all stabilizing controllers for S is

$$R = \frac{y - aW}{x + bW}$$

where x , y are polynomials such that $ax + by = 1$ and W is a free stable rational parameter. Let $W = w/d$ for a stable polynomial d . Then

$$R = \frac{dy - aw}{dx + bw} := \frac{q}{p}$$

and the closed-loop pole polynomial is given by

$$ap + bq = d(ax + by) = d.$$

Thus W parameterizes all stabilizing controllers for S , the denominator polynomial d of W specifies the positions of the control system poles, and the numerator polynomial w of W represents the remaining degrees of freedom, i.e., parameterizes all stabilizing controllers that assign the specified poles.

Example 4 Consider the plant $S(s) = 1/(s-1)$ and the set of stabilizing controllers for S :

$$R(s) = \frac{1 - (s-1)W}{W}, \quad W \neq 0.$$

Let the desired pole locations be given by the polynomial $d = s^2 + 2s + 1$. This is achieved by putting $W = w/d$ for an arbitrary numerator polynomial $w \neq 0$.

It is to be noted that d specifies the poles at *finite* positions only. Poles at $s = \infty$ will occur whenever R is not proper rational. To avoid this situation, w should be constrained to $w = s + \omega$ for any real ω . Then the set of controllers that achieve the desired pole placement is

$$R(s) = \frac{(3 - \omega)s + (1 + \omega)}{s + \omega}.$$

Alternatively, one can solve the pole placement equation $ap + bq = d$ directly. The solution set is

$$p = t, \quad q = s^2 + 2s + 1 - (s-1)t$$

and q/p is proper if and only if $t = s + \omega$, ω real.

H₂ Optimal Control

The H_2 optimal control is a special case of pole placement. Indeed, the optimal controller is given by

$$R = \frac{y - aW}{x + bW} = \frac{y - a\frac{r}{\alpha\beta}}{x + b\frac{r}{\alpha\beta}} = \frac{\alpha y\beta - ar}{\alpha x\beta + br} := \frac{q}{p}$$

and

$$\begin{aligned} ap + bq &= a(\alpha x\beta + br) + b(\alpha y\beta - ar) \\ &= \alpha\beta(ax + by) = \alpha\beta. \end{aligned}$$

Thus, the (finite) pole positions of the H_2 optimal control system are given by the pole polynomial $d = \alpha\beta$. The system has no poles at $s = \infty$ as the optimal complementary sensitivity function H_c is strictly proper.

The pole placement equation, however, has more than one solution. Which one is optimal? The one with q/a is strictly proper. It is the solution pair p, q with q having a least degree.

Example 5 Let us reconsider Example 2. As an alternative, one can solve the Diophantine equation

$$(s-1)p + q = s + 1$$

for the solution pair p, q such that $q/(s-1)$ is strictly proper. This yields the least-degree solution pair with respect to q , namely, $p = 1$, $q = 2$. The optimal controller is $R = q/p = 2$.

Summary and Future Directions

The benefits of representing stabilizing controllers by a single parameter include (1) easy accommodation of additional design specifications by selecting an appropriate parameter, (2) all transfer functions in a stabilized system are linear in the parameter (while they are nonlinear in the controller), and (3) the parameter belongs to a smaller set of stable rational functions (while the controller is any rational).

The results presented here for linear time-invariant systems with rational transfer functions can be generalized to extend the scope of the theory to include distributed-parameter systems, time-varying systems, and even nonlinear systems.

The transfer functions of distributed-parameter systems are no longer rational, and coprime factorizations cannot be assumed a priori to exist. The coefficients of time-varying systems are functions of time, and the operations of multiplication and differentiation do not commute. In nonlinear systems, transfer functions are replaced by input-output maps. Technical assumptions may prevent one from parameterizing the *entire* set of internally stabilizing controllers; still, the subset may be large enough for practical purposes. For many systems of physical and engineering interest, the above difficulties can be circumvented and the algebraic/polynomial approach carries over with suitable modifications.

Cross-References

- [Control of Linear Systems with Delays](#)
- [Feedback Stabilization of Nonlinear Systems](#)

- [H-Infinity Control](#)
- [H₂ Optimal Control](#)
- [Linear State Feedback](#)
- [Robust Synthesis and Robustness Analysis Techniques and Tools](#)
- [Spectral Factorization](#)
- [Tracking and Regulation in Linear Systems](#)

Recommended Reading

The use of polynomials, in one way or another, in feedback control system design can be traced back to Newton et al. (1957) and Jury (1958). The authors noted that for a closed-loop system to be stable, H_c must absorb the plant unstable zeros. The plant was assumed to be stable; if this assumption were dropped, H_s would have been found to absorb the plant unstable poles. These conditions are equivalent to polynomial divisibility conditions and hence to the Bézout stability equation, which appears later in Kučera (1974).

The first attempt to use polynomials in an explicit manner is due to Volgin (1962), a student of Tsytkin. He obtained a solution of the pole placement problem through the solution of a polynomial equation, known as the pole placement equation. Åström (1970) published a polynomial equation solution to the minimum variance control problem for minimum-phase plants. The ultimate publication that presents the polynomial equation approach to multi-input multi-output control system design is Kučera (1979).

The underlying problem in any control system design is that of stability. It is logical to design the control system step by step: stabilization first and then the additional performance specifications. To do this, we need to know any and all stabilizing controllers for the given plant.

This problem was first addressed and solved for finite-dimensional, linear time-invariant systems using transfer function methods; see Larin et al. (1971), Kučera (1975), Youla et al. (1976a, b), and Kučera (1979). A state-space representation of all stabilizing controllers was derived later by Nett et al. (1984).

It took decades to appreciate the importance of the result and come up with applications. The

milestones were the observations by Desoer et al. (1980) that the polynomial fraction approach can be extended to linear systems with nonrational transfer functions, as well as the result by Hammer (1985) showing that the approach is applicable to a broad class of nonlinear systems. Further generalizations were obtained by Paice and Moore (1990), Anderson (1998), and Quadrat (2003, 2006).

The parameterization of all controllers that stabilize a given plant was labeled the Youla-Kučera parameterization in Anderson (1998). This result launched an entirely new area of research and has ultimately become a new paradigm for control system design.

Tutorial textbooks on this subject include Vidyasagar (1985), Doyle et al. (1992), and Kučera (2003, 2011). The reader is further referred to the survey papers by Kučera (1993), Anderson (1998), and Kučera (2007).

Advanced and recent applications of the Youla-Kučera parameterization include stabilization under constrained inputs (Henrion et al. 2001), robust stabilization with fixed-order controllers (Henrion et al. 2003), accommodation of time-domain constraints on inputs and outputs (Henrion et al. 2005a), and determination of least-order stabilizing controllers (Henrion et al. 2005b).

Bibliography

- Anderson BDO (1998) From Youla-Kučera to identification, adaptive and nonlinear control. *Automatica* 34:1485–1506
- Åström KJ (1970) Introduction to stochastic control theory. Academic, New York
- Desoer CA, Liu RW, Murray J, Saeks R (1980) Feedback system design: the fractional representation approach to analysis and synthesis. *IEEE Trans Autom Control* 25:399–412
- Doyle JC, Francis BA, Tannenbaum AR (1992) Feedback control theory. Macmillan, New York
- Hammer J (1985) Nonlinear system stabilization and coprimeness. *Int J Control* 44:1349–1381
- Henrion D, Tarbouriech S, Kučera V (2001) Control of linear systems subject to input constraints: a polynomial approach. *Automatica* 37:597–604
- Henrion D, Šebek M, Kučera V (2003) Positive polynomials and robust stabilization with fixed-order controllers. *IEEE Trans Autom Control* 48:1178–1186

- Henrion D, Tarbouriech S, Kučera V (2005a) Control of linear systems subject to time-domain constraints with polynomial pole placement and LMIs. *IEEE Trans Autom Control* 50:1360–1364
- Henrion D, Kučera V, Molina-Cristobal A (2005b) Optimizing simultaneously over the numerator and denominator polynomials in the Youla-Kučera parametrization. *IEEE Trans Autom Control* 50:1369–1374
- Jury EI (1958) *Sampled-data control systems*. Wiley, New York
- Kučera V (1974) Closed-loop stability of discrete linear single variable systems. *Kybernetika* 10: 146–171
- Kučera V (1975) Stability of discrete linear feedback systems. In: *Proceedings of the 6th IFAC world congress*, Boston, vol 1, pp 44.1
- Kučera V (1979) *Discrete linear control: the polynomial equation approach*. Wiley, Chichester
- Kučera V (1993) Diophantine equations in control – a survey. *Automatica* 29:1361–1375
- Kučera V (2007) Polynomial control: past, present, and future. *Int J Robust Nonlinear* 17:682–705
- Kučera V (2003) Parametrization of stabilizing controllers with applications. In: Voicu M (ed) *Advances in automatic control*. Kluwer, Boston, pp 173–192
- Kučera V (2011) Algebraic design methods. In: Levine WS (ed) *The control handbook: control system advanced methods*, 2nd edn. CRC, Boca Raton
- Larin VB, Naumenko KI, Suntsev VN (1971) Spectral methods for synthesis of linear systems with feedback (in Russian). *Naukova Dumka*, Kiev
- Nett CN, Jacobson CA, Balas MJ (1984) A connection between state-space and doubly coprime fractional representations. *IEEE Trans Automat Control* 29:831–832
- Newton G, Gould L, Kaiser JF (1957) *Analytic design of linear feedback controls*. Wiley, New York
- Paice ADB, Moore JB (1990) On the Youla-Kučera parametrization of nonlinear systems. *Syst Control Lett* 14:121–129
- Quadrat A (2003) On a generalization of the Youla-Kučera parametrization. Part I: the fractional ideal approach to SISO systems. *Syst Control Lett* 50:135–148
- Quadrat A (2006) On a generalization of the Youla-Kučera parametrization. Part II: the lattice approach to MIMO systems. *Math Control Signal* 18:199–235
- Vidyasagar M (1985) *Control system synthesis: a factorization approach*. MIT, Cambridge
- Volgin LN (1962) *The fundamentals of the theory of controlling machines* (in Russian). Soviet Radio, Moscow
- Youla DC, Bongiorno JJ, Jabr HA (1976a) Modern Wiener-Hopf design of optimal controllers, part I: the single-input case. *IEEE Trans Autom Control* 21:3–14
- Youla DC, Jabr HA, Bongiorno JJ (1976b) Modern Wiener-Hopf design of optimal controllers, part II: the multivariable case. *IEEE Trans Autom Control* 21: 319–338

Port-Hamiltonian Systems: From Modeling to Control

Arjan van der Schaft

Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, Jan C. Willems Center for Systems and Control, University of Groningen, Groningen, The Netherlands

Abstract

Port-based modeling of general multi-physics systems leads to port-Hamiltonian system formulations, which make explicit the underlying network and energetic structure. Key properties are (shifted) passivity, compositionality, and existence of conserved quantities. This provides powerful tools for simulation, analysis, and control.

Keywords

Multi-physics systems · Network modeling · Passivity · Compositionality · Interconnection · Nonlinear control

Introduction

Port-Hamiltonian systems theory is rooted in the *port-based modeling* approach to complex multi-physics systems (Paynter 1961), viewing the system as the interconnection of ideal energy storing, energy dissipating, and energy routing elements, via pairs of conjugate variables whose product equals power. It brings together classical Hamiltonian dynamics (originating mostly from mechanics) with network dynamics (motivated in particular by electrical network theory). Thus identifying the underlying physical structure of the (nonlinear) system leads to analysis and control strategies that exploit the nonlinearities and are insightful and often robust.

Port-Hamiltonian Systems

A *port-Hamiltonian system* (briefly, *pH system*), with n -dimensional state space $\mathcal{X}$, input and output spaces $U = Y = \mathbb{R}^m$, and Hamiltonian $H : \mathcal{X} \rightarrow \mathbb{R}$, is defined as (Maschke and van der Schaft 1992; van der Schaft 2017; Ortega et al. 2002; van der Schaft and Jeltsema 2014)

$$\begin{aligned}\dot{x} &= J(x)e - \mathcal{R}(x, e) + G(x)u, \quad e = \nabla H(x) \\ y &= G^T(x)e,\end{aligned}\quad (1)$$

where the $n \times n$ matrix $J(x)$ satisfies $J(x) = -J^T(x)$ for all x , and the mapping $\mathcal{R}$ is such that $e^T \mathcal{R}(x, e) \geq 0$ for all x, e . Here $e = \nabla H(x)$ denotes the n -dimensional column vector of partial derivatives of H . The function H represents the *stored energy* of the system, the matrix J its internal *interconnection structure*, and $\mathcal{R}$ its *energy dissipation structure*. In the case of *linear* dissipation (1) reduces to

$$\begin{aligned}\dot{x} &= [J(x) - R(x)] \nabla H(x) + G(x)u \\ y &= G^T(x) \nabla H(x)\end{aligned}\quad (2)$$

where the $n \times n$ matrix $R(x)$ satisfies $R(x) = R^T(x) \geq 0$ for all x .

The definition of port-Hamiltonian system can be further extended (van der Schaft 2017; van der Schaft and Jeltsema 2014) to include direct feedthrough terms and to mixtures of differential and algebraic equations (DAE systems), as often arise in network modeling of large-scale physical systems. The definition also extends to distributed-parameter physical systems described by partial differential equations; see, e.g., van der Schaft and Maschke (2002).

Passivity and Shifted Passivity

By the properties of $J(x)$ and $\mathcal{R}$, it immediately follows that any pH system satisfies

$$\begin{aligned}\frac{dH}{dt}(x(t)) &= \nabla^T H(x(t)) \dot{x}(t) \\ &= -e^T(t) \mathcal{R}(x(t), e(t)) + y^T(t) u(t) \\ &\leq y^T(t) u(t),\end{aligned}\quad (3)$$

implying *passivity* if H is *bounded from below* (and *cyclo-passivity* for general H). Conversely, it can be shown that most (cyclo-)passive nonlinear systems can be represented as port-Hamiltonian systems for *some* $J(x)$ and $\mathcal{R}(x, e)$. On the other hand, in port-Hamiltonian systems modeling H , J , G , and $\mathcal{R}$, all arise from first principles physical modelling and so have a direct physical interpretation (Duindam et al. 2009; van der Schaft 2017).

In the case of *linear* dissipation, *constant* J, R, G , and *convex* H , port-Hamiltonian systems (2) are also passive with respect to *non-zero* equilibria. Indeed, let $\bar{u}$ be any constant input such that $0 = [J - R] \nabla H(\bar{x}) + G\bar{u}$ for some $\bar{x}$. Then, denoting $\bar{y} = G^T \nabla H(\bar{x})$, (2) can be rewritten as the *shifted* pH system

$$\begin{aligned}\dot{x} &= [J - R] \nabla \hat{H}_{\bar{x}}(x) + G(u - \bar{u}) \\ y - \bar{y} &= G^T \nabla \hat{H}_{\bar{x}}(x),\end{aligned}\quad (4)$$

where the *shifted Hamiltonian* is defined as $\hat{H}_{\bar{x}}(x) = H(x) - \nabla H(\bar{x})(x - \bar{x}) - H(\bar{x})$ (Bregman divergence). By convexity of H it follows that $\hat{H}_{\bar{x}} \geq 0$, thus showing passivity with respect to the shifted inputs and outputs $u - \bar{u}, y - \bar{y}$ (van der Schaft 2017).

Stability Analysis by Energy-Casimir Method

The dissipation inequality (3) implies that H , provided it has a strict minimum at x_0 , can be used as a candidate Lyapunov function for an equilibrium x_0 of the port-Hamiltonian system (1) for $u = 0$. The same applies to non-zero equilibria $\bar{x}$ corresponding to non-zero constant $\bar{u}$ using the shifted Hamiltonian $\hat{H}_{\bar{x}}$. An additional degree of freedom in shaping H (or $\hat{H}_{\bar{x}}$) to a suitable Lyapunov function is offered by the existence of *Casimirs*. These are mappings $C : \mathcal{X} \rightarrow \mathbb{R}^k$ satisfying (in the case of linear dissipation) $\nabla^T C(x) J(x) = 0 = \nabla^T C(x) R(x)$ and thus are *conserved quantities* for the dynamics (2) for $u = 0$. Any $V := H + \Phi(C)$ for $\Phi : \mathbb{R}^k \rightarrow \mathbb{R}$ then satisfies $\frac{d}{dt} V \leq 0$.

Control by Interconnection

A fundamental property of port-Hamiltonian systems is compositionality: power-conserving interconnections of port-Hamiltonian systems are

again port-Hamiltonian. The simplest example is the standard negative feedback interconnection $u_1 = -y_2 + e_1, u_2 = y_1 + e_2$ of two port-Hamiltonian systems (2), indexed by 1, 2, leading to the *closed-loop port-Hamiltonian system*

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} J_1(x_1) - R_1(x_1) - G_1(x_1)G_2^T(x_2) \\ G_2(x_2)G_1^T(x_1) & J_2(x_2) - R_2(x_2) \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial x_1}(x_1, x_2) \\ \frac{\partial H}{\partial x_2}(x_1, x_2) \end{bmatrix} + \begin{bmatrix} G_1(x_1) & 0 \\ 0 & G_2(x_2) \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} G_1^T(x_1) & 0 \\ 0 & G_2^T(x_2) \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial x_1}(x_1, x_2) \\ \frac{\partial H}{\partial x_2}(x_1, x_2) \end{bmatrix}, \quad H(x_1, x_2) := H_1(x_1) + H_2(x_2) \end{aligned} \quad (5)$$

with inputs e_1, e_2 and outputs y_1, y_2 . Interpreting the pH system indexed by 1 as the *plant system*, and the pH system indexed by 2 as the *controller system*, this leads to the following strategy for, e.g., *stabilization* of a prescribed set-point x_1^* . Design J_2, R_2, G_2 in such a way that one obtains a Casimir $C(x_1, x_2)$ for the closed-loop system, such that for a certain choice of the controller Hamiltonian H_2 and mapping Φ , the function

$$V(x_1, x_2) := H_1(x_1) + \Phi(H_2(x_2), C(x_1, x_2)) \quad (6)$$

has a strict minimum at (x_1^*, x_2^*) for some x_2^* , thus serving as Lyapunov function. Preservation of Casimirs of the special form $C(x_1, x_2) = x_2 - F(x_1)$ implies $x_2 = F(x_1)$ modulo a constant, leading to Lyapunov functions $\tilde{V}(x_1) = H_1(x_1) + \tilde{\Phi}(H_2(F(x_1)))$ in terms of x_1 only, and thus to a *state feedback* interpretation (Ortega et al. 2002; van der Schaft 2017). This can be generalized to *Interconnection-Damping Assignment Passivity-Based Control* (Ortega et al. 2002; Duindam et al. 2009; van der Schaft 2017).

Summary and Future Directions

Port-based modeling of multi-physics systems leads to port-Hamiltonian formulations, bridging the gap between first principles modeling and control. If system parameters or system components are (partly) unknown, this calls

for the development of port-Hamiltonian identification and realization theory. Large-scale systems or spatial discretization of distributed-parameter systems yield high-dimensional port-Hamiltonian system formulations, necessitating structure-preserving model order reduction.

Cross-References

- [Differential Geometric Methods in Nonlinear Control](#)
- [Feedback Stabilization of Nonlinear Systems](#)
- [Passivity-Based Control](#)
- [Robot Motion Control](#)

Recommended Reading

van der Schaft AJ (2017) *L₂-gain and passivity techniques in nonlinear control*, Chapters 6 and 7
 van der Schaft AJ, Jeltsema D (2014) *Port-Hamiltonian systems theory: and introductory overview*

Duindam et al. (2009) *Modeling and control of complex physical systems; the port-Hamiltonian approach*

Bibliography

Duindam V, Macchelli A, Stramigioli S, Bruyninckx H (eds) (2009) *Modeling and control of complex physical systems; the port-Hamiltonian approach*. Springer, Berlin/Heidelberg

- Maschke B, van der Schaft AJ (1992) Port-controlled Hamiltonian systems: modelling origins and system theoretic properties. In: Proceedings of 2nd IFAC Symposium on Nonlinear Control Systems, Fliess M (ed), Bordeaux
- Ortega R, van der Schaft AJ, Maschke BM, Escobar G (2002) Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems. *Automatica* 38:585–596
- Paynter HM (1961) Analysis and design of engineering systems. MIT Press, Cambridge
- van der Schaft AJ (2017) L_2 -gain and passivity techniques in nonlinear control, 3rd edn. Springer-International AG, Cham
- van der Schaft AJ, Maschke BM (2002) Hamiltonian formulation of distributed-parameter systems with boundary energy flow. *J Geom Phys* 42:166–194
- van der Schaft AJ, Jeltsema D (2014) Port-Hamiltonian systems theory: an introductory overview. *Found Trends Syst Control* 1(2–3):173–378

Power System Voltage Stability

Costas Vournas

School of Electrical and Computer Engineering,
National Technical University of Athens,
Zografou, Greece

Abstract

Voltage stability of electric power systems is a challenging topic both theoretically and in practice. This entry touches briefly on the main aspects of the problem and highlights theoretical foundations and fundamental methods for voltage stability analysis. The single-load radial system is used to introduce basic concepts, such as the PV curve, and the instability mechanism, while the implications for a meshed, multiple-load system are briefly outlined. Some applications to practical problems are briefly enumerated.

Keywords

Maximum power transfer · Active and reactive power · Load dynamics · Load tap changers (LTC) · Stability conditions · PV curve

Introduction

Voltage stability is related to the maximum power transfer in an AC (alternating current) network. In normal conditions, system load demand should never come close to this limit. As, however, electricity demand started swelling after the 1970s with an increasingly faster pace, transmission network investments could not follow closely enough. Investment cost in transmission is usually high and difficulties with environmental constraints and “not in my back yard” mentality of local communities did not make transmission network expansion any easier. Power systems are thus relying for their continuing operation more and more on (reactive power) compensation and automatic controls to maintain sufficient transmission capacity of relatively weakening networks.

As a result, several instances of voltage instability started to appear in several industrialized countries after the 1980s (Taylor 1994) leading to smaller or larger area blackouts, much to the surprise of the power engineering community that was not prepared to deal with this type of events, in which a usual and expected phase of gradual voltage decline suddenly precipitates to an uncontrollable voltage drop leading to partial or total blackout after a succession of equipment disconnection by protection devices.

In power system engineering practice, voltage drops following load ramping or sudden events, such as equipment loss (line, generator switching, etc.), usually referred to in power engineering literature as contingencies, are calculated by solving a set of nonlinear algebraic equations known as the power flow problem. As these are “steady-state” equations, the dynamic aspect leading to an accelerating, cascading failure is not obvious. One should notice however in the above account the key word “nonlinear”: nonlinear equations at the maximum power transfer limit no longer have a solution. This was and is one of the keys in understanding the voltage stability problem. To take it one step further, close to the loss of solution (loss of equilibrium), a set of dormant (up to this point) dynamics become dominant

leading the system to instability. The following sections will explain these notions.

Single Generator-Load (Radial) System

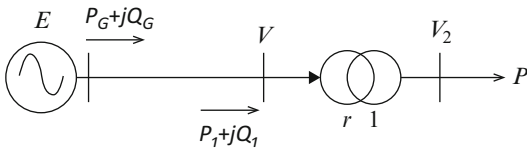
Maximum Power Transfer

In any electrical network (DC or AC), there is a maximum power that can be transferred between any two nodes. In a two-node radial system, the maximum power transfer coincides with the well-known impedance matching conditions. For a radial AC system, when the load is restricted to a constant power factor, the impedance matching condition is that the source (network) impedance is equal in magnitude to the load impedance.

Consider the radial system of Fig. 1. In this system we assume that the load active power P (and possibly reactive power Q) is fed through a transformer with adjustable tap ratio r (in per unit). The tap is automatically adjusted by a load tap changer (LTC) so as to keep the secondary voltage V_2 within a deadband. We will consider throughout that the LTC is a part of the load.

The simplest case for this radial system is when both the line and transformer are lossless ($P_G = P_1 = P$) and the load is kept to unity power factor ($Q = 0$). The generator is assumed as a constant voltage source E . If we further assume that the transformer leakage reactance is negligible ($Q_1 = Q = 0$ in Fig. 1), the maximum power transfer in this simple case is encountered when the load impedance, as seen from the primary (r^2/G), is equal to the line reactance:

$$X = r^2/G \quad (1)$$



Power System Voltage Stability, Fig. 1 Single load radial system

where the load conductance $G = P/V_2^2$. It can be readily shown that the maximum power in this case is $P_{\max} = E^2/2X$. Note that this is a static condition that is not related to how the load varies with the voltage V_2 .

The most popular way of visualizing the maximum power condition is through the PV curve of Fig. 2, in which the consumed (transferred) power P is plotted versus the primary (transmission) side voltage V .

In Fig. 2 the nose-shaped solid line is the *network characteristic* corresponding to all possible solution of the network equations for a given P (or V). The maximum power transfer is easily identified as the tip of the curve (point C). Note that PV curves can be plotted for any load power factor and line resistance.

Load Dynamics and Voltage Stability

As stated above, maximum power transfer is a static condition based on network equations only. To identify its relation to voltage stability some form of load dynamics must be introduced. Load dynamics are generally changing the load characteristics so as to adjust load power *consumption* P to a given load *demand* P_o . As a disturbance usually reduces voltage (and thus consumption of a voltage-sensitive load), load dynamics tend to restore the consumption to the pre-disturbance demand.

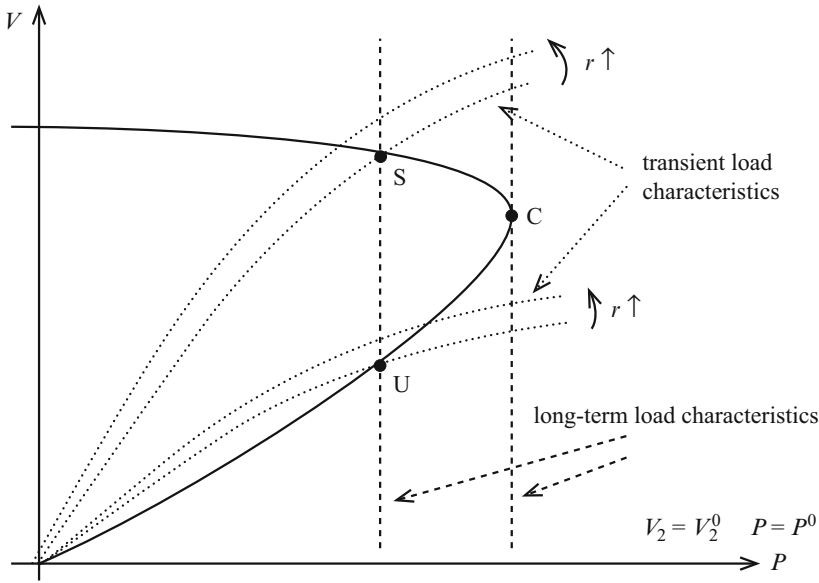
Load restoration can be continuous, for instance, represented by a time-varying conductance following the ODE:

$$T \dot{G} = \frac{P_o}{V_o^2} - \frac{P}{V_o^2} = \frac{P_o}{V_o^2} - G \left(\frac{V_2}{V_o} \right)^2 \quad (2)$$

Clearly in this case the stability condition is that the consumption $P = GV_2^2$ increases with the increase of the load conductance G :

$$\frac{\partial P}{\partial G} > 0 \quad (3)$$

It is easily verified from Fig. 2 that this condition is met only in the upper part of the PV curve before point C, whereas in the lower part, after point C, increased conductance results in lower



Power System Voltage Stability, Fig. 2 PV curve of radial system

consumption violating (3). Clearly at C (3) holds as an equality.

Assuming the load on the secondary side to be a constant admittance, we can distinguish two types of load characteristics in Fig. 2: the *transient (short-term) load characteristic* shown with dotted lines corresponds to a specific transformer tap ratio r , whereas the *long-term load characteristic* corresponds to equilibrium conditions where V_2 is within the deadband and approximately equal to V_o and is shown with dashed lines for different load demands.

Load dynamics can also be discrete, e.g., driven by the tap changing transformer of Fig. 1. As the LTC is trying to restore the secondary voltage, it will reduce r when $V_2 < V_o - d$ and will increase r when $V_2 > V_o + d$, where d is one half of the deadband.

The effect of tap ratio increase in the upper and lower part of the PV curve is shown in Fig. 2 (points S and U). In the upper part, increased r will reduce consumption which implies that V_2 is also reduced as expected. In the lower part (point U), increased r will increase consumption indicating an increased V_2 and thus an unstable LTC operation. The stability condition in this case is:

$$\frac{\partial V_2}{\partial r} < 0 \quad (4)$$

Clearly for either discrete or continuous dynamics, at the maximum power point C, a stable and an unstable equilibrium branch come together, leaving no equilibrium points for higher demand. In bifurcation theory, this point is known as a saddle-node bifurcation (SNB).

Effect of Generation

The generator behind the constant voltage source E of Fig. 1 supplies the active power consumed by the load (it would cover also active losses, if present). In practice this means that it requires a governor with PI (proportional plus integral) control, as is customary for autonomous systems and a prime mover of the required capacity. The generator also maintains the constant voltage E assumed in the calculations. This requires an automatic voltage regulator (AVR), which can be assumed in this simple example as being also of PI type. The AVR is adjusting the DC rotor (field) current of the synchronous generator so as to maintain the terminal voltage constant.

For a given load, the active and reactive generation $P_G + jQ_G$ required is directly calculated from the network equations. The

electromotive force (EMF) corresponding to the field current can then be determined using standard synchronous machine equations and preferably taking into account the saturation of the machine iron core (Van Cutsem and Vournas 1998). It should be noted that due to thermal constraints, it is not possible to exceed a maximum rotor current in continuous operation. This results in a rotor current limit that is enforced by the generator overexcitation limiter (OEL). If loading conditions are such that the OEL is activated, the generator terminal voltage E cannot be maintained constant, and thus the voltage source E has to be replaced by a constant EMF in series with the generator reactance. This leads to a much more restrictive limit for the maximum power transfer.

In power flow calculations, the generator excitation limit is usually represented by a maximum allowable reactive generation $Q_G^{\max}$. When this limit is reached, the reactive generation remains constant and thus the terminal voltage is allowed to drop, i.e., the generator becomes a PQ bus. Note however that $Q_G^{\max}$ of an actual generator is not constant, but depends on terminal voltage and on active generation.

In any case the overexcitation limit of synchronous generators, and the resulting limitation of the reactive support they offer, are an important factor determining maximum power and thus voltage stability limits. In practice, voltage instability is reached only after some critical generators have reached the overexcitation limit.

Voltage Instability Mechanism

Following the preceding discussion, it is possible to describe the mechanism of voltage instability as follows (Van Cutsem and Vournas 1998):

Voltage instability stems from the attempt of load dynamics to restore power consumption beyond the capability of the combined transmission and generation system.

A voltage instability incident can occur either through a gradual load increase up to the maximum power limit, or most commonly following a contingency (or a cascade of contingencies)

drastically reducing the maximum power transfer below the pre-contingency demand. Thus any attempt at restoring power to the pre-contingency demand will induce an unstable response leading to voltage collapse.

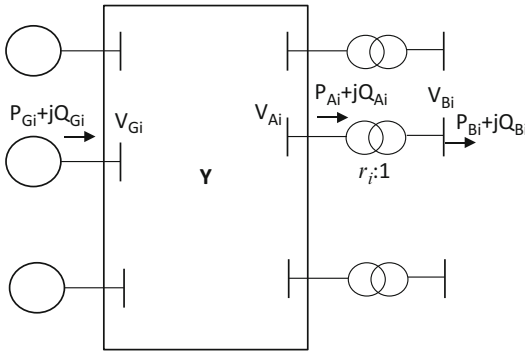
As the load dynamics are the driving force of voltage instability, the time-scale of load restoration is the one characterizing voltage stability. Thus, fast recovering loads, such as induction motors and power electronic-driven devices, tend to restore load in a second or less and constitute what is known in power system dynamic analysis as the short-term time scale (Kundur 2004). Study of relevant problems (motor stalling, delayed post-fault voltage recovery, etc.) is part of *short-term voltage stability* analysis.

In a slower time scale of several seconds up to minutes, load recovery dynamics include the LTCs and thermostatically controlled loads. This is the time-scale of *long-term voltage stability* analysis (Kundur 2004). Note that for long-term voltage stability the short-term dynamics such as those of motors and generators are considered to be at equilibrium. In mathematical system representation, this assumption leads to the replacement of short-term differential equations with algebraic equilibrium equations. This assumption is known as the quasi-steady-state (QSS) approximation.

Multiple-Load (Meshed) System

The single-load system of Fig. 1 serves well in defining the voltage stability problem and helps visualize its significance through the PV curve representation and the maximum loading, or critical point C. In actual power systems, however, there are multiple loads defining a multidimensional space where it is sometimes risky to apply the simple concepts of Fig. 1. For instance, it is important to distinguish between the supply system, which can be represented by a Thevenin equivalent, and the consumption part where loads affect each other and cannot be examined individually, one at a time.

Consider the power system of Fig. 3, where multiple generators are feeding a number of loads



Power System Voltage Stability, Fig. 3 Meshed power system

through a meshed network represented by the complex admittance matrix $\mathbf{Y}$. The steady-state conditions of the system including generation and load are traditionally represented by the power flow equations:

$$\begin{aligned} (P_{Gi} + jQ_{Gi}) - (P_{Ai} + jQ_{Ai}) \\ - \hat{V}_i \sum_{j=1}^N Y_{ij}^* \hat{V}_j^* = 0 \\ i = 1, \dots, N \end{aligned} \quad (5)$$

Using real variables, (5) can be written as:

$$\mathbf{g}(\mathbf{x}, \mathbf{p}) = 0 \quad (6)$$

where $\mathbf{p}$ is the vector of independent parameters (load demands, generator setpoints) and $\mathbf{x}$ the vector of dependent variables (voltage magnitudes and angles, or real and imaginary parts).

Note that in this representation, the load is referred to the primary side of LTC transformers as in Fig. 1. This can be considered constant at long-term equilibrium assuming secondary (distribution side) voltage restoration at its setpoint value $V_{Bi} = V_{oi}$.

Concerning generators the active power P_{Gi} should not be treated as constant when load is varying, so there has to be a participation factor attached to each generator that will represent primary (or secondary) frequency regulation characteristic (Van Cutsem and Vournas 1998). This

is sometimes referred to as *distributed slack bus* approach. For reactive power, the limits $Q_{Gi}^{\max}$ of reactive support should be set, beyond which the generator voltage is no longer constant (switch from PV to PQ bus).

The solution of the N nonlinear complex equations (5) for given load demand determines all complex voltages in the system. As in the simple radial system case, there may exist multiple solutions, some of which are unstable, or no solutions at all. The stability limits, where (3) and (4) hold as equalities for the radial system, are now given by the singularity of the Jacobian of the equilibrium conditions (6):

$$\det \mathbf{D}_{\mathbf{x}} \mathbf{g} = 0 \quad (7)$$

The stability limit can be determined also by the singularity of the state matrix (Van Cutsem and Vournas 1998; Medanic 1987):

$$\det \mathbf{A} = \det \left[\frac{\partial V_i}{\partial r_j} \right] = 0 \quad (8)$$

Note that the impedance matching condition for a single load amounts to the diagonal element of the state matrix becoming zero ($a_{ii} = 0$), which is much more strict than the singularity condition (8) that marks the actual onset of instability. The points satisfying (7) or (8) are critical points and form a multidimensional manifold in parameter space called *bifurcation surface*.

Applications

The above analysis briefly touches on fundamentals. Detailed analysis tools for Voltage Stability include (but are not limited to) continuation power flow, optimal power flow, VQ curves, time simulation (short-term, long-term, QSS), sensitivity and eigenvalue/singular value analysis. Voltage Security Analysis is presently applied on-line in various control centers based on the above methods of analysis for a large number of contingencies. Countermeasures to voltage instability and collapse cover a wide spectrum, from automatic reactive devices switching to special pro-

tection controls and load shedding as a last resort. Further details can be sought in textbooks Taylor (1994) and Van Cutsem and Vournas (1998).

Cross-References

- [Modeling of Dynamic Systems from First Principles](#)
- [Modeling Hybrid Systems](#)
- [Power System Voltage Stability](#)
- [Time-Scale Separation in Power System Swing Dynamics: Singular Perturbations and Coherency](#)

Bibliography

- Kundur et al. P (2004) Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions. IEEE Trans Power Syst 19:1387–1401
- Medanic J, Ilic-Spong M, Christensen J (1987) Discrete models of slow voltage dynamics for under load tap-changing transformer coordination. IEEE Trans Power Syst 2:873–882
- Taylor CW (1994) Power system voltage stability. EPRI power system engineering series. Mc Graw-Hill, New York
- Van Cutsem T, Vournas C (1998) Voltage stability of electric power systems. Kluwer Academic Publishers, Boston (Springer, 2008)

Powertrain Control for Hybrid-Electric and Electric Vehicles

Giorgio Rizzoni

Department of Mechanical and Aerospace Engineering, Center for Automotive Research, The Ohio State University, Columbus, OH, USA

Abstract

Powertrain electrification and hybridization have rapidly become part of the portfolio of all major automotive manufacturers, ranging from hybrid-electric, to plug-in hybrid-electric, to battery-electric vehicles,

to hybrid-hydraulic and hybrid-mechanical solutions. The increased complexity of the powertrain systems associated with hybrid vehicles presents interesting control challenges and problems, and this entry describes the more common architectures of hybrid-electric vehicle powertrains and their operation, focusing on the important problem of optimal control for energy management of hybrid-electric vehicles, on mode switching, and on battery management. In the conclusion, a connection is made between these problems and their interaction with intelligent transportation systems.

Keywords

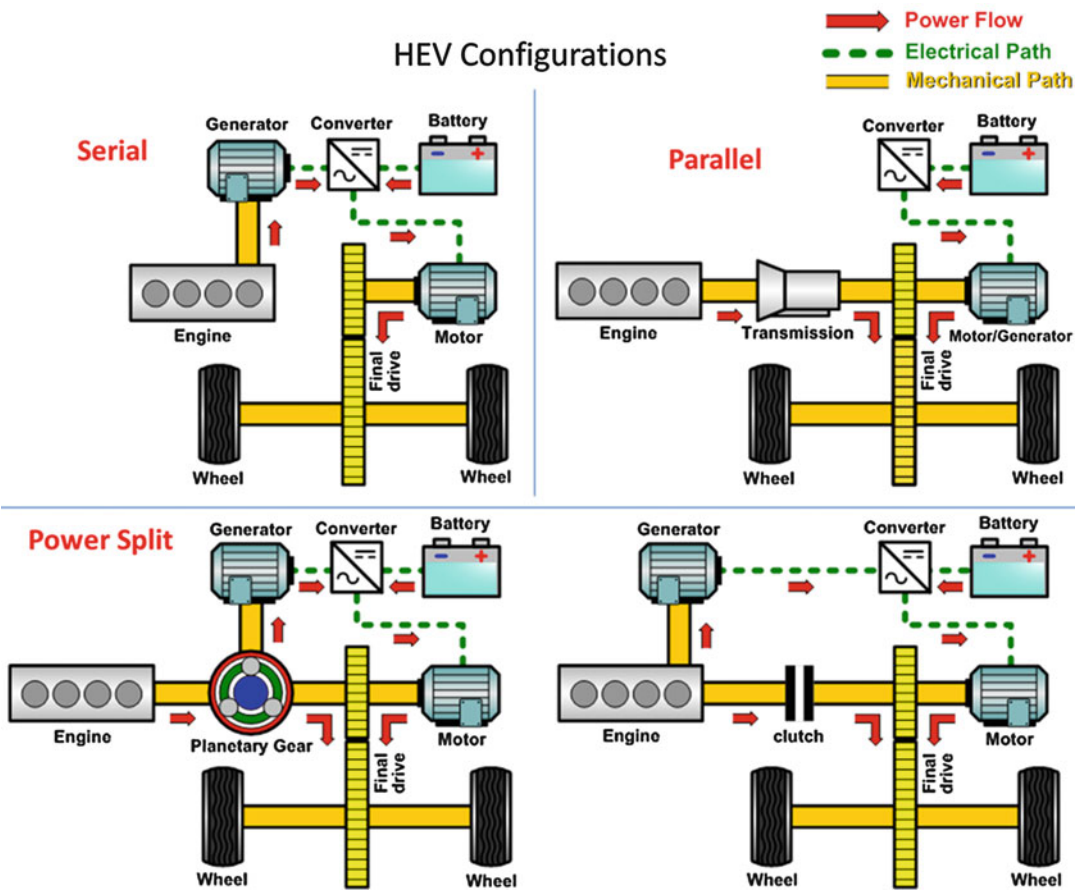
Battery management · Intelligent transportation systems · Vehicle-grid interaction

Introduction

Increasingly stringent fuel economy and emissions regulations have required the automotive industry to consider more fuel-efficient powertrains and alternative primary sources of transportation fuels. Powertrain electrification and hybridization have rapidly become part of the portfolio of all major automotive manufacturers, ranging from hybrid-electric, to plug-in hybrid-electric, to battery-electric vehicles, to hybrid-hydraulic and hybrid-mechanical solutions. The increased complexity of the powertrain systems associated with hybrid vehicles presents interesting control challenges and problems. This entry describes control problems associated with hybrid-electric vehicles (HEVs) and battery-electric vehicles (BEVs).

HEV Powertrains

An HEV powertrain contains at least two power sources: a primary engine – typically a combustion engine or a fuel cell fueled by a chemical fuel (in liquid or gaseous form) – and a secondary power source that makes use of



Powertrain Control for Hybrid-Electric and Electric Vehicles, Fig. 1 Hybrid powertrain configurations (After Rizzoni and Peng (2013), courtesy: Dr. Chiao-Ting Li, the University of Michigan)

a rechargeable energy storage system (RESS) that permits buffering the power demand of the vehicle so as to provide choices in the use of the power sources. While it is possible to design hybrid powertrains using secondary hydraulic or mechanical energy conversion and storage devices (hydraulic pump/motors and accumulators, mechanical flywheels), the majority of hybrid powertrains in use today employ electric machines and electrochemical energy storage devices (batteries and supercapacitors); thus, this entry focuses exclusively on *hybrid-electric* vehicles (HEVs). Electric vehicles (EVs) can be viewed as a special case of HEVs in which no internal combustion engine is present, and many of the considerations that follow apply also to hydraulic and mechanical

hybrids. HEVs may be classified according to their powertrain architecture as shown in Fig. 1.

A *series HEV powertrain* employs an electric machine (EM) to propel the vehicle while using an internal combustion engine (ICE) coupled to a second EM as an electrical generator set. In a series HEV, the electrical generator set can provide power directly to the electric traction system, via an electrical DC bus, or can charge an RESS (e.g., battery), or can perform both functions. Motive power to the vehicle is delivered by the primary EM. Thus, a series HEV blends electrical power from an RESS with electrical power generated by an ICE-powered generator set to provide motive power to the vehicle. Deciding how much electrical power to draw from

each of the two power sources to meet the power demand of the vehicle is an important control objective. A further feature of interest is the ability to recover some of the kinetic energy of the vehicle during braking events by using the traction EM in generator mode to recharge the RESS.

A *parallel HEV powertrain* blends mechanical power from the ICE and one or more EMs through appropriate mechanical coupling and transmission elements to deliver motive power to the vehicle or to recharge the RESS. In a parallel HEV powertrain, the same EM is used to provide power to the vehicle (motor mode) and to provide energy to the RESS (generator mode); in the latter case, the RESS can be recharged either by providing power from the ICE in excess of that required by the vehicle or by converting the kinetic energy of the vehicle into electrical power through the braking action of the EM.

A third configuration, the one that is most commonly found among passenger vehicles in commercial production today, is the *power-split HEV*, in which the benefits of both series and parallel HEVs are achieved most commonly by using one or more planetary gear sets to couple two EMs – to the ICE on one side and to the driveline on the other.

Regardless of architecture, HEV powertrains enable fuel savings and emissions reductions by operating in a variety of modes that include *load leveling*, *regenerative braking*, *engine start-stop*, and *transmission optimization* (Rizzoni and Peng 2013; Miller 2004). All of these functions benefit from the availability of an RESS and of bidirectional power converters, that is, the electric drive system(s) that can serve both motor and generator functions.

HEV Operation

An HEV is considered *charge sustaining* if the RESS is recharged only by power supplied by the ICE or by regenerative braking. If, on the other hand, the vehicle is designed to deplete energy stored in the RESS during the course of a trip, ending the trip with a lower state of stored energy than at the start and requiring recharging from the

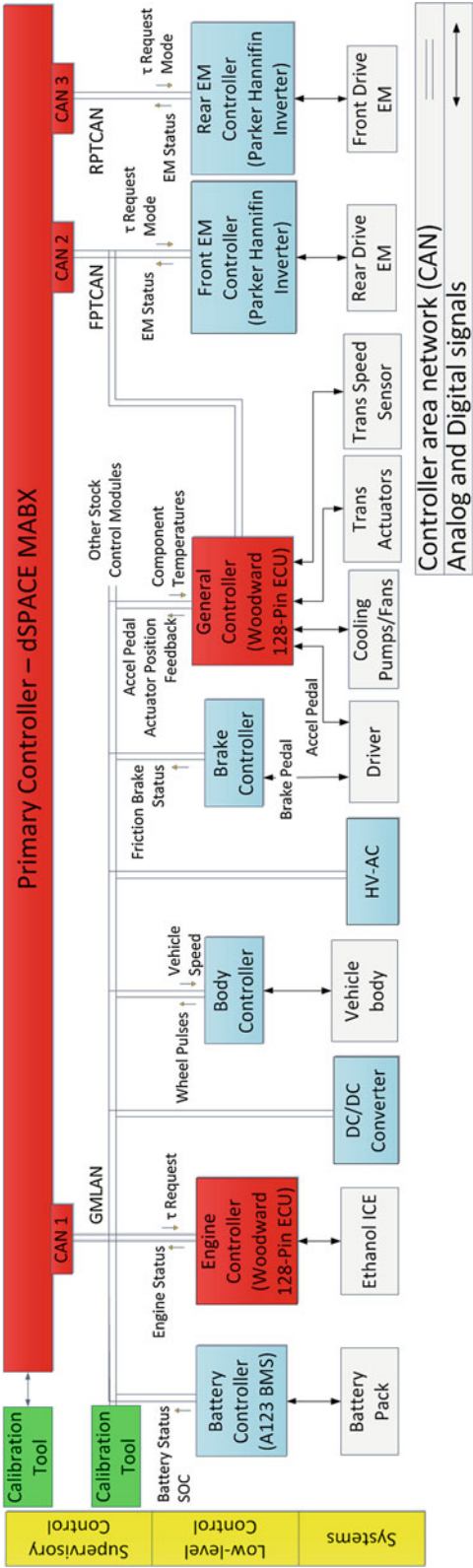
electrical grid, the vehicle is called *charge depleting* and is commonly referred to as a *plug-in HEV* (PHEV). PHEVs can in turn be subdivided into blended-mode PHEVs, in which stored electrical energy and fuel chemical energy are used jointly to achieve minimum overall energy use, and extended-range electric vehicles (EREVs), in which electrical energy is used exclusively to power the vehicle, until a lower bound is reached, at which point the vehicle uses both ICE and EM(s) to behave like a charge-sustaining hybrid. In principle, any of the architectures of Fig. 1 can be used in any of these modes. A battery-electric vehicle, or BEV, is an extreme case of an EREV, in which the vehicle is not equipped with an ICE. Miller (2004) provides an excellent overview of the technology underlying each of the powertrain architectures mentioned so far.

Control Problems in X-EVs

Let us refer to the general case of a hybrid or electric vehicle as an X-EV, with the X in X-EV representing any of the architecture discussed so far: X = H, PH, ER, or B. X-EVs enable multiple configurations and operating modes of the powertrain, presenting a number of interesting control problems above and beyond those that are already present in non-hybrid powertrains (e.g., engine and transmission control). In general, the control architecture of an HEV is hierarchical, with a higher-level (supervisory) controller that manages the power flows and mode changes (e.g., from electric to hybrid in an EREV) to meet the vehicle fuel economy, emissions, performance, and drivability requirements. Figure 2 depicts a hierarchical control architecture in use in a prototype PHEV.

In an X-EV, two problems are especially important: *optimal energy management*, that is, the ability to optimize the energy use of a vehicle during a trip, and *mode switching*, that is, the ability to select the appropriate operating mode and to smoothly switch between modes.

The higher-level controller issues set points to lower-level controllers that are used to



Powertrain Control for Hybrid-Electric and Electric Vehicles, Fig. 2 Hierarchical structure of an HEV controller (Courtesy: The Ohio State University EcoCAR 2 Team)

manage the ICE, the EM(s), the mechanical transmission system, the brake system, and the RESS, as well as other auxiliary functions in the vehicle. In this article we primarily consider the higher-level controller and focus on two problems that are especially relevant to HEVs: *optimal energy management* and *mode switching*. In addition, we also consider the battery controller (often called *battery management system* or BMS), which while being a low-level control is very specific to X-EVs.

Optimal Energy Management

The optimal energy management problem in an X-EV consists of finding the control $u(t)$ that leads to the minimization of a performance index J over the time horizon $t - t_f$, corresponding to a driving cycle, or *trip*; the problem is subject to constraints that are related:

- (i) To physical limitations of the actuators and the energy stored in the RESS
- (ii) To the requirement to maintain the RESS state of energy within prescribed limits (in a charge-sustaining X-EV) or to track a specified RESS stored energy trajectory (in charge-depleting X-EVs)

Let $L(\cdot)$ be a suitable function of the system states and inputs that accounts for the quantities we wish to minimize, for example, fuel consumption or emissions of carbon dioxide. Then, we define the cost function

$$J(x(t), u(t)) = \int_{t_0}^{t_f} L(x(t), u(t), t) dt \quad (1)$$

which is to be minimized for every trip. In general, the exact driving cycle, or profile, associated with a trip is not completely known; thus, a causal solution to this problem is impossible to achieve without making some assumptions. Various approaches to solve (1) have been proposed over the years; we cite (i) dynamic programming (DP), (ii) local optimization solutions as surrogates of a global solution, (iii) Pontryagin's

minimum principle (PMP), and (iv) rule-based methods. Onori et al. (2014) provide a comprehensive overview of the problem as well as detailed examples. We briefly review approaches (i), (ii), and (iii) in the present article.

Global Optimization by Dynamic Programming

If the driving cycle, represented by the vehicle instantaneous velocity over time, $v(t)$ is known, it is possible to cast (1) in such a form that a DP solution is possible. For example, in a charge-sustaining X-EV, one can find the sequence of inputs that minimizes the trip fuel consumption while sustaining the desired state of charge of a battery and meeting the speed profile of the vehicle. In this problem, the input is the power supplied by the battery to the electric machine, and the state of charge of the battery, SOC, is the only state; all other subsystems (engine, electric drives, transmission, etc.) are modeled via quasi-static efficiency models that can be represented by algebraic equations (e.g., Willans lines, Rizzoni et al. 1999) or by maps. The vehicle velocity profile, $v(t)$ is converted to a vehicle power request, $P_{REQ}(t)$, knowing the vehicle load characteristics (aerodynamic, inertial, rolling and drivetrain friction, and road grade). In turn, the power required to meet a specific load profile is the sum of the power delivered by the ICE and EM, $P_{REQ}(t) = P_{ICE}(t) + P_{BAT}(t)$. So, for example, we seek the control input, $P_{BAT}(t)$, that corresponds to the minimum fuel consumption, that is, $\min_{\{P_{ICE}(t), P_{BAT}(t) \forall t\}} \int_{t_0}^{t_f} \dot{m}_f(t) dt$, while delivering the requested vehicle power. The problem has physical constraints in the actuators (maximum and minimum power that can be delivered by ICE and EM), as well as the requirement for the control policy to be charge sustaining, which is translated into the additional condition $SOC(t_0) = SOC(t_f)$. While this is only a sketch of the problem formulation (see Onori et al. (2014) for a detailed treatment), it should be clear that it is possible to find a DP solution. If the

vehicle is charge depleting, the problem can be similarly formulated with $\text{SOC}(t_f) < \text{SOC}(t_0)$.

In practice, this approach requires complete information of the vehicle velocity profile, and DP is not an implementable, causal solution to the X-EV energy management problem. It is, however, a very useful tool to establish a benchmark for a problem or as an aid in developing a rule base (Onori et al. 2014). Stochastic DP methods have been proposed to circumvent the need to know the driving cycle exactly (see, e.g., Tate et al. 2007).

Local Optimization by Equivalent Fuel Consumption Minimization

A heuristic approach that has met with success is to solve (1) as a local optimization problem,

wherein $\int_{t_0}^{t_f} \min_{\{P_{ICE}(t), P_{BAT}(t) \forall t\}} \dot{m}_f(t) dt$ is used as an

approximation for $\min_{\{P_{ICE}(t), P_{BAT}(t) \forall t\}} \int_{t_0}^{t_f} \dot{m}_f(t) dt$.

This approach gives rise to the *Equivalent fuel Consumption Minimization Strategy* (ECMS) (Paganelli et al. 2001), which accounts for the use of stored electrical energy, in units of chemical fuel use (g/s), such that one can define an “equivalent fuel consumption” taking into account the cost of the electrical energy used to produce $P_{BAT}(t)$ by way of the fuel that must be used at a future time to replenish the stored electrical energy in the RESS. The equivalent fuel consumption is defined in (2):

$$\dot{m}_{f,eq}(t) = \dot{m}_f(t) + \dot{m}_{eq}(t) = \dot{m}_f(t) + s(t) \frac{E_{batt}}{Q_{LHV}} \dot{\text{SOC}}(t) \quad (2)$$

In (2), $\dot{m}_{f,eq}$ is the equivalent fuel consumption, $\dot{m}_f$ is the actual chemical fuel consumption, $\dot{m}_{eq}$ is the virtual fuel consumption corresponding to the use of electricity stored in the battery (to be replenished in the future), E_{BAT} is the energy capacity of the battery, Q_{LHV} is the lower heating value of the chemical fuel, and $s(t)$ is the *equivalence factor* that assigns a cost to the use of electricity. Then, the global minimization problem of (1), with $L(\cdot)$ equal to $\dot{m}_{f,eq}$, becomes the

problem of finding $\int_{t_0}^{t_f} \min_{\{P_{ICE}(t), P_{BAT}(t) \forall t\}} \dot{m}_f(t) dt$.

This approach, which can be easily implemented, has been used widely and has been shown to closely approximate the global optimal solution if sufficient knowledge of the vehicle driving cycle is available. The method does require empirical calibration and tuning of the equivalence factor, $s(t)$, the optimal value of which is dependent on the driving cycle. Such calibration could be automated by using a predictor to generate a short-horizon estimate of the driving cycle and an adaptor to generate an appropriate $s(t)$ (Musardo et al. 2005).

Optimization by Pontryagin's Minimum Principle

Pontryagin's minimum principle (PMP) can also be employed to solve the X-EV energy management problem. If, again, the fast dynamics of the system are neglected the state equation is

$$\dot{x}(t) = f(x, u, t) = -\frac{1}{E_{BAT}} I_{BAT}(x, u, t) \quad (3)$$

where $x = \text{SOC}$ is the state of charge of the battery, E_{BAT} is the energy capacity of the battery, and I_{BAT} is the instantaneous battery current. If the input is the power requested of the battery, $P_{BAT}(t)$, which in turn determines the engine power request, $P_{ICE}(t)$, and hence the fuel consumption, then the Hamiltonian function can be defined to be

$$H(x(t), P_{BAT}(t), \lambda(t)) = \dot{m}_f(P_{BAT}(t)) - \lambda(t) \cdot f(x(t), P_{BAT}(t), t) \quad (4)$$

In (4), $f(\cdot)$ is given by Eq. (3), and the control $P_{BAT}(t)$ that which minimizes Eq. (4) at each time instant is

$$P_{BAT}^*(t) = \arg \min_{P_{BAT}} H(x(t), P_{BAT}(t), \lambda(t)) \quad (5)$$

The co-state variable, $\lambda(t)$, is the solution of

$$\dot{\lambda}(t) = -\lambda(t) \frac{\partial f(x(t), u(t))}{\partial x} \quad (6)$$

Eqs. 3 and 5, with boundary conditions $x(t_0)$ and $x(t_f)$, can be solved numerically; in Serrao et al. (2009, 2011) it is shown that the co-state $\lambda(t)$ is related to the *equivalence factor* of Eq. (2), confirming that the intuitive ECMS solution is in fact the PMP solution, providing that the equivalence factor (or co-state) is time varying and satisfies

$$H(t, x, u, \lambda) = \dot{m}_f + \lambda(t)\dot{x}(t) \quad \text{and} \\ s(t) = -\lambda(t) \frac{Q_{LHV}}{E_{BAT}} \quad (7)$$

The PMP solution is also cycle dependent, as the optimal initial condition for the co-state is dependent on the driving cycle. This dependence on the driving cycle, whether expressed in terms of an equivalent fuel consumption in the ECMS solution or as the initial condition of the co-state in the PMP solution, is an unavoidable consequence of the fact that the fuel consumption of a vehicle is strongly dependent on the driving conditions, which affect the vehicle load.

The basic concepts outlined above continue to be the subject of further development; for example, integrating available trip information available from navigation and geographical information systems into predictive energy management algorithms and considering battery aging as a cost in the optimization function are but two of the research areas being pursued.

Mode Switching

X-EV architectures permit multiple operating modes to exploit the design and control flexibility available in the powertrain. Some examples are the following: an X-EV could operate in pure EV mode or in hybrid mode (whether series, parallel, or power-split), could use special control algorithms during regenerative braking events to provide maximum energy recovery without adversely affecting brake and vehicle stability control systems, and could implement special start-stop control strategies that minimize fuel consumption at idle without adversely affecting engine cold- or warm-start emissions and

without inducing unwanted transient vibrations (Canova et al. 2009). Figure 3 depicts an example of a state flow diagram that could be implemented in a finite state machine. Mode switching can result in *drivability* problems (Wei and Rizzoni 2004), that is, in undesirable transient response characteristics during mode changes. An X-EV can, in this context, be represented as a hybrid system (Koprubasi et al. 2007).

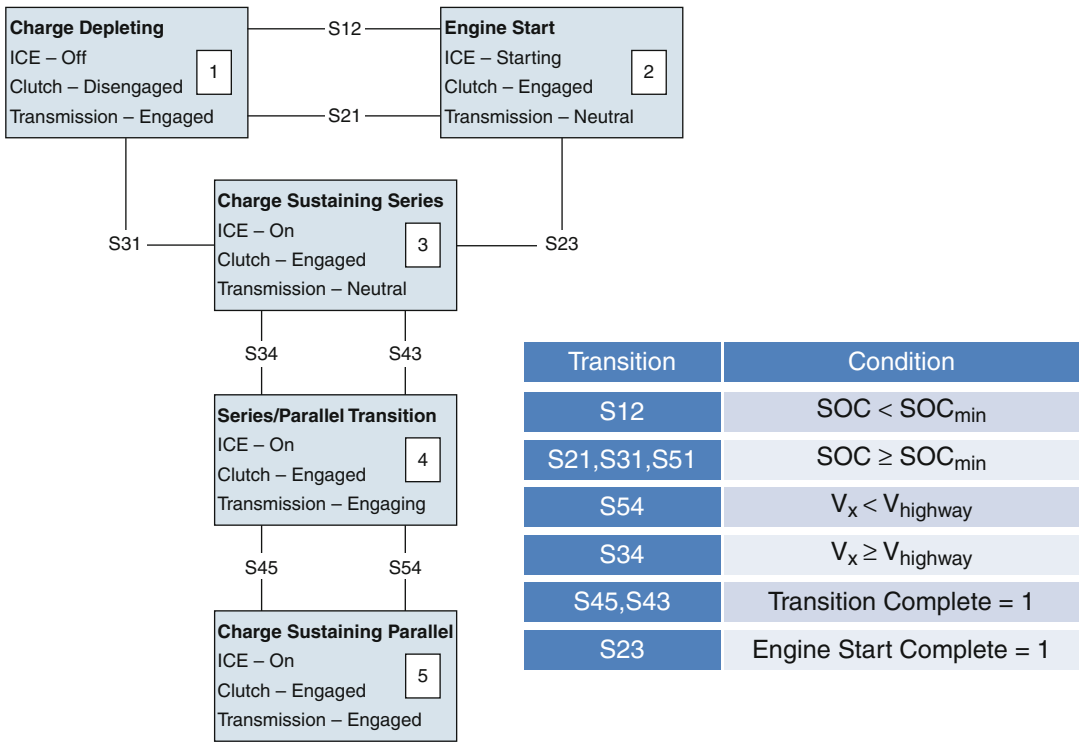
Battery Management Systems

The most common RESS in hybrid vehicle is the electrochemical battery. A hybrid or electric vehicle uses a battery pack that is typically composed of modules, which are in turn comprised of battery cells connected in series and parallel. Battery management systems are necessary to provide charge balancing, cell protection, state of charge and state of health estimation, and other functions related to the management of the stored energy. A good overview of battery systems and associated control problems may be found in Rahn and Wang (2013).

Two important problems related to battery management are state of charge (SOC) and state of health (SOH) estimation. SOC estimation is a necessary component of any battery management system. The SOC of battery is defined by the following equations, in which x is the SOC, Q_{BAT} is the battery capacity in ampere-hours, and η is the battery charging/discharging efficiency:

$$\dot{x}(t) = \frac{\eta}{Q_{BAT}(t)} \cdot I_{BAT}(t) \quad x(t) = x(t_0) \\ + \frac{1}{3,600 \cdot Q_{BAT}(t)} \int_{t_0}^{t_f} I_{BAT}(\tau) \cdot d\tau \quad (8)$$

In practice, there are two problems with using current integration (also called *Coulomb counting*) to estimating SOC: (i) errors in numerical integration accumulate and may cause significant bias error in the estimate, and (ii) the actual capacity of the battery is unknown during vehicle operation, as it changes over



Powertrain Control for Hybrid-Electric and Electric Vehicles, Fig. 3 State diagram illustrating mode switching in a PHEV (Courtesy: The Ohio State University EcoCAR 2 Team)

time due to battery aging. A second SOC estimation approach consists of correlating the battery open-circuit voltage to the SOC, but this approach also suffers from significant uncertainty, as the open-circuit voltage-SOC correlation curves are only accurate in stationary conditions (constant temperature, with battery at rest). SOC estimation has been the subject of much research and has seen the use of Kalman filters, extended Kalman filters, particle filters, and other estimation approaches (Chaturvedi et al. 2010).

The SOH of a battery degrades over time due to two principal factors: *capacity fade* and *power fade* (which can also be thought of as *conductance fade* caused by an increase in the internal resistance of the battery). These phenomena are the result of complex electrochemical interactions that are specific to battery chemistry. The ability to estimate the capacity and resistance of a battery during actual operation is a very important aspect of battery management. As in

the case of SOC estimation, no direct measurement is possible outside of controlled laboratory conditions; hence, estimation algorithms must be employed (Chaturvedi et al. 2010). It is important to observe that SOC and SOH estimation algorithms operate on two completely different time scales, as the SOC of a battery fluctuates over time windows of minutes or hours, while the SOH changes very slowly over time, with measurable changes occurring over periods of months or years.

Summary and Future Directions

In summary, the control of X-EV powertrains is a rich subject for control theoreticians and practitioners, presenting topics related to optimization and optimal control (for energy management, battery aging), hybrid control (for drivability), adaptive and predictive control, and estimation. Further, the electrification of

ground vehicles presents interesting opportunities to integrate vehicles with the electric power and communication networks infrastructures. The following paragraphs describe two such opportunities.

Vehicle-Grid Interaction

As the penetration of plug-in vehicles, PHEVs and BEVs, increases, their impact on the electric power grid cannot be neglected; the consideration of increased electric power demand and of the timing of vehicle charging must be included in the control/optimization of the electric power grid.

The electric grid and the transportation system are the two largest sectors that produce greenhouse gas emissions. When large numbers of vehicles are electrified and draw power from the electric grid, it is important to aim for reduced overall greenhouse gas emissions rather than just shifting emissions from tailpipes to power plant stacks. Controlling the charging of plug-in vehicles to alleviate the impact to the grid has been studied, including the idea of using plug-in vehicles as ancillary services to the grid, possibly with significant renewable power sources connected to the grid. Modeling and simulating this integrated system require information on detailed grid load profiles, power generation pricing and carbon emissions, wind statistics, and vehicle usage statistics. In addition, charging control must balance multiple factors: grid stability, fully charging all vehicles, minimizing data collection and communication, and overall system carbon emission minimization.

Intelligent Transportation Systems

X-EVs, as well as conventional vehicles, will benefit from the ability to analyze traffic and geographical information in real time to quantify the effects of infrastructure, environment, and traffic flow on vehicle fuel economy and emissions, and to permit the application of forecasting and optimization methods for energy management (Gong et al. 2011; Wollaeger et al. 2012). There are significant opportunities to achieve significant fuel savings and emissions reduction by considering the large-scale interactions of vehicles with

one another and with the infrastructure, further exploiting the flexibility inherent in X-EVs.

Cross-References

- [Engine Control](#)
- [Fuel Cell Vehicle Optimization and Control](#)
- [Optimal Control and Pontryagin's Maximum Principle](#)
- [Optimal Control and the Dynamic Programming Principle](#)

Bibliography

- Canova M, Guezennec Y, Yurkovich S (2009) On the control of engine start/stop dynamics in a hybrid electric vehicle. *ASME J Dyn Syst Meas Control* 131: 061005
- Chaturvedi NA, Klein R, Christensen J, Ahmed J, Kojic A (2010) Algorithms for advanced battery management systems. *IEEE Control Syst Mag* 30(2):49–68
- Gong Q, Tulpule P, Midlam-Mohler S, Marano V, Rizzoni G (2011) The role of ITS in PHEV performance improvement. In: American control conference, San Francisco
- Koprubasi K, Westervelt ER, Rizzoni G (2007) Toward the systematic design of controllers for smooth hybrid electric vehicle mode changes. In: Proceedings of the American control conference, Anchorage
- Miller JM (2004) Propulsion systems for hybrid vehicles. The Institution of Electrical Engineers, London
- Musardo C, Rizzoni G, Guezennec Y, Staccia B (2005) A-ECMS: an adaptive algorithm for hybrid electric vehicle energy management. *Eur J Control* 11(4–5): 509–524
- Onori S, Serrao L, Rizzoni G (2014) Energy management strategies for hybrid electric vehicles. Springer, Berlin
- Paganelli G, Ercole G, Brahma A, Guezennec Y, Rizzoni G (2001) General supervisory control policy for the energy optimization of charge-sustaining hybrid electric vehicles. *JSAE* 22: 511–518
- Rahn C, Wang C-Y (2013) Battery systems engineering. Wiley, New York
- Rizzoni G, Peng H (2013) Hybrid and electric vehicles: the role of dynamics and control. *ASME Dyn Syst Control Mag* 1(1):10–17
- Rizzoni G, Guzzella L, Baumann B (1999) Unified modeling of hybrid-electric vehicle drivetrains. *IEEE/ASME Trans Mechatron* 4(3):246–257
- Serrao L, Onori S, Rizzoni G (2009) ECMS as a realization of Pontryagin's minimum principle for HEV

control. In: Proceedings of the 2009 American control conference, Portland

Serrao L, Onori S, Rizzoni G (2011) A comparative analysis of energy management strategies for hybrid electric vehicles. *ASME J Dyn Syst Meas Control* 133:1–9

Tate ED, Grizzle JW, Peng H (2007) Shortest path stochastic control for hybrid electric vehicles. *Int J Robust Nonlinear Control* 18(14):1409–1429

Wei X, Rizzoni G (2004) Objective metrics of fuel economy, performance and driveability – a review. SAE Technical paper 2004-01-1338

Wollaeger SK, Onori S, Di Cairano S, Filev D, Ozguner U, Rizzoni G (2012) Cloud-computing based velocity profile generation for minimum fuel consumption: a dynamic programming based solution. In: American control conference, Montreal, 27–29 June 2012

Privacy in Network Systems

Jerome Le Ny

Department of Electrical Engineering and GERAD, Polytechnique Montreal, Montreal, QC, Canada

Abstract

A growing number of large-scale monitoring and control systems enabling a more intelligent infrastructure, such as smart grids and intelligent transportation systems, rely for their operation on private data obtained from individuals or from privacy-sensitive entities more generally. This entry discusses how to capture and enforce formal quantitative privacy constraints to design estimation and control algorithms that can trade off performance for the level of privacy offered. A state-of-the art definition of privacy called *differential privacy*, which provides guarantees against adversaries with arbitrary side information, is presented. It is shown that tools from systems and control can be used to enforce differential privacy guarantees when publishing real-time statistics or solving distributed optimization and control problems based on sensitive data.

Keywords

Data privacy · Differential privacy · Stochastic filtering · Networked systems

Introduction

As real-time prediction, decision, and control technologies increasingly target sensor-rich large-scale *socio-technical* systems, such as intelligent transportation systems or smart grids, new challenges arise to design algorithms that properly interact with and take into account the human component of these systems. The focus of this entry is on the sensing side, where heavy reliance on processing large amounts of detailed data collected from individuals implies that certain privacy constraints must be satisfied, either due to legal requirements or more simply because people tend to value privacy and might refuse to provide their data to overly intrusive and insecure systems.

The development of formal approaches to reason about privacy in data analysis and of mechanisms to embed verifiable privacy-preserving guarantees directly into our algorithms is motivated by the fact that privacy attacks are often difficult to anticipate based on intuition alone. The naive anonymization of microdata (i.e., data directly related to individuals) by removing names and other obvious means of personal identification is generally insufficient, since de-anonymization attacks can be carried out by combining the information contained in the dataset with other sources of information (Sweeney 2002). As another example, consider the fact that the detailed activities of a household can be inferred from the electricity consumption traces collected by smart meters (Hart 1992).

Defining Privacy

In general, privacy-preserving data analysis requires some form of data obfuscation, thereby sacrificing some level of accuracy. Traditionally, various types of “disclosure limitation”

techniques (Duncan and Lambert 1986) have been applied by statisticians and economists when publishing statistics computed from private microdata, such as census data. This includes adding noise, clustering, modifying, and discarding certain parts of the original dataset, but typically does not aim to achieve a formally defined notion of privacy.

Somewhat more recently, several definitions of privacy have thus been proposed, including k -anonymity (Sweeney 2002) and its various extensions, information-theoretic measures of privacy (Sankar et al. 2013), conditions based on observability (Manitara and Hadjicostis 2013), and differential privacy (Dwork et al. 2006). More details can be found in Le Ny (2020, Chapter 1). For example, k -anonymity is aimed at countering re-identification attacks on microdata and constrains the publication of a dataset in such a way that each combination of values for the sensitive attributes occurs in at least k individual records.

Currently the notion of differential privacy is the focus of most of the research in privacy-preserving data analysis. It is motivated by the fact that useful statistical analyses on a dataset result in unavoidable information disclosure about individuals, even those individuals whose data is *not* in the dataset. For example, paraphrasing (Dwork and Roth 2014), a study that links smoking to an increased risk of cancer provides new information about every person known to be a smoker, not just the participants in the study. Hence, differential privacy does not aim to prevent information disclosure per se, as some other approaches, but instead ensures that new inferences cannot be made much more easily about the individuals who participated in a study compared to those who did not.

To introduce the formal definition of differential privacy, fix a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and let $\mathbf{D}$ be a set of possible datasets of interest, which could be, for example, a set of data tables with a certain structure, or a set of signals. Introduce a (application-dependent) symmetric binary relation on $\mathbf{D}$, called adjacency and denoted Adj . Adjacent datasets might correspond, for example, to adding or removing the record of one

individual. A *mechanism* M is a randomized map with inputs in $\mathbf{D}$, i.e., for any $d \in \mathbf{D}$, $M(d)$ is a random variable with values in some measurable space $(\mathbf{R}, \mathcal{M})$, where $\mathcal{M}$ denotes a σ -algebra on $\mathbf{R}$. Then, M is said to be differentially private if it is hard to distinguish $M(d)$ from $M(d')$ for any two adjacent d, d' (i.e., any $(d, d') \in \text{Adj}$). Concretely, these two random variables should have almost indistinguishable distributions, as captured by the following definition (Dwork et al. 2006).

Definition 1 Let $\epsilon, \delta \geq 0$ be nonnegative scalars. With the above notation, M is said to be (ϵ, δ) -differentially private for Adj (abbreviated (ϵ, δ) -dp) if for all sets $S \in \mathcal{M}$ and all adjacent datasets $(d, d') \in \text{Adj}$, we have

$$\mathbb{P}(M(d) \in S) \leq e^\epsilon \mathbb{P}(M(d') \in S) + \delta. \quad (1)$$

If (1) holds with $\delta = 0$, we say simply that the mechanism is ϵ -differentially private.

In Definition 1, the parameters ϵ and δ characterize the level of privacy, with smaller values for these parameters corresponding to a stronger privacy guarantee, since the distributions of $M(d)$ and $M(d')$ are then closer. Note also that the privacy guarantee offered depends on the choice of adjacency relation, not just ϵ and δ . Hence, if we change the adjacency relation to one that admits more adjacent datasets, it might become more difficult to achieve the same levels ϵ, δ .

An important property of any privacy guarantee is that it should be *resilient to post-processing*, i.e., once a private output is released, it should remain private no matter how one further uses it, as long as the original sensitive data is not re-accessed. For differential privacy, resilience to post-processing corresponds to composing an (ϵ, δ) -dp mechanism with a subsequent 0-dp one to obtain a mechanism that should remain (ϵ, δ) -dp. More generally, we have the following basic composition theorem:

Theorem 1 Let D be a space with an adjacency relation Adj . Let M_1 be an (ϵ_1, δ_1) -dp mechanism on D with values in $\mathbf{R}_1$. Let M_2 be a mechanism on $\mathbf{R}_1 \times D$ such that, for any $r_1 \in \mathbf{R}_1$, $M(r_1, \cdot)$

is (ϵ_2, δ_2) -dp. Then the composed mechanism $M(d) := M_2(M_1(d), d)$ is $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ -dp. By immediate induction from Theorem 1, the ϵ, δ parameters add when composing any finite number of mechanisms, although this is a generic result providing only an upper bound on these privacy parameters. More refined composition theorems also exist (Dwork and Roth 2014).

Basic Differentially Private Mechanisms

Definition 1 specifies a constraint that a given algorithm should satisfy to be considered private, but not how to achieve this constraint. Since the methods used for differential privacy depend on the applications considered, we now specialize the discussion to datasets consisting of multi-dimensional numeric signals and view mechanisms as (stochastic) dynamic systems transforming these signals. Let $\mathbf{S} := \mathbb{R}^N$ and $\mathbf{S}^m$ represent the set of discrete-time sequences with values in $\mathbb{R}$ and $\mathbb{R}^m$, respectively. For an integer $1 \leq p < \infty$, define the ℓ_p -norm of a signal x in $\mathbf{S}^m$ by $\|x\|_p = (\sum_{t=0}^{\infty} |x_t|_p^p)^{1/p}$, where $|\cdot|_p$ represents the p -norm on $\mathbb{R}^m$, i.e., $|v|_p = (\sum_{i=1}^m |v_i|^p)^{1/p}$, with v_i the i^{th} component of $v \in \mathbb{R}^m$. Two basic mechanisms, the Laplace and the Gaussian mechanism, produce differentially private signals by simply adding an appropriate level of white noise. A standard approach to set the noise level involves the following quantity:

Definition 2 Let $G : \mathbf{S}^n \rightarrow \mathbf{S}^m$ be a system with n inputs and m outputs. Given an adjacency relation Adj on $\mathbf{S}^n$, the ℓ_p -sensitivity of G is defined for $1 \leq p < \infty$ as

$$\Delta_p G = \sup_{(u, u') \in \text{Adj}} \|Gu - Gu'\|_p.$$

In other words, the ℓ_p -sensitivity of a system is a type of incremental gain but for variations in the input restricted to adjacent signals.

The following definitions are needed to introduce the basic mechanisms. The Laplace probability distribution with mean zero and variance

$2b^2$, which has density $\frac{1}{2b} e^{-|x|/b}$ on $\mathbb{R}$, is denoted $\text{Lap}(b)$. We write $X \sim \text{Lap}(b)^n$ to specify that a random vector $X \in \mathbb{R}^n$ has independent and identically distributed (iid) coordinates, all following a $\text{Lap}(b)$ distribution, with corresponding density $\frac{1}{(2b)^n} e^{-|x|_1/b}$ on $\mathbb{R}^n$. The multivariate Gaussian distribution with mean vector μ and covariance matrix Σ is denoted $\mathcal{N}(\mu, \Sigma)$. The Q -function $Q(x) := \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-u^2/2} du$ represents the tail distribution of the standard normal distribution. Since Q is monotonically decreasing, it has an inverse Q^{-1} , which maps the interval $(0, 1)$ into $(-\infty, \infty)$. For $\epsilon > 0$, $1 > \delta > 0$, define the notation $\kappa_{\delta, \epsilon} = \frac{1}{2\epsilon} (Q^{-1}(\delta) + \sqrt{(Q^{-1}(\delta))^2 + 2\epsilon})$.

Theorem 2 Let $G : \mathbf{S}^n \rightarrow \mathbf{S}^m$ be a system with finite ℓ_1 - or ℓ_2 -sensitivity $\Delta_1 G$ and $\Delta_2 G$ with respect to a given adjacency relation Adj on $\mathbf{S}^n$. Then, the mechanism $Mu := Gu + w$, where w is a white noise sequence (iid zero-mean random vectors) such that for all $t \geq 0$, $w_t \sim \text{Lap}(b)^m$ with $b \geq \Delta_1/\epsilon$, is ϵ -dp for Adj . If instead the white noise w is Gaussian with covariance matrix $\sigma^2 I_m$, where $\sigma \geq \kappa_{\delta, \epsilon} \Delta_2 G$, then the mechanism is (ϵ, δ) -dp for Adj .

The Laplace and Gaussian mechanisms of Theorem 2 are already useful by themselves to produce differentially private signals. In particular, if G is the identity operator in Theorem 2, we can typically compute its ℓ_p -sensitivity $\sup_{(u, u') \in \text{Adj}} \|u - u'\|_p$ accurately for reasonable adjacency relations. This gives a simple mechanism, often called the *input perturbation mechanism*, which makes a signal differentially private by simply adding sufficient white noise to it, at the data source, and does not require any knowledge about the computations subsequently performed on the signal, relying instead on the resilience to post-processing property mentioned above Theorem 1. However, much better accuracy can often be achieved by tailoring the noise to the desired computational task specified by a non-trivial system G in the mechanism of Theorem 2, which is then called an *output perturbation mechanism*.

Example 1 Suppose that n individuals produce each a signal $u_i \in \mathbf{S}$, for $1 \leq i \leq n$. The space

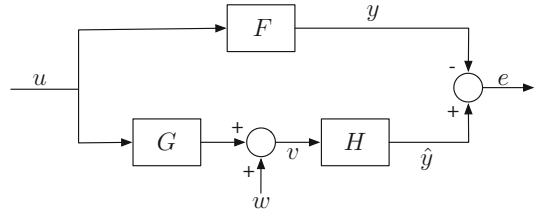
D of datasets is $\mathbf{S}^n$. Fix some constant $\rho > 0$ and the adjacency relation

$$\begin{aligned} (u, u') \in \text{Adj} &\iff \exists i \in \{1 \dots, n\} \\ &\text{such that } \|u_i - u'_i\|_2 \leq \rho \\ &\text{and } u_j \equiv u'_j, \forall j \neq i. \end{aligned}$$

Differential privacy for this adjacency relation aims to hide bounded variations in the energy of any single individual's signal. Suppose that an analyst wishes to process the signal u through a stable linear time-invariant (LTI) filter G (with zero initial condition), to obtain an aggregate signal $y = Gu = \sum_{i=1}^n G_i u_i$. In the presence of a differential privacy constraint, a stochastic signal $\hat{y}$ is produced instead. The accuracy of the output can be measured by the steady-state mean squared error (MSE)

$$\lim_{T \rightarrow \infty} \mathbb{E} \left[\frac{1}{T+1} \sum_{t=0}^T |y_t - \hat{y}_t|_2^2 \right]. \quad (2)$$

Suppose for concreteness that we use the Gaussian mechanism. With input perturbation, we have $\hat{y} = G(u + w) = y + \sum_{i=1}^n G_i w_i$, where $w_{i,t} \sim \mathcal{N}(0, \kappa_{\delta, \epsilon}^2 \rho^2)$. The MSE is then the variance of the error term Gw , which is $\kappa_{\delta, \epsilon}^2 \rho^2 \sum_{i=1}^n \|G_i\|_2^2$, where $\|G_i\|_2$ is the H_2 -norm of G_i . On the other hand, for the output perturbation mechanism, we start by computing the ℓ_2 -sensitivity $\Delta_2 G = \max_{1 \leq i \leq n} \sup_{\|u_i - u'_i\|_2 \leq \rho} \|G_i u_i - G_i u'_i\|_2 = \rho \max_{1 \leq i \leq n} \{\|G_i\|_\infty\}$, where $\|G_i\|_\infty$ is the H_∞ -norm of G_i . The signal produced is then $\hat{y} = y + \zeta$, and the MSE is now the variance of ζ , namely, $\rho^2 \kappa_{\delta, \epsilon}^2 \max_i \{\|G_i\|_\infty^2\}$. If all the subsystems G_i are identical, for example, the second expression is independent of n , and hence the performance of the output perturbation mechanism becomes much better than with input perturbation for a sufficient large number of input signals.



Privacy in Network Systems, Fig. 1 Two-stage architecture for differentially private filtering

Differentially Private Filtering

Since the output perturbation mechanism releases signals perturbed by raw white noise, one can expect intuitively that performance could be improved by filtering out this noise. This leads to the two-stage architecture shown on Fig. 1. In this figure, F is the desired filter through which the sensitive signal u should be processed in absence of privacy constraint and G and H are systems to be designed in such a way that the error signal e between the desired output y and the private output $\hat{y}$ is small. Laplace or Gaussian noise w proportional to the sensitivity of G is added according to Theorem 2 to make the signal v differentially private, as well as $\hat{y}$ by resilience to post-processing. Note that the input and output perturbation mechanisms are special cases of this two-stage architecture, taking (G, H) to be (Id, F) and (F, Id) respectively.

Given an adjacency relation on the space of input signals u in Fig. 1 and a choice of sanitization mechanism (adding Laplace or Gaussian noise) to make the signal v private, the filters G and H of the two-stage differentially private mechanism can be designed using the following methodology:

1. Fix a measure of performance to evaluate the quality of a differentially private approximation $\hat{y}$ for a desired signal y .
2. Then, choose a structure for the block H , whose role is to estimate $y = Fu$ based on $v = Gu + w$. Since the block G has not yet been specified at this stage, H is expressed as a function of G .

3. Finally, express the performance measure as a function of G only, and find a system G optimizing this performance.

This methodology has been applied to a number of different scenarios. If nothing is known about the signal u , if adjacent signals differ at a unique time period, and if the performance measure is the MSE, then one can take $H = G^{-1}$ and find the optimal filter G by solving a spectral factorization problem (Le Ny and Pappas 2014). On the other hand, in many cases, some public information is available about the characteristics of u , even though the particular realization that needs to be processed is private. Consider, for example, the following scenarios, for the same adjacency relation:

- If the input signal u is wide-sense stationary with known second-order statistics and G is linear, then one should use a Wiener filter H .
- If moreover the input signal takes discrete values, then a decision-feedback equalizer can be used for H ; see Le Ny (2013).
- If the input signal u is modeled by the output of a linear system and G is linear, then H should be a Kalman filter; see Degue and Le Ny (2017).

Privacy-Preserving Distributed Computations

This section briefly discusses a class of distributed computing, optimization, and control problems subject to privacy constraints. These problems arise, for instance, when a number of agents wish to allocate resources or coordinate decisions in some optimal way but each agent is unwilling to share with the other agents some of the data necessary to set up a standard optimization problem. For example, the cost functions of firms attempting to develop a joint project could provide strategic information to their competitors, or some patients' data necessary to carry out a health-related study might be stored at multiple locations subject to regulations that prevent sharing them.

Consider n agents connected through a communication graph, where the state vector $x_{i,t}$ of agent i at time t is updated through a local rule such as

$$x_{i,t+1} = f_i(x_{i,t}, u_{i,t}, y_{\mathcal{N}_i,t}), \quad 1 \leq i \leq n, \quad (3)$$

where f_i is some function, u_i is a local control input for agent i , $\mathcal{N}_i$ is the set of indices of the neighbors of i in the graph, and $y_{\mathcal{N}_i,t}$ is a vector containing all the messages sent by these neighbors. Each agent generates messages of the form

$$y_{i,t} = g_i(x_{i,t}, v_{i,t}), \quad 1 \leq i \leq n, \quad (4)$$

for some function g_i and some random vector $v_{i,t}$, the latter being used by the agents to protect their private data, which might consist of their state x_i and possibly their local function f_i . Indeed, the agents might want to control their state trajectories in a coordinated manner in order to reach a certain global state or optimize a certain cost function, while keeping part of their individual data private, even though some or all the messages y_i are published, e.g., accessible by a third party monitoring the network.

In the class of distributed computing problems, privacy-preserving average consensus has been studied in several papers. In this case, the agents aim to asymptotically reach a consensus $x_{1,\infty} = \dots = x_{n,\infty} = \frac{1}{n} \sum_{i=1}^n x_{i,0}$ by exchanging messages (4). At the same time, each agent wants to keep its own initial state $x_{i,0}$ hidden, either from an adversary capable of observing all exchanged messages (Huang et al. 2012; Nozari et al. 2017) or from the other agents in the network (Manitara and Hadjicostis 2013; Mo and Murray 2016). The dynamics (3) are typically linear, uncontrolled, and known to all agents, and the individual privacy-preserving noise v_i in (4) needs to be designed. In general, one cannot achieve exact average consensus under a differential privacy constraint, simply because publishing the exact average of a set of private numbers violates differential privacy. Hence, either one must be satisfied with achieving only a randomized approximate average, or another notion of privacy

must be used (Manitara and Hadjicostis 2013; Mo and Murray 2016).

A number of distributed optimization problems have been studied under privacy constraints. For example, Hsu et al. (2016) propose a privacy-preserving primal-dual algorithm for convex optimization, when the global cost and the constraints coupling the agents can be written as sums of individual private functions. The iterations (3) describe each agent implementing a local primal step, using local dynamics f_i known only to this agent and a message y received from a trusted central data aggregator, as there is no direct communication between the agents in this case. After each primal iteration, the agents then send their current local state value to the aggregator performing a dual step updating the dual variables for the coupling constraints, which are then broadcasted back to all agents in the aggregator's message. Since the dual variables depend on the agents' data, privacy-preserving noise is added in this dual step.

Summary and Future Directions

When estimation and control algorithms rely on privacy-sensitive data, the participation of the independent agents providing this data must be encouraged through incentives and data protection guarantees. Formal definitions of privacy can be used to balance privacy costs and system performance improvements in such cases. This entry has discussed in particular how tools from systems and control can be used to design estimators for dynamic systems enforcing differential privacy guarantees, or reason about privacy in distributed computing and control systems. Another relevant and currently active research topic is the privacy-preserving detection of anomalies and abrupt changes in these systems. Significant challenges remain to formulate important privacy-preserving data analysis problems arising in large-scale monitoring and control systems, carefully characterize the achievable performance-privacy trade-offs, or identify the privacy definitions most relevant for different scenarios.

Cross-References

- Cyber-Physical Security
- Game Theory for Security

Recommended Reading

Differential privacy is covered in depth in Dwork and Roth (2014). For the privacy-preserving analysis of dynamic datasets, one can find much work using entropy-based definitions of privacy as well as differential privacy in the information theory and signal processing literature; see, for example, the special issue (Trappe et al. 2015). The tutorial paper Cortés et al. (2016) and the monograph Le Ny (2020) provide additional details on the topics discussed in this entry.

Bibliography

- Cortés J, Dullerud GE, Han S, Le Ny J, Mitra S, Pappas GJ (2016) Differential privacy in control and network systems. In: IEEE conference on decision and control systems. In: CDC)
- Degue KH, Le Ny J (2017) On differentially private Kalman filtering. In: IEEE global conference on signal and information processing (GlobalSIP)
- Duncan G, Lambert D (1986) Disclosure-limited data dissemination. *J Am Stat Assoc* 81(393):10–28
- Dwork C, Roth A (2014) The algorithmic foundations of differential privacy. *Found Trends Theor Comput Sci* 9(3–4):211–407
- Dwork C, McSherry F, Nissim K, Smith A (2006) Calibrating noise to sensitivity in private data analysis. In: Proceedings of the third theory of cryptography conference, New York
- Hart GW (1992) Nonintrusive appliance load monitoring. *Proc IEEE* 80(12):1870–1891
- Hsu J, Huang Z, Roth A, Wu Z (2016) Jointly private convex programming. In: Proceedings of the 27th annual ACM-SIAM symposium on discrete algorithms (SODA)
- Huang Z, Mitra S, Dullerud G (2012) Differentially private iterative synchronous consensus. In: Proceedings of the 2012 ACM workshop on privacy in the electronic society
- Le Ny J (2013) On differentially private filtering for event streams. In: IEEE conference on decision and control
- Le Ny J (2020) Differential privacy for dynamic data. Springer
- Le Ny J, Pappas GJ (2014) Differentially private filtering. *IEEE Trans Autom Control* 59(2):341–354

- Manitara NE, Hadjicostis CN (2013) Privacy-preserving asymptotic average consensus. In: Proceedings of the European control conference, Zurich
- Mo Y, Murray R (2016) Privacy preserving average consensus. *IEEE Trans Autom Control* 62(2):753–765
- Nozari E, Tallapragada P, Cortés J (2017) Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design. *Automatica* 81:221–231
- Sankar L, Rajagopalan SR, Poor HV (2013) Utility-privacy tradeoffs in databases: an information-theoretic approach. *IEEE Trans Inf Forensics Secur* 8(6):838–852
- Sweeney L (2002) k-anonymity: a model for protecting privacy. *Int J Uncertain Fuzziness Knowl-Based Syst* 10(05):557–570
- Trappe W, Sankar L, Poovendran R, Lee H, Capkun S (2015) Special issue on signal and information processing for privacy. *IEEE J Sel Top Signal Process* 9(7):1173–1175

Programmable Logic Controllers

Georg Frey
Saarland University, Saarbrücken, Germany

Abstract

The programmable logic controller (PLC) is still the most used automation device in industry. This chapter presents the core concepts of the technology, discusses the software model and programming techniques, and describes the underlying execution model.

Keywords

PLC · IEC 61131 · Ladder diagram

Introduction

Since the 1970s, the programmable logic controller (PLC) has been the primary workhorse of industrial automation. For a long time, it has provided a distinct field of research, development, and application, mainly for control engineering. This area has produced its own design methods and programming languages. Due to its importance for industrial application, a lot of these methods have been standardized by

the International Electrotechnical Commission (IEC). Currently the most influential standards are IEC 61131 (John and Tiegelkamp 2010) and IEC 61499 (Vyatkin 2011). While the latter one is dedicated to distributed systems, IEC 61131 covers the PLC as such. This standard consists of several parts. The most important ones are:

Part 1 : General information. This part covers the *CONCEPT* of PLCs. It describes the general idea and typical functionalities, most importantly, the cyclic processing of the application program working on a stored image of the input and output values.

Part 2 : Equipment requirements and tests. Here requirements on the PLC *HARDWARE* (electrical, mechanical, and functional) and corresponding tests are defined.

Part 3 : Programming languages. This is the most important part of the standard. Based on already existing PLC programming languages, a harmonization of the *SOFTWARE* structure was achieved. This includes a general software model together with a set of different standardized programming languages. IEC 61131-3 paved the way from proprietary programming solutions to a set of well-accepted languages, allowing easier training of PLC programmers and – to some extent – the reuse of application solutions on different hardware platforms.

While Part 2 is of importance for PLC manufacturers only, Parts 1 and 3 contain relevant information for PLC users, especially for designers of PLC control applications. Before discussing these points, the definition of PLC from IEC 61131-1 is reproduced and discussed:

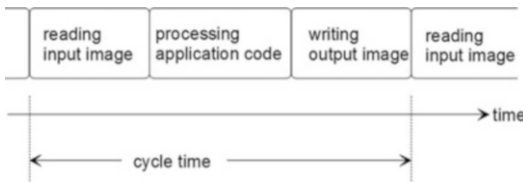
A PLC is a digital electrical system used in manufacturing. It utilizes programmable memory to store practice-oriented control programs. Thus is suitable for implementation of specific functions such as combinatorial control, sequence control, time-, count- and arithmetic functions. Due to its special arrangement of digital or analog input/output, it is used for controlling various machines and processes. (. . .)

This definition is focused on the usage of the device and would – taken out of the context – also cover industrial PCs or microcontroller-based

control solutions. The specifics of PLC hardware are discussed in Part 2 of the standard. However, much more important for distinguishing a PLC from other control hardware are the properties of the execution model described in Part 2 and discussed in the following.

Execution Model

In designing PLC applications, the execution model has to be considered. The main idea is the cyclic execution together with an I/O image. While microcontrollers and PCs typically use an event-based execution model (the application waits for external events from the environment – interrupts – and reacts accordingly), the PLC follows a time-based scheme (the application scans the environment at instances in time, often a fixed cycle time, and reacts on the new status of the input ports).



A PLC cycle consists of three iterated steps: input reading, program execution, and output writing. Together with the concept of the process image – a reserved memory space where input and output variables are stored – this execution model leads to the following:

- (a) During one cycle, input and output values are kept fixed, i.e., a change in input signal values during a cycle will not be seen by the program executed. This means that a temporal change in an input signal value that is shorter than the cycle time may not be registered by the PLC at all.
- (b) Changes in output signal settings by the program will be switched to the actual output ports only after execution of the complete program. This actually means that for an output signal where the value is changed several

times during one program execution, only the last change will be set to the hardware output of the PLC.

- (c) The response time of a PLC, i.e., the time between a change in an input signal and the corresponding reaction at the output port of a PLC, lies between one and two PLC cycles, depending on when the change at the input port occurs relative to the PLC cycle.

While the time needed for input reading and output writing is constant over all cycles, the time for program execution may vary due to conditional execution of some program parts. However, normally the PLC is operated with a fixed cycle time set high enough to allow for the worst-case execution time of the application program.

The advantage of the described concept is the deterministic behavior of the resulting system with a very simple way to determine the timing behavior. This is important for most PLC applications:

- (a) Open-loop control where the reaction to a change of an input signal has to be reached in a limited time, especially in safety-critical applications.
- (b) Closed-loop control where the design of a discrete-time control algorithm is based on the assumption of a fixed sample-time.

To realize control functions, an application program has to be written for the PLC. To this end, Part 3 of the IEC 61131 defines a software model together with a set of programming languages.

Software Model and Programming

The original idea that led to the development of the first programmable logic controller (PLC) in 1968 was to replace hardwired control equipment at machines. Back then, the controllers of machines, for example, lathes or grinders, typically consisted of a cabinet of interconnected relays. The size of such a controller could be considerable, and its failure rate was high due to mechanical defects of single relays. Furthermore, the initial setup was very time-consuming and

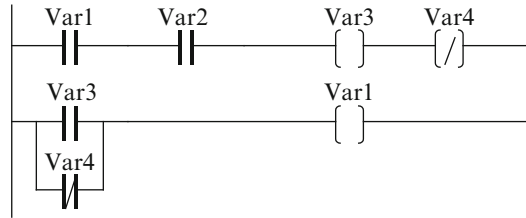
error prone, because the relays (often hundreds of them) had to be wired by hand. The biggest drawback of this technology, however, was the problems arising if a controller had to be changed, employing a new function or adjusting to a new production task. Then the hardwired structure had at least partially to be disassembled and rewired. Here was the main advantage of a controller that could be adjusted by changing software instead of hardware.

Since the first PLCs in the early 1970s reached the market, graphical programming methods are used to develop the control algorithms. These are ladder diagram (LD, sometimes also referred to as ladder logic) and later Function Block Diagram (FBD). The implementation of LD on the very first PLC (the Modicon 084) was intended to allow an easy access for the people doing hardwired relay logic until then. (More on the history of PLCs can be found on the website of Dick Morley, commonly known as the father of the PLC (<http://www.barn.org/FILES/historyofplc.html>).)

LD, at least in its early forms, is basically the graphical representation of its hardwired forefather. The name ladder comes from the fact that on both sides of the drawing, there is a power rail, and horizontally between those rails, like rungs on a ladder, sequences of logical element are drawn. The basic of these elements are relays (switches), depending on input signals or internal variables, and coils (memories to store variables and set output signals). The ladder is processed in a top-down and left-right fashion.

Figure 1 shows an example of an LD. Every rung can be read as an IF THEN ELSE statement. The first rung of the ladder means IF (Var1 = 1 AND Var2 = 1) THEN (Var3 := 1; Var4 := 0) ELSE (Var3 := 0; Var4 := 1). The second rung is IF (Var3 = 1 OR Var4 = 0) THEN (Var1 := 1) ELSE (Var1 := 0).

While LD resembles relay logic, FBD is a graphical mimicking of the wiring of simple logic gates, like AND, OR, NOT, or FLIP-FLOP. Both languages (LD as well as FBD) are still part of the IEC 61131-3. However, they are not well suited for the description of sequential and concurrent algorithms because they have no means for the



Programmable Logic Controllers, Fig. 1 Example of a PLC program written in ladder diagram (LD)

visual description of the control flow in a program.

The IEC 61131-3 standard also contains a language that is intended for the graphical description of sequential and concurrent behavior: the sequential function chart (SFC). The SFC is based on Grafcet (David 1995) and represents a form of Petri net (with very special dynamics and functionality). Due to its high functionality, SFC can be easily applied for the structuring of a PLC program on a high level. However, it is cumbersome (and by the standard also not intended) to use for the specification of a low-level sequential algorithm, as, for example, the alternative switching between two motors.

In addition to the three graphical languages, there are also two textual languages in the standard: the assembler-like Instruction List (IL) and the Pascal-like Structured Text (ST).

The decision for one of the languages is based on functional aspects of the application to be realized (high-level languages SFC and ST vs. low-level languages LD, IL, and FBD) but also on traditions in the application domain (e.g., LD in automotive manufacturing vs. FBD in process industry), the geographical region (e.g., LD in the US vs. IL in Germany), and the preferences of the programmer (graphical vs. textual). To allow for flexible solutions and the optimal choice of languages, IEC 61131-3 allows the use of different languages for different parts of the control application.

An application in IEC 61131-3 is structured into Program Organization Units (POUs). Each of the POUs contains a header in a unified syntax for parameter and variable definitions and a body for the actual program code. This body can be written in any one of the defined PLC languages.

There are three types of POU: Program, Function Block, and Function. A program is the top-level POU of a PLC application. Only in a program, variables can be linked to actual input and output ports. A program can call Function Blocks which in turn may call other function blocks. Programs and Function Blocks can also call Functions. A POU of type function has no internal memory, while a Function Block has memory.

IEC 61131-3 introduced the type and instance concept into PLC programming. A Function Block is always the instantiation of a Function Block Type. Each instantiation gets its own name and variable space. This concept is similar to – but much older than – the class-object instantiation idea of object-oriented programming languages. The exclusive use of symbolic variables without direct references to hardware addresses or ports in Function Blocks allows their easy reuse in one or more applications and the definition of widely applicable Function Block (Type) Libraries.

Summary and Future Directions

PLCs are a proven technology in industrial automation. They follow a simple but deterministic execution and software model. This is the main reason why PLCs are still here and will be here for quite some time to come even if there is faster and fancier technology like embedded PCs available.

The current (third) edition of IEC 61131-3 was published in February 2013. In addition to minor corrections, this new edition adds some concepts from object-oriented programming to the existing software model. First tools on the market already support these extensions.

For the future, two trends can be seen. First, there is a growing trend to integrate PLC programming into model-based software development processes, either by generating PLC code from existing model-based toolchains or by integrating model-based approaches, especially from the object-oriented domain into PLC programming environments. Either way this is due to

the fact that the complexity in PLC application is rising, while the development time should be decreased.

Second, there is a growing interest in the use of formal methods in the PLC domain. In recent years, a lot of interdisciplinary work was aimed in this direction. This work results in the formalization of different steps in the control design process depending on what problems are to be solved (Frey and Litz 2000):

1. The demand for reduced development time and the possible reuse of existing software modules result in the need for a formal approach to the development of the PLC programs.
2. The demand for high-quality solutions and especially the application of PLC in safety-critical processes result in the need for validation procedures, i.e., formal methods to prove specific static and dynamic properties of the programs.
3. The large numbers of already installed PLC programs, together with the high expense of programming, lead to the search for verification and validation methods that can be applied directly to programs written in PLC-specific programming languages such as ladder diagram.

To conclude, more than 50 years after its invention, the PLC is still an industrial success story, and due to ever-increasing demands on the complexity and correctness of its applications, it also still provides much room for further research and development.

Cross-References

- [Applications of Discrete Event Systems](#)
- [Control Hierarchy of Large Processing Plants: An Overview](#)
- [Industrial Cyber-Physical Systems](#)
- [Industrial MPC of Continuous Processes](#)
- [PID Control](#)
- [Real-Time Optimization of Industrial Processes](#)

Bibliography

- David R (1995) Grafcet: a powerful tool for specification of logic controllers. *IEEE Trans Control Syst Technol* 3:253–268
- Frey G, Litz L (2000) Formal methods in PLC programming. In: *Proceedings of the IEEE conference on systems man and cybernetics SMC 2000*, Nashville, Tennessee, pp 2431–2436
- John K, Tiegelkamp M (2010) IEC 61131-3: programming industrial automation system: concepts and programming languages, requirements for programming systems, aids to decision-making tools, 2nd edn. Springer, Berlin
- Vyatkin V (2011) IEC 61499 as enabler of distributed and intelligent automation: state-of-the-art review. *IEEE Trans Ind Inform* 7:768–781

Proportional-Integral-Derivative Control

► PID Control

Pursuit-Evasion Games and Zero-Sum Two-Person Differential Games

Pierre Bernhard
INRIA-Sophia Antipolis Méditerranée, Sophia
Antipolis, France

Abstract

Differential games arose from the investigation, by Rufus Isaacs in the 1950s, of pursuit-evasion problems. In these problems, *closed-loop strategies* are of the essence, although defining what is exactly meant by this phrase, and what is the “Value” of a differential game, is difficult. For closed-loop strategies, there is no such thing as a “two-sided maximum principle,” and one must resort to the analysis of Isaacs’ equation, a Hamilton Jacobi equation. The concept of viscosity solutions of Hamilton-Jacobi equations has helped solve several of these issues.

Keywords

Closed loop strategies · Isaacs’ condition · Viscosity solutions

Historical Perspective

The history of differential games (DG in short) starts with Rufus Isaacs, who coined the phrase in his pioneering work of the early 1950s (Isaacs 1951), which was largely ignored until the publication of his book Isaacs (1965). Through the investigation of particular problems, Isaacs invented by himself (with his own names) the concepts of state and control variables, of feedback, his “tenet of transition” – better known as Bellman’s optimality principle – the (Hamilton-Jacobi-Caratheodory-)Isaacs equation, barriers, some difficult corner conditions (“equivocal lines”), singular arcs (“universal lines”), etc.

Another very early work was Kelendzerize’s chapter “A Pursuit Problem” in the historical book by Pontryagin et al. (1962), but it lacked closed-loop strategies.

John Breakwell and a few followers (Breakwell 1977; Breakwell and Merz 1969) picked up Isaacs’ work where he had left it, still working on particular problems, but adding the power of the computer to analyze the solution of Isaacs’ equation via the structure and singularities of fields of extremal trajectories, while most of the literature concentrated on making precise the concepts of closed-loop strategies and of the Value of the game. Prominent figures in that quest are Krasovskii and Subbotin (1977), Fleming (1961), Friedman (1971), Blaquière et al. (1969), Elliot and Kalton (1972), Emilio and Roxin (1969), and Varaiya and Lin (1969) who together invented the concept of non-anticipative strategies.

The major later innovation was Crandall and Lions’ viscosity solutions of PDEs (Lions 1982; Crandal and Lions 1983) applied to DGs and its Isaacs equation by Evans and Souganidis (1984) and Lions and Souganidis (1985).

We also refer the reader to the entry (Quincampoix 2009) of another Springer Encyclopedia.

General Setup

We shall be interested in (continuous time) two-person zero-sum DGs with complete information, this last phrase meaning that both players know exactly and instantly the state of the system, but (usually) not their opponent's control.

The available space of a short article does not allow us to attempt to give the most general setup of a zero-sum two-person perfect-information differential game. We shall therefore concentrate on a typical class, with a finite dimensional state space, as follows. The data are:

1. A two-player dynamical system with state $x \in \mathbb{R}^n$, control variables $u \in \mathbf{U} \subset \mathbb{R}^\ell$, $v \in \mathbf{V} \subset \mathbb{R}^m$ ($\mathbf{U}$ and $\mathbf{V}$ will often be assumed compact), and its dynamics

$$\dot{x} = f(t, x, u, v), \quad x(t_0) = x_0.$$

$$J(t_0, x_0; u(\cdot), v(\cdot)) = \begin{cases} K(t_1, x(t_1)) + \int_{t_0}^{t_1} L(t, x(t), u(t), v(t)) dt & \text{if } t_1 < \infty, \\ \infty & \text{if } t_1 = \infty. \end{cases}$$

We let

$$G(t_0, x_0; \phi, \psi) := J(t_0, x_0; \Gamma(t_0, x_0; \phi, \psi)).$$

5. A concept of “solution,” where the first player wants to minimize the performance index while the second one wishes to maximize it. (In our choice of definition of J , we have assumed that player one wants over anything else to make the game terminate. If we define J as the integral even for infinite end-time, Isaacs' tenet of transition may not hold.)

If

$$\begin{aligned} & \inf_{\phi \in \Phi} \sup_{\psi \in \Psi} G(t_0, x_0; \phi, \psi) \\ &= \sup_{\psi \in \Psi} \inf_{\phi \in \Phi} G(t_0, x_0; \phi, \psi) = V(t_0, x_0), \end{aligned}$$

then V is called the Value function of the game. Several concepts of upper Value and lower Value may be defined (including the first

Denoting $\mathbf{U}$ and $\mathbf{V}$ the sets of measurable functions from $\mathbb{R}$ to $\mathbf{U}$ and $\mathbf{V}$ respectively, one assumes regularity and growth conditions on f to guarantee existence and uniqueness of the solution $x(\cdot)$ for all (t_0, x_0) and all $(u(\cdot), v(\cdot)) \in \mathbf{U} \times \mathbf{V}$.

2. A termination condition, often given by a target set $\mathcal{T} \in \mathbb{R} \times \mathbb{R}^n$, open or closed according to necessity, defining a final time as $t_1 = \inf\{t \mid (t, x(t)) \in \mathcal{T}\}$. If $\mathcal{T} = \{T\} \times \mathbb{R}^n$, final time is fixed and equal to T . The question of whether there is a finite t_1 is one of central interest in pursuit-evasion games.
3. Sets of admissible closed-loop strategies Φ and Ψ . One should choose them in such a way that replacing (u, v) by a pair $(\phi, \psi) \in \Phi \times \Psi$ in the dynamics always produces a (unique) admissible pair of control functions $(u(\cdot), v(\cdot)) = \Gamma(t_0, x_0; \phi, \psi) \in \mathbf{U} \times \mathbf{V}$.
4. A performance measure, or payoff, typically

and second terms above) that have to coincide for a Value to exist.

Isaacs' Condition In the framework of this short entry, we shall always assume that the game satisfies *Isaacs' condition*. It bears on the *Hamiltonian* $H(t, x, p, u, v) := L(t, x, u, v) + \langle p, f(t, x, u, v) \rangle$ and reads

$$\begin{aligned} & \forall (t, x, p) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n, \\ & \inf_{u \in \mathbf{U}} \sup_{v \in \mathbf{V}} H(t, x, p, u, v) \\ &= \sup_{v \in \mathbf{V}} \inf_{u \in \mathbf{U}} H(t, x, p, u, v). \end{aligned} \quad (1)$$

Strategies and Value

In pursuit-evasion games, the concept of closed-loop strategies is of the essence, and

it is extremely important for all DGs. Yet, allowing state feedback strategies such as $u(t) = \phi(t, x(t))$, $v(t) = \psi(t, x(t))$, poses a difficult problem: what classes Φ and Ψ of functions ϕ and ψ to allow? The notations $\inf_{\phi}$ or $\sup_{\psi}$ have no meaning if one does not answer that question. Experience tells us that discontinuous feedbacks are necessary to find the solution of many examples, but then existence, or uniqueness, of the solution of the dynamical equation cannot be guaranteed.

Isaacs' K-strategies were a partial attempt to address this issue. More developed concepts were proposed, from limit of piecewise constant, or piecewise open-loop, controls (Fleming 1961; Friedman 1971) to extensions of the notion of solution of a differential equation (Krasovskii and Subbotin 1977), also proving the existence of a Value. The equivalence of all these Values was an issue until the advent of viscosity solutions of Isaacs' equation.

A tool used to accommodate state-feedback strategies (Bernhard 1977) is

Lemma 1 (Berkovitz) *If $\mathcal{V} \subset \Psi$, then, $\forall \phi$ for which this expression is well defined,*

$$\sup_{\psi \in \Psi} G(t_0, x_0; \phi, \psi) = \sup_{v(\cdot) \in \mathcal{V}} G(x_0, t_0, \phi, v(\cdot)).$$

As a consequence, a saddle-point (ϕ^*, ψ^*) solution is defined by

$$\begin{aligned} \forall u(\cdot) \in \mathcal{U}, \forall v(\cdot) \in \mathcal{V} \\ G(t_0, x_0; \phi^*, v(\cdot)) &\leq V(t_0, x_0) \\ &\leq G(t_0, x_0; u(\cdot), \psi^*), \end{aligned} \quad (2)$$

confronting the closed-loop saddle point strategies to open-loop controls only. (This proves useful in the analysis of Nash equilibria of nonzero-sum DGs.)

Another consequence of Berkovitz' lemma is that if a DG has a saddle point in open-loop controls, it is a saddle point over closed-loop controls as well. (But the existence condition may be less stringent for the later.) The relationship between different forms of the strategies has been

further clarified by Başar (1977) and Başar and Olsder (1982).

As far as the existence of the Value is concerned, the problem for a large class of DGs is solved with *non-anticipative strategies* defined as $\Phi : \mathcal{V} \rightarrow \mathcal{U}$ such that

$$\begin{aligned} \forall t, [\forall s < t \quad v_1(s) = v_2(s)] &\Rightarrow [\phi(v_1(\cdot))(t) \\ &= \phi(v_2(\cdot))(t)], \end{aligned}$$

and likewise for Ψ (notice that for this concept of strategies, (2) is the natural formulation of a saddle point) and with the notion of *viscosity solution of Isaacs' equation*. See Theorem 1 below.

Games of Pursuit Evasion

An important class of DGs is the game of pursuit evasion. Typically, in these games the state x is composed of a sub-vector y of *Pursuer state(s)* and a sub-vector z of *Evader state(s)*. The dynamical function f is separated likewise, the dynamics of the Pursuer depending on the *Pursuer's control(s)* and that of the Evader on the *Evader's control(s)*. Typically, the payoff is time until *capture* defined as $(t, x(t)) \in \mathcal{T}$ (the target is often called *capture set*). This form of DG automatically satisfies Isaacs' condition (1).

Qualitative Game

In pursuit-evasion games, the main issue is to distinguish initial states, called *capturable*, for which a Pursuer's strategy causing finite-time capture against any defense exists, from those, called *safe*, for which the Evader has a strategy guaranteeing *escape* against any defense. This is the topic of the *qualitative game* or *game of kind* (Isaacs). A *theorem of the alternative* is one which states that for a particular (class of) game(s), every initial state is either capturable or safe. Such theorems have been proved for classes of pursuit-evasion games covering essentially all cases of interest, under Isaacs condition (1) with $L = 0$ (Krasovskii and Subbotin 1977; Cardaliaguet 1996; Cardaliaguet et al. 2001).

Capturable states are separated from safe states by a *barrier*, a piecewise smooth manifold which has to be *semipermeable*. This means that for all (t, x) on the barrier where this barrier is a smooth manifold with normal $\nu(t, x)$, it should hold that

$$\begin{aligned} \min_{u \in U} \max_{v \in V} \langle \nu(t, x), f(t, x, u, v) \rangle \\ = \max_{v \in V} \min_{u \in U} \langle \nu(t, x), f(t, x, u, v) \rangle = 0. \end{aligned}$$

A minimax pair $(u, v) = (\hat{\phi}(t, x, v), \hat{\psi}(t, x, v))$ is called a pair of semipermeable strategies. If the boundary of the capture set is a smooth manifold with local outward normal $n(t, x)$, its *usable part* is the region where $\inf_{u \in U} \sup_{v \in V} \langle n(t, x), f(t, x, u, v) \rangle < 0$. The *natural barrier* is a semipermeable manifold constructed backward from its boundary (the BUP), with n as final ν and using the characteristic equations:

$$\dot{x} = f(x, \hat{\phi}, \hat{\psi}), \quad \dot{v}^t = -v^t \frac{\partial f(t, x, \hat{\phi}, \hat{\psi})}{\partial x}.$$

(These trajectories are *abnormal trajectories* of the calculus of variations). In most examples, only part of the manifold thus constructed is a barrier, and the complete barrier is made of manifolds pieced together according to a junction condition insuring that the corners “do not leak” (Breakwell), analogous to the corner conditions of the next section.

Quantitative Game

The quantitative game, or *game of degree* (Isaacs), is played inside the capture zone, typically with time of capture as the payoff. It is ruled by Isaacs’ equation in a fashion similar to that of games of finite duration (see below). Yet, the interplay between the qualitative and the quantitative game may be quite subtle and plays a prominent role in determining the actual capture zone. The Value function is usually discontinuous across other barriers inside the capture zone.

Other Approaches

Other approaches have been developed to solve games of pursuit evasion.

An early approach by Pontryagin (1967), extended by Pshenichnyi (1968), used geometric methods for linear pursuit-evasion games with convex compact control sets. Krasovskii’s *stable bridges* (Krasovskii and Subbotin 1977) are a concept close to Isaacs’ semipermeability. Patsko and Turova (2001) have developed, for some families of DGs, an efficient numerical procedure to compute recursively hypersurfaces of constant time-to-capture, whose discontinuities display the barriers. Cardaliaguet et al. (1999) have developed a theoretical and numerical procedure building on Aubin’s viability theory, which requires less regularity on the data than other approaches.

Provided that care be applied, a quantitative game may be transformed into a family of qualitative games – an approach used by Krasovskii, Blaqui  re et al., and Cardaliaguet et al. – and conversely, a fruitful approach is to investigate capturability of initial states as a function of a parameter defining the “size” of the capture set, imbedding the qualitative game into a quantitative game of the type *game of approach*.

Games of Finite Duration

Wherever termination of the game is not an issue, the major tool in investigating a DG is Isaacs’ equation, a partial differential equation bearing on the Value function:

$$\begin{aligned} \forall (t, x) \notin \mathcal{T}, \quad \frac{\partial V}{\partial t}(t, x) \\ + \min_{u \in U} \max_{v \in V} H(t, x, \nabla_x V, u, v) = 0, \\ \forall (t, x) \in \mathcal{T}, \quad V(t, x) = K(t, x). \end{aligned} \quad (3)$$

For any DG where all trajectories are transverse to the boundary $\partial \mathcal{T}$, and with adequate regularity conditions on the data (and still under condition (1)), it holds that

Theorem 1 *The DG has a Value in non-anticipative strategies, which is the only bounded,*

uniformly continuous viscosity solution of the equation obtained by changing signs in (3) as $-\partial V/\partial t - \min_{u \in U} \max_{v \in V} H = 0$. And all other Values coincide.

One possible way to solve Isaacs' equation is via the investigation of its field of characteristics. Their equations are Isaacs' *retrograde path equations*: let $(\hat{u}, \hat{v}) = (\hat{\varphi}(t, x, p), \hat{\psi}(t, x, p))$ be the saddle point of $H(t, x, p, u, v)$, assumed here to be unique, one integrates from the target set backward:

$$\dot{x} = f(t, x, \hat{u}, \hat{v}), \quad (4)$$

$$\dot{p} = - \left(\frac{\partial H(t, x, p, \hat{u}, \hat{v})}{\partial x} \right)^t. \quad (5)$$

The above equations are similar to Pontryagin's maximum principle equations. However, a major difference lies in the *corner conditions*. While Pontryagin's theorem extends to control theory the Erdman-Weierstrass condition stating that the adjoint vector (here p) is continuous along an extremal trajectory, in (4) and (5), p is to coincide with $\nabla_x V$ and **may be discontinuous along an extremal trajectory**. These discontinuities cannot be found by a local analysis along an isolated trajectory and require that a complete field of extremals be constructed, synthesizing a state feedback strategy.

The analysis of the conditions that hold at such corners, equivocal manifolds (Isaacs), envelope manifolds (Breakwell), and focal manifolds (Merz), has been a large part of the early Isaacs-Breakwell theory. It has been for its larger part synthesized by Bernhard (1977), except a general constructive analysis of focal manifolds which had to wait until Melikyan and Bernhard (2005).

The absence of a "two-sided Pontryagin principle" for closed-loop differential games forces one to resort to the solution of Isaacs' equation or an equivalent. This is the reason why no practical method of solution exists beyond a state dimension of 3 or 4, counting time if the game is not time invariant. An exception is the linear quadratic game. (See article ► [Linear Quadratic Zero-sum Two-person Differential Games](#) in this encyclopaedia).

Conclusion

Except for very particular games, "solving" a DG remains a difficult task. Numerical methods suffer the famous "curse of dimensionality." Moreover, many of them strive to compute the Value function. But the optimal strategies typically depend on the gradient of the Value function, requiring a stronger convergence of the approximation algorithms than pointwise, or C^0 or L^2 , if they are to be computed as well. Further advances in numerical algorithms tackling this problem would be useful, as well as uncovering new classes of DGs for which further analytical results could be obtained.

Cross-References

- [Dynamic Noncooperative Games](#)
- [Game Theory: A General Introduction and a Historical Overview](#)
- [Linear Quadratic Zero-Sum Two-Person Differential Games](#)

Bibliography

- Başar T (1977) Informationally nonunique equilibrium solutions in differential games. *SIAM J Control Optim* 15:636–660
- Başar T, Olsder GJ (1982) *Dynamic noncooperative game theory*. Academic, London/New York
- Bernhard P (1977) Singular surfaces in differential games, an introduction. In: Hagedorn P, Olsder GJ, Knobloch H (eds) *Differential games and applications. Lecture notes in information and control sciences*, vol 3. Springer, Berlin, pp 1–33
- Blaquière A, Gérard F, Leitmann G (1969) *Quantitative and qualitative games*. Academic, New York
- Breakwell JV, Merz AV (1969) Toward a complete solution of the homicidal chauffeur game. In: Ho Y-C, Leitmann G (eds) *Proceedings of the first international conference on the theory and applications of differential games*, Amherst
- Breakwell JV (1977) Lecture notes. In: Hagedorn P, Knobloch HW, Olsder G-J (eds) *Theory and applications of differential games. Springer lecture notes in control and information sciences*, vol 3. Springer, Berlin, pp 70–95
- Cardaliaguet P (1996) A differential game with two players and one target. *SIAM J Control Optim* 34:1441–1460

- Cardaliaguet P, Quincampoix M, Saint-Pierre P (1999) Set-valued numerical methods for optimal control and differential games. In: Nowak A (ed) *Stochastic and differential games. Theory and numerical methods*. Annals of the international society of dynamic games. Birkhäuser, Boston, pp 177–247
- Cardaliaguet P, Quincampoix M, Saint-Pierre P (2001) Pursuit differential games with state constraints. *SIAM J Control Optim* 39:1615–1632
- Crandal MG, Lions P-L (1983) Viscosity solutions of Hamilton Jacobi equations. *Trans Am Math Soc* 277:1–42
- Elliot RJ, Kalton NJ (1972) The existence of value in differential games of pursuit and evasion. *J Differ Equ* 12:504–523
- Evans LC, Souganidis PE (1984) Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations. *Indiana Univ Math J* 33:773–797
- Fleming WK (1961) The convergence problem for differential games. *Math Anal Appl* 3:102–116
- Friedman A (1971) *Differential games*. Wiley, New York
- Isaacs R (1965) *Differential games, a mathematical theory with applications to optimization, control and warfare*. Wiley, New York
- Isaacs RP (1951) *Games of pursuit*. Technical report P-257, The Rand corporation
- Krasovskii N, Subbotin A (1977) *Jeux différentiels*. MIR, Moscow
- Lions P-L (1982) *Generalized solutions of Hamilton-Jacobi equations*. Pitman, Boston
- Lions P-L, Souganidis PE (1985) Differential games, optimal control, and directional derivatives of viscosity solutions of Bellman's and Isaacs' equations. *SIAM J Control Optim* 23:566–583
- Melikyan A, Bernhard P (2005) Geometry of optimal trajectories around a focal singular surface in differential games. *Appl Math Optim* 52:23–37
- Patsko VS, Turova VL (2001) Level sets of the value function in differential games with the homicidal chauffeur dynamics. *Int Game Theory Rev* 3:67–112
- Pontryagin LS (1967) Linear differential games I and II. *Soviet Math Doklady* 8:769–771, 910–912
- Pontryagin LS, Boltyanskii VG, Gamkrelidze RV, Mishenko EF (1962) *The mathematical theory of optimal processes*. Wiley, New York
- Pshenichnyi BN (1968) Linear differential games. *Autom Remote Control* 29:55–67
- Quincampoix M (2009) Differential games. In: Meyers N (ed) *Encyclopaedia of complexity and system science*. Springer, New York, pp 1948–1956
- Roxin EO (1969) The axiomatic approach in differential games. *J Optim Theory Appl* 3(3):153–163
- Varaiya P, Lin Y (1969) Existence of saddlepoint in differential games. *SIAM J Control* 7: 141–157

Q

Quantitative Feedback Theory

Mario Garcia-Sanz
Case Western Reserve University, Cleveland,
OH, USA

Abstract

Designing reliable and high-performance control systems is an essential priority of every control engineering project. In many practical situations, the presence of model uncertainty challenges the design. A robust control approach for these cases, deeply rooted in the classical frequency domain, is the *Quantitative Feedback Theory* (QFT). As an intrinsically multi-objective methodology, QFT is able to design control solutions that guarantee the achievement of a multi-objective set of performance specifications for every plant within the model uncertainty. It gives the control engineer a physical understanding of the trade-offs, limitations, and possibilities of the control system at every step of the design process, balancing also the trade-off between the simplicity of the controller structure and the minimization of the control activity at each frequency of interest. Control solutions designed by the QFT

methodology reach from simple P, PI, or PID (proportional, integral, derivative) structures to much more complex high order and multi-block/multivariable structures. QFT has a proven track record of success in an extensive variety of real-world control problems, including linear and nonlinear plants, stable and unstable systems, multi-input multi-output processes, minimum and non-minimum phase plants, containing time-delay, lumped and distributed parameters, and advanced control topologies. Recent developments on computer-aided design tools facilitate the use of QFT to the control engineer.

Keywords

Robust control · Quantitative feedback theory · Frequency domain control

Acronym

QFT

Definition

Quantitative Feedback Theory (QFT) is a robust control engineering methodology that uses feed-

back to simultaneously (1) reduce the effects of *plant uncertainty* and (2) satisfy *stability and performance control specifications*. As an intrinsically multi-objective methodology, QFT designs control systems that quantitatively guarantee the satisfaction of all the required control specifications for all the plants within the associated model uncertainty (*robust control*).

QFT is rooted in the classical *frequency domain*. It deals with Bode diagrams and Nichols charts (magnitude/phase diagrams). It relies on the observation that a system needs feedback control when (1) the plant contains model uncertainty and/or (2) external unknown disturbances affect the plant. QFT balances quantitatively (a) the complexity of the controller structure, (b) the minimization of the so-called cost of feedback – controller magnitude at each frequency, (c) the plant model uncertainty, and (d) the achievement of all the required control specifications, all at each frequency of interest. QFT has a proven track record of success in an extensive variety of real-world control problems.

Historical Notes

Hendrik Bode established many of the frequency domain fundamentals in his seminal book *Network Analysis and Feedback Amplifier Design* (Bode 1945). The book strongly influenced the understanding of automatic control theory, especially where system's sensitivity and feedback constraints are concerned.

Almost 20 years later, Isaac Horowitz wrote another very influential book entitled *Synthesis of feedback systems* (Horowitz 1963). The book proposed for the first time a formal combination of frequency-domain control design methodologies and plant model uncertainty (*robust control*) under a quantitative analysis. It addressed numerous sensitivity problems in feedback control, being also the first work in which a control problem was treated quantitatively in a systematic way. The book laid the foundation for a new control design methodology that had been

introduced briefly in a previous paper by Horowitz in 1959. This new methodology became known as *Quantitative Feedback Theory* (or QFT) in the early 1970s.

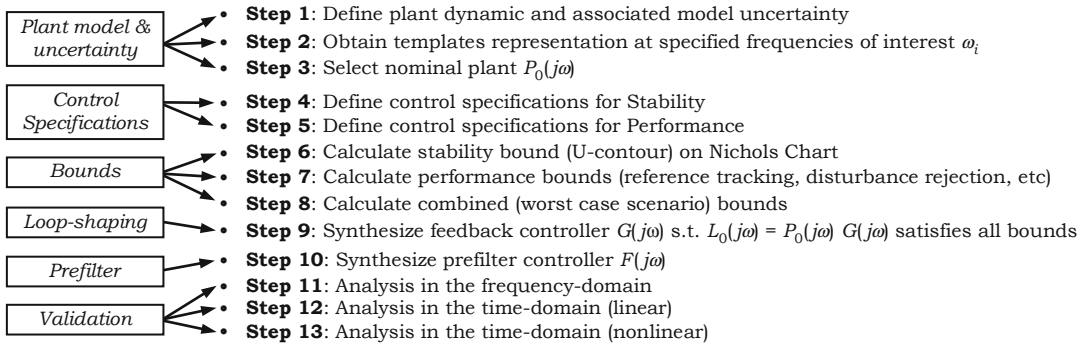
Fundamentals

A detailed study of the *QFT fundamentals and applications* can be found in the books written by Garcia-Sanz (2012, 2017), Houpis (2006), Sidi (2002), Yaniv (1999), and Horowitz (1993) – see recommended reading.

QFT gives the control engineer a comprehensive and physical understanding of the trade-offs, limitations, and possibilities of the control system at every step of the design process. It quantifies the balance among (1) the complexity/simplicity of the controller structure, (2) the multi-objective performance specifications, (3) the plant model uncertainty, and (4) the cost of feedback, all at each frequency of interest. Control solutions designed by the QFT methodology reach from simple P, PI, or PID (proportional, integral, derivative) structures to much more complex high order and multi-block/multivariable structures. The basic steps of the QFT methodology are summarized in Fig. 1 and are introduced in the following subsections.

Define Plant Dynamics, Model Uncertainty, and QFT Templates (Steps 1, 2 and 3)

The QFT methodology starts by modeling the dynamics of the plant and the associated model uncertainty. This step requires an engineering understanding of the system and typically involves multi-physical models, Euler-Lagrange methods, differential equations, state-space representations, and other techniques – see case studies in Garcia-Sanz (2017). Once the mathematical description of the system is obtained, it is translated into a transfer-function representation with block diagrams, parametric/nonparametric uncertainty, and model structure uncertainty. This representation contains the frequency domain



Quantitative Feedback Theory, Fig. 1 Summary of QFT controller design methodology

information ($s \approx j\omega$) and is used to easily calculate the so-called QFT “templates”, which are sets of complex numbers at each frequency of interest $\omega \in [\omega_{\min}, \omega_{\max}]$ rad/second: a projection of the n -dimensional parametric-space through the transfer function(s) onto the Nichols chart.

As an example and for “ $\omega = 1$ rad/s, Fig. 2b represents the QFT template of the 3-parameter plant $P(j\omega) = \exp(-j\omega\tau) / ((j\omega)^2 + 2\zeta\omega_n(j\omega) + \omega_n^2)$, with $\omega_n \in [0.7, 1.2]$, $\tau \in [0, 2]$, and $\zeta = 0.02$.

Each template $\mathfrak{P}(j\omega_i) = \{P(j\omega_i)\}$ represents all the possible plants within the model uncertainty, on the Nichols chart and at a given frequency ω_i (Fig. 2c). A particular case, defined as a set of specific parameters of the n -dimensional parameter space, is arbitrarily selected to define the nominal plant $P_0(j\omega)$, a member of the family of plants within the uncertainty (Fig. 2b).

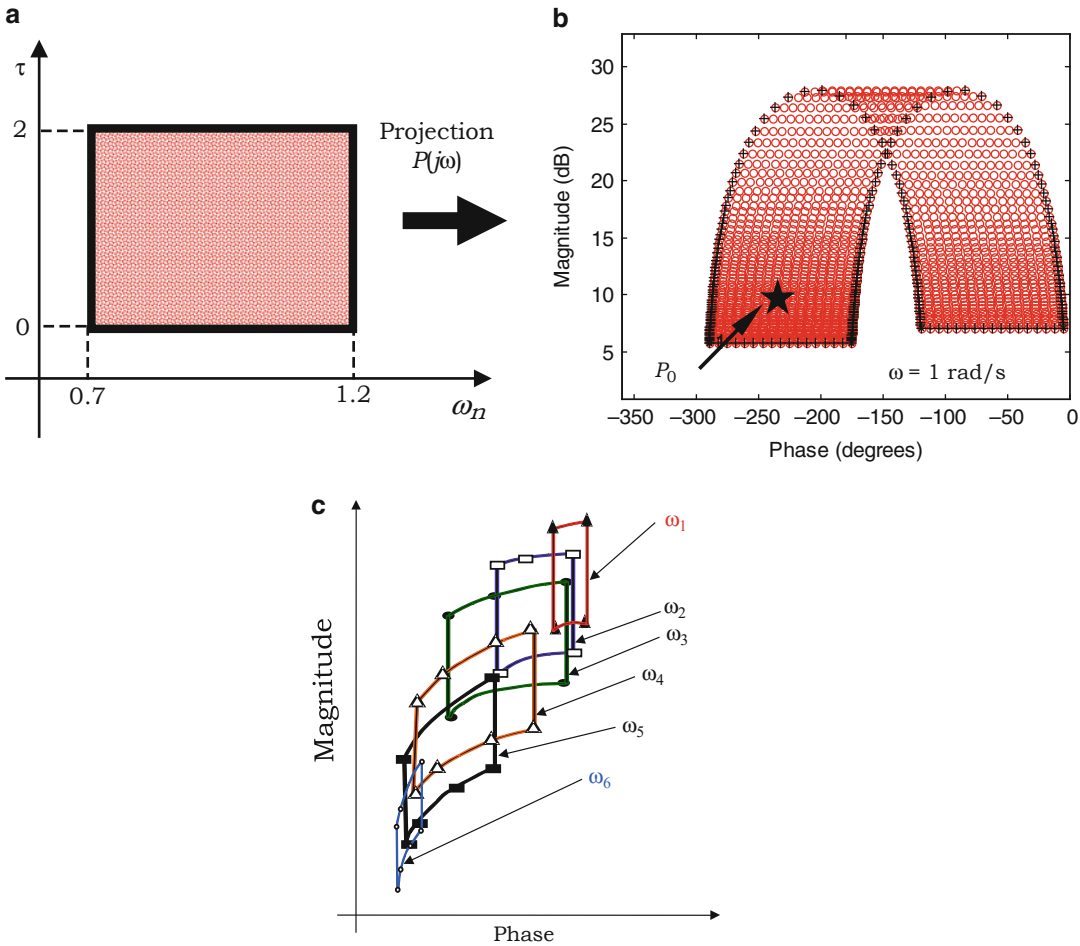
$U(j\omega)$, $Y(j\omega)$, $D(j\omega)$, and $N(j\omega)$ are signals representing, respectively, the reference input to follow, the signal error, controller output, plant output, disturbance input, and sensor noise input. Using a simple block diagram algebra, the following equations are easily calculated (note that the dependency on $s \approx j\omega$ is removed to simplify the notation):

$$\begin{aligned}
 Y &= \frac{P G}{1 + P G H} F R + \frac{M}{1 + P G H} D \\
 &\quad - \frac{P G H}{1 + P G H} N; \\
 U &= \frac{G}{1 + P G H} F R \\
 &\quad - \frac{G H}{1 + P G H} (M D + N) \quad \text{and} \\
 E &= \frac{1}{1 + P G H} F R - \frac{H M}{1 + P G H} D \\
 &\quad - \frac{H}{1 + P G H} N
 \end{aligned}$$

Define Control Specifications (Steps 4 and 5)

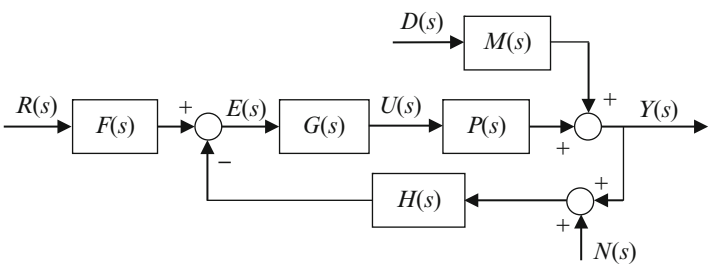
A standard two degree of freedom (2DOF) control system diagram is shown in Fig. 3. It includes the set of uncertain plants $P(j\omega)$ to be controlled, the disturbance dynamics $M(j\omega)$, and the feedback path dynamics $H(j\omega)$. It also shows the loop controller $G(j\omega)$ and the prefilter $F(j\omega)$, both to be designed. In addition, $R(j\omega)$, $E(j\omega)$,

Based on these expressions, the *stability and performance specifications* can now be defined by choosing $\delta_k(\omega)$ functions that limit the magnitude of the transfer functions $T_k(j\omega)$ that appear in these equations, being $|T_k(j\omega)| \leq \delta_k(\omega)$, $k = 1, 2, \dots$, over different frequencies of interest ($\omega \in \Omega_k$) for each specification – see section 2.15 in Garcia-Sanz (2017).



Quantitative Feedback Theory, Fig. 2 From the parameter space to the Nichols chart: (a) two-dimensional parameter space, (b) template of the plant $P(j\omega)$ at $\omega = 1 \text{ rad/s}$, and (c) typical templates for frequencies $\omega \in [\omega_{\min}, \omega_{\max}]$

Quantitative Feedback Theory, Fig. 3
Multi-Input-Single-Output 2DOF feedback control system. $s \approx j\omega$



Stability and noise reduction: $|T_1(j\omega)| = \left| \frac{Y(j\omega)}{R(j\omega)F(j\omega)} \right| = \left| \frac{Y(j\omega)}{N(j\omega)} \right| = \left| \frac{P(j\omega)G(j\omega)}{1+P(j\omega)G(j\omega)} \right| \leq \delta_1(\omega), \quad \omega \in \Omega_1,$

Disturbance rejection: $|T_2(j\omega)| = \left| \frac{Y(j\omega)}{D(j\omega)} \right| = \left| \frac{M(j\omega)}{1+P(j\omega)G(j\omega)} \right| \leq \delta_2(\omega), \quad \omega \in \Omega_2,$

Control effort reduction: $|T_3(j\omega)| = \left| \frac{U(j\omega)}{M(j\omega)D(j\omega)} \right| = \left| \frac{U(j\omega)}{N(j\omega)} \right| = \left| \frac{U(j\omega)}{R(j\omega)F(j\omega)} \right| = \left| \frac{G(j\omega)}{1+P(j\omega)G(j\omega)} \right| \leq \delta_3(\omega), \quad \omega \in \Omega_3$

Reference tracking: $\delta_{4\inf}(\omega) < |T_4(j\omega)| = \left| \frac{Y(j\omega)}{R(j\omega)} \right| = \left| F(j\omega) \frac{P(j\omega)G(j\omega)}{1+P(j\omega)G(j\omega)} \right| \leq \delta_{4\sup}(\omega), \quad \omega \in \Omega_4,$

$$\frac{|G(j\omega)P_d(j\omega)|}{|G(j\omega)P_e(j\omega)|} \frac{|1+G(j\omega)P_e(j\omega)|}{|1+G(j\omega)P_d(j\omega)|} \leq \delta_4(\omega) = \frac{\delta_{4\sup}(\omega)}{\delta_{4\inf}(\omega)}, \quad \omega \in \Omega_4$$

Note that without losing generality and with a straightforward block diagram manipulation, the previous expressions have been rearranged to have a unitary $H(s)$.

QFT Bounds (Steps 6, 7 and 8)

Once the calculations of the previous steps are ready, the QFT methodology combines the stability and performance specifications $\delta_k(\omega)$, the plant model and associated uncertainty, and the selected nominal plant $P_0(j\omega)$ into a set of iso-lines or bounds on the Nichols chart for each frequency of interest (the *Horowitz-Sidi Bounds*).

In particular, the ω_i plant template, $\Im P(j\omega_i) = \{P(j\omega_i)\}$, is approximated by a finite set of plants $\{P_r(j\omega_i), r = 1, 2, \dots\}$, where each plant is expressed in its polar form as $P_r(j\omega_i) = p(\omega_i) \exp(j\theta(\omega_i)) = p \angle \theta$. Similarly, the controller is expressed in its polar form as $G(j\omega_i) = g(\omega_i) \exp(j\phi) = g \angle \phi$, with a controller phase ϕ that varies from -2π to 0 rad. With these representations, and for every frequency ω_i , the control specifications $\{|T_k(j\omega_i)| \leq \delta_k(\omega_i), k = 1, 2, \dots\}$ are translated into a set of quadratic inequalities with the format $I_{\omega_i}^k(p, \theta, \delta_k, \phi) = a g^2 + b g + c \geq 0$, such that,

$$\text{Stability and noise reduction: } p^2 \left(1 - \frac{1}{\delta_1^2}\right) g^2 + 2 p \cos(\phi + \theta) g + 1 \geq 0,$$

$$\text{Disturbance rejection: } p^2 g^2 + 2 p \cos(\phi + \theta) g + \left(1 - \frac{m^2}{\delta_2^2}\right) \geq 0, \text{ with typically } m = p, 1, \text{ or other options,}$$

$$\text{Control effort reduction: } \left(p^2 - \frac{1}{\delta_3^2}\right) g^2 + 2 p \cos(\phi + \theta) g + 1 \geq 0, \text{ and}$$

$$\text{Reference tracking: } p_e^2 p_d^2 \left(1 - \frac{1}{\delta_4^2}\right) g^2 + 2 p_e p_d \left(p_e \cos(\phi + \theta_d) - \frac{p_d}{\delta_4^2} \cos(\phi + \theta_e)\right) g + \left(p_e^2 - \frac{p_d^2}{\delta_4^2}\right) \geq 0$$

Subsequently, with an appropriate algorithm – see section 2.7 in Garcia-Sanz (2017), the above quadratic inequalities are easily translated into a set of curves on the Nichols chart for each frequency of interest and type of specification: the *individual specification*

bounds. Then, the more demanding (worst case) bound, i.e., the most restrictive one at every phase and each frequency of interest, is selected to obtain the intersection of bounds, or the *combined QFT bounds* (see B_i lines in Fig. 4a).

Controller $G(j\omega)$ Design: Loop Shaping (Step 9)

Designing a controller for an infinite number of plants (models with uncertainty) and a set of multi-specification objectives seems to be a very arduous task. However, QFT significantly simplifies this job by combining all these information in a set of simple curves: the QFT bounds. As a result, the designer can use just only a single plant, the already selected nominal plant P_0 , and compare it with the QFT bounds to design the controller.

In this design stage, also called loop shaping, the controller $G(j\omega)$ is synthesized on the Nichols chart by adding poles and zeros until the nominal loop, defined as $L_0(j\omega) = P_0(j\omega)G(j\omega)$, lies near its bounds. This is

achieved when for each frequency of interest, represented by a given color, the corresponding small circle in L_0 lies near its bound, which is denoted by the corresponding B_i line of the same color (Fig. 4a). As mentioned, these QFT bounds represent the plant models with uncertainty and the performance specifications at each frequency. An optimal controller in the sense of QFT will be obtained if $L_0(j\omega)$ (circles) lies exactly on these bounds (B_i lines) at each frequency. Practically speaking, a good design will place $L_0(j\omega)$ above the continuous line B_i bounds and below the dashed line B_i bounds and will have the minimum possible magnitude at every frequency. A general formulation for the controller structure $G(s)$ is expressed by the following transfer function:

$$G(s) = k_G \frac{\prod_{i=1}^{n_{rz}} \left(\frac{s}{z_i} + 1 \right) \prod_{i=1}^{n_{cz}/2} \left(\frac{s^2}{|z_i|^2} + \frac{2 \operatorname{Re}(z_i)}{|z_i|^2} s + 1 \right)}{s^r \prod_{j=1}^{m_{rp}} \left(\frac{s}{p_j} + 1 \right) \prod_{j=1}^{m_{cp}/2} \left(\frac{s^2}{|p_j|^2} + \frac{2 \operatorname{Re}(p_j)}{|p_j|^2} s + 1 \right)}$$

where k_G is the controller gain, z_i is a zero (real or complex) with m_{rz} and m_{cz} the number of real and complex zeroes, respectively, and p_j is a pole (real or complex) with m_{rp} and m_{cp} the number of real and complex poles, respectively (m_{cz} and m_{cp} even numbers). The controller may have also some poles at the origin (integrators), with $r = 0, 1$ or 2 , etc. For practical tips to design the controller (loop shaping), see section 2.15 in Garcia-Sanz (2017).

Prefilter $F(j\omega)$ Design (Step 10)

If the feedback system includes a reference tracking problem, then the best choice is to use a prefilter $F(s)$ – the second degree of freedom. While the feedback controller $G(s)$ reduces the effect of the uncertainty and improves stability, disturbance rejection, and other specifications, the prefilter $F(s)$ is designed to achieve some required reference tracking objectives. Figure 4b shows a typical prefilter design in the Bode diagram. $\delta_{4\sup}(\omega)$ and $\delta_{4\inf}(\omega)$ are the reference tracking

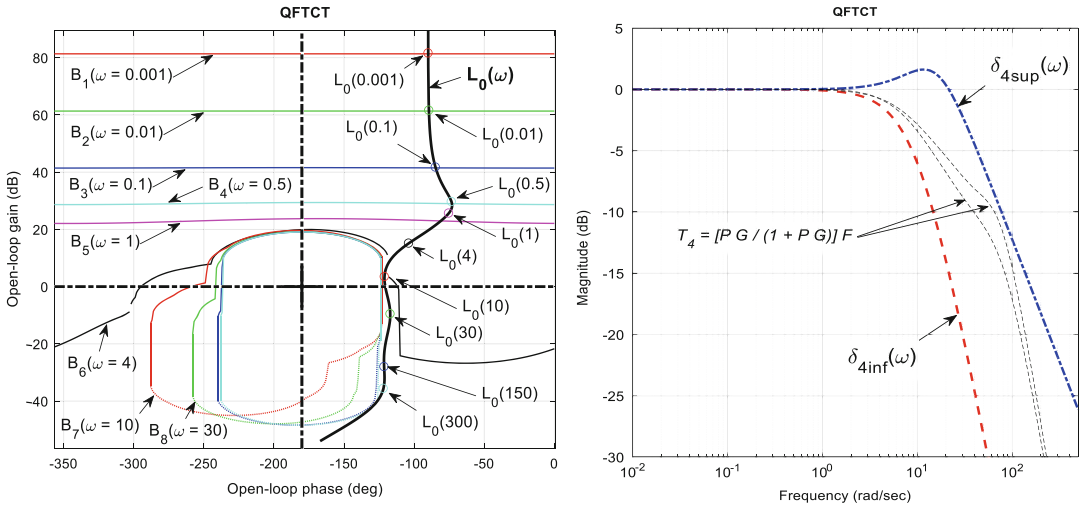
specifications, defined as a band in the frequency domain (outer dashed lines in Fig. 4b), which correspond to upper and lower limits of the plant output in the time domain, $\delta_{4\sup}(t) \geq y(t) \geq \delta_{4\inf}(t)$, when an input reference is applied. In this case, the transfer function T_4 shows an upper and a lower limit (inner gray dashed lines, Fig. 4b) due to the plant uncertainty. After an appropriate prefilter design, the T_4 limits will be inside the $\delta_{4\sup} - \delta_{4\inf}$ band:

$$\delta_{4\inf}(\omega) \leq |T_4| \leq \delta_{4\sup}(\omega),$$

$$\begin{aligned} |T_4(j\omega)| &= \left| \frac{Y(j\omega)}{R(j\omega)} \right| \\ &= \left| \frac{P(j\omega) G(j\omega)}{1 + P(j\omega) G(j\omega)} F(j\omega) \right| \end{aligned}$$

Validation (Steps 11, 12, 13)

Once the design of the controller (and prefilter if needed) is finished, it will be convenient to analyze the performance of the complete control system under different scenarios. This includes



Quantitative Feedback Theory, Fig. 4 (a) QFT-bounds $B_i(\omega)$ and loop shaping $L_0(j\omega) = P_0(j\omega)G(j\omega)$. (b) Prefilter $F(j\omega)$ design: reference tracking specifications

(a) frequency-domain analysis of each specification for all the significant plants within the model uncertainty and (b) time-domain simulations, typically using a Monte Carlo campaign for the uncertainty, first with the linear system and then with nonlinear elements (saturation, etc.).

Programs and Data

Computer-aided design tools facilitate the use of QFT. The most recent QFT control toolbox for Matlab developed by Garcia-Sanz can be found at <http://codypower.com/CDP-QFTCT.htm>. Another popular QFT tool, developed in the 1990s by Borghesani, Chait, and Yaniv, can be found at <http://www.terasoft.com/products/QFT/index.html>.

Applications and Future Directions

- QFT has been successfully applied to an extensive variety of control problems, including stable and unstable plants minimum and non-minimum phase systems, single-input single-output and multiple-input multiple-output processes, with linear and nonlinear characteristics, long time delay,

$\delta_{4sup}(\omega)$ and $\delta_{4inf}(\omega)$, and upper and lower limits of T_4 due to the plant uncertainty: $\delta_{4inf}(\omega) \leq |T_4| \leq \delta_{4sup}(\omega)$

distributed parameter systems, and time-varying plants, and has been combined with feedforward control topologies, multi-loop systems, etc. In addition, QFT has been used in many real-world applications, including flight control, wind energy, water treatment plants, spacecraft, power systems, mechanical systems, motion control, chemical reactors, etc. – see Garcia-Sanz (2017).

- Future research on QFT includes new multiple-input multiple-output techniques, nonlinear control, distributed parameter systems, load-sharing control, multi-agent systems, etc.

Cross-References

- [Frequency-Response and Frequency-Domain Models](#)
- [Robust Control in Gap Metric](#)

Bibliography

- Bode H (1945) Network Analysis and Feedback Amplifier Design. Van Nostrand, New York, NY
- Garcia-Sanz M (2017) Robust control engineering: practical QFT solutions. CRC Press/Taylor & Francis, Boca Raton, FL

- Garcia-Sanz M (2008–2019) The QFT control toolbox for Matlab (QFTCT). http://codypower.com/CDP_QFTCT.htm
- Garcia-Sanz M, Houpis CH (2012) Wind energy systems: control engineering design. Part I: QFT control. Part II: Wind turbines control with QFT. CRC Press/Taylor & Francis, Boca Raton, FL
- Horowitz I (1963) Synthesis of feedback systems. Academic Press, New York, NY
- Horowitz I (1993) Quantitative feedback design theory (QFT). QFT Pub, Denver, CO
- Houpis CH, Rasmussen SJ, Garcia-Sanz M (2006) Quantitative feedback theory: fundamentals and applications, 2nd edn. CRC Press/Taylor & Francis, Boca Raton, FL
- Sidi M (2002) Design of robust control systems: from classical to modern practical approaches. Krieger Publishing, Malabar, FL
- Yaniv O (1999) Quantitative feedback design of linear and non-linear control systems. Kluwer Academic Publishers, Boston, MA

Quantized Control and Data Rate Constraints

Girish N. Nair

Department of Electrical and Electronic Engineering, University of Melbourne, Melbourne, VIC, Australia

Abstract

This entry briefly describes the topic of quantized control with limited data rates. The focus is on the problem of stabilizing a linear time-invariant plant over a digital channel and the associated *data rate theorems*. It is shown that the deepest results in this area require a unified treatment of its communications and control aspects.

Keywords

Quantized control · Quantization · Control under communication constraints

Supported by the Australian Government via grant AUSMURIB000001 associated with ONR MURI grant N00014-19-1-2571, and via Australian Research Council grant FT140100527.

Introduction

One of the standard assumptions of classical control theory is that the signals sent from sensors to controllers and from controllers to actuators take continuous values with infinite precision. The advent of computer-based and digitally networked control systems challenged this assumption, since the analog plant outputs or control variables in such systems must be reduced to finite bit strings or discrete symbols for storage, manipulation, and transmission. This process of converting a continuous-valued variable into a finite-valued one is called *quantization* and entails a potentially significant loss of resolution and closed-loop performance. Quantized control is concerned with the analysis and design of control systems which feature such analog-to-digital conversions in the feedback loop.

There is a vast literature on this topic, and the aim of this entry is to briefly explain some of its key ideas. For reasons of space, the discussion is largely confined to the question of how to stabilize a linear time-invariant plant over a digital channel. It is shown that the deepest results here emerge from treating the communications and control aspects jointly, instead of separately. The reader is referred to the survey (Nair et al. 2007) and the references therein for a discussion of other issues such as optimality and transient performance.

Quantization

Quantization has long been an object of study in communications and information theory – see Gersho and Gray (1993) and the references therein. In its simplest form, a signal $x(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^n$ is first *sampled* at regular time intervals $t = 0, \tau, 2\tau, \dots$ to yield a discrete-time signal $(x(k\tau))_{k \in \mathbb{Z}}$, with the sampling frequency $1/\tau$ chosen to be greater than the Nyquist frequency of x (i.e., twice its bandwidth). Each sample $x_k := x(k\tau)$ is then passed through a static, memoryless quantizer Q to yield a quantized discrete-time signal

$$x_k^q = Q(x_k) \in \{q^1, \dots, q^M\} \subset \mathbb{R}^n, \quad k \in \mathbb{Z}_{\geq 0}. \quad (1)$$

which can take M distinct values in $\mathbb{R}^n$. If the quantizer is known to both transmitter and receiver, each of these M values can be represented by a binary string with $\lceil \log_2 M \rceil$ bits. When the input dimension $n = 1$, the quantizer is called scalar; otherwise, it is a vector quantizer. The regions $R^i := Q^{-1}(q^i)$, $1 \leq i \leq M$, are called the quantizer cells and together form a partition of $\mathbb{R}^n$. Thus an M -valued quantizer is fully defined by its quantizer cells R^i and associated quantizer points q^i , $1 \leq i \leq M$.

The quantization error or *quantizer noise* is defined as $n_k := x_k^q - x_k$. When the inputs x_k are identically distributed random variables (rv's), then a standard goal is to design Q so as to minimize the mean-square quantizer noise

$$D := E[||Q(x_k) - x_k||^2], \quad (2)$$

where $E[\cdot]$ is the expectation functional. This yields an optimal quantizer Q_* with cells that satisfy the nearest-neighbor property, i.e.,

$$x \in R_*^i \Rightarrow ||Q_*(x) - q_*^i|| \leq ||Q_*(x) - q_*^j||, \quad \forall j \neq i.$$

When $||\cdot||$ is the Euclidean norm (possibly weighted), the quantizer cells R_*^i , $1 \leq i \leq m$, are convex polygons and form a *Voronoi partition* of $\mathbb{R}^n$, and furthermore q_*^i is the centroid of R_*^i with respect to the stationary distribution F_X of x_k , i.e., $q_*^i = E[x_k | x_k \in R_*^i]$. As a consequence, the optimal quantizer is statistically unbiased, i.e., $E[n_k] = 0$, and furthermore x_k^q and the quantizer noise n_k are uncorrelated at time k , i.e., $E[x_k n_k^T] = 0$. However, note that n_k and x_k^q may be correlated for $j \neq k$, and (n_k) may itself be a correlated process.

If M is large (i.e., the quantizer is *high resolution* or *fine*), then each region R_i will typically be very small, so that f_X will not vary much on each R^i . This yields a conditional pdf of x_k given R_i that is approximately uniform on R_i . For the case when X is scalar-valued and Q is a uniform, midpoint quantizer on an interval $[a, b]$, these considerations yield the asymptotic formula

$$D \approx (b - a)^2 / (12M^2), \quad (3)$$

provided that the overload regions – i.e., the tails of $f_X(x)$ on the regions $x < a$ or $x > b$ – make negligible contributions to D . Note that this expression does not depend on the distribution of the input.

For large M , it can also be shown that the optimal vector quantizer has a normalized point density (proportion of points per unit volume) proportional to $f_X^{1/3}$ and an optimal distortion

$$D_{\min} \approx \frac{c}{M^2} \left(\int f_X(x)^{1/3} d\mu(x) \right)^3, \quad (4)$$

where the constant c depends only on n .

Quantized Control: Basic Formulation

Much of the theory of quantized control concerns finite-dimensional linear time-invariant (LTI) plants. A formulation is provided in this section to help fix ideas, for the case of a single feedback loop containing a single errorless digital channel.

Consider the discrete-time plant

$$x_{k+1} = Ax_k + Bu_k + v_k, \quad y_k = Fx_k + w_k, \quad (5)$$

where at every time $k \in \mathbb{Z}_{\geq 0}$, $x_k \in \mathbb{R}^n$ is the state with x_0 unknown, $u_k \in \mathbb{R}^m$ is the control input, $y_k \in \mathbb{R}^p$ is the measured output, $v_k \in \mathbb{R}^n$ is unknown process noise, $w_k \in \mathbb{R}^p$ is unknown measurement noise, and A , B , and F are constant known matrices of appropriate dimensions. For the problem to be well posed, assume that the matrix pairs (A, B) and (F, A) are, respectively, reachable and observable. Suppose that the output sensors communicate with the controller over a digital channel that can carry one symbol s_k from a finite, possibly time-varying alphabet $\mathcal{S}_k$ of cardinality $M_k \geq 1$ during the $(k + 1)$ -th sampling interval. Assume for simplicity that the channel is errorless, with negligible propagation delay. The asymptotic average rate at which the channel transports data may then be defined as

$$R := \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{j=0}^{k-1} \log_2 M_j \text{ (bits/sample)}. \quad (6)$$

Note that if the channel alphabet $\mathcal{S}_k$ is constant or varies periodically with k , the inferior limit reduces to a straight limit.

In full generality, each transmitted symbol may depend on all past and present measurements and past symbols,

$$s_k = \gamma_k(y_0^k, s_0^{k-1}) \in \mathcal{S}_k, \quad \forall k \in \mathbb{Z}_{\geq 0}, \quad (7)$$

where γ_k is the coder mapping at time k . At time k the controller has $s_0, \dots, s_k$ available and then applies a control law of the general form

$$u_k = \delta_k(s_0^k) \in \mathbb{R}^m, \quad \forall k \in \mathbb{Z}_{\geq 0}, \quad (8)$$

where δ_k is the controller mapping at time k .

In practice, additional memory or structural constraints are usually placed on the general coding and control rules (7) and (8). For instance, if a static quantizer of the form (1) is used, then the coding alphabet $\mathcal{S}_k \equiv \mathcal{S}$ will be constant, and $s_k \equiv \gamma(y_k)$ will represent the index of the quantizer cell that contains y_k . Similarly, a static, memoryless controller is captured by setting $u_k = \delta(s_k)$ in (8).

Finite-dimensional coding and control laws may be formulated by defining internal coder and controller states ψ_k^γ and ψ_k^δ with local updates of the form

$$s_k = \gamma(y_k, \psi_{k-1}^\gamma), \quad \psi_k^\gamma = \phi(s_k, \psi_{k-1}^\gamma), \quad (9)$$

$$\psi_k^\delta = \eta(s_k, \psi_{k-1}^\delta), \quad u_k = \delta(\psi_k^\delta). \quad (10)$$

If the states ψ_k^γ and ψ_k^δ are finite valued, then the coding and control laws are called *finite-state*.

Additive Noise Model

Early approaches to quantized control modeled quantization errors as additive uncorrelated noise, in order to allow the use of well-developed tools from linear stochastic control (Curry

1970). While this is reasonable at high quantizer resolution, it fails to capture two key properties.

A simple example illustrates this. Consider a scalar, noiseless, fully observed, unstable LTI plant – i.e., (5) with $n = 1$, $A = a$ with $|a| > 1$, $B, C = 1$, and $w_k, v_k = 0$ – where x_0 is a random variable. Under static, high-resolution uniform quantization, the data available to the controller is expressed as a noisy linear measurement

$$y'_k := Q(x_k) = x_k + n_k, \quad k \in \mathbb{Z}_{\geq 0},$$

where the quantizer error process (n_k) is treated as zero mean white noise uncorrelated with (x_k) and having constant variance given by (3).

The first shortcoming of the additive noise approach is that it precludes the possibility of asymptotic mean-square stability, which would effectively require the controller to estimate the initial state x_0 with a mean-square error diminishing strictly faster than a^{-2k} . This turns out to be impossible under the uncorrelatedness assumption and the constraint $|a| > 1$.

However, in the seminal paper (Delchamps 1990), it was shown that asymptotic stability could in fact be achieved, by using a nonlinear controller that exploited the correlation between successive quantizer errors. To see this, suppose that the unknown initial state x_0 is confined to a known interval $[-l_0, l_0]$. At time $k \geq 0$, suppose that $l_k \geq 0$, $k = 1, 2, \dots$ represent bounds to be determined on the future states x_k . Let Q be a static one-bit quantizer – i.e., with $M = 2$ – such that $Q(x) = 1$ if $x \geq 0$ and $Q(x) = -1$ if $x < 0$. At time k let $u_k = -0.5al_k Q(x)$ so that

$$x_{k+1} = \begin{cases} a(x_k - 0.5l_k) & \text{if } 0 \leq x_k \leq l_k \\ a(x_k + 0.5l_k) & \text{if } -l_k \leq x_k < 0 \end{cases}.$$

$$\Rightarrow |x_{k+1}| \leq 0.5|a|l_k =: l_{k+1}.$$

If $|a| < 2$ then $l_k \rightarrow 0$, and asymptotic stability is achieved uniformly and with exponential convergence.

The second, and main, drawback of the additive white noise model is that it does not predict the loss of closed-loop stability when the quantizer resolution is too coarse. This is because the number M of quantizer points only serves

to determine the variance of the additive noise n_k : reducing M increases the variance of n_k and the mean-square states, but they remain bounded over time. In contrast, a rigorous analysis reveals that stability is impossible by any means, linear or nonlinear, when M drops below a certain threshold.

Numerous proofs of this loss of stability exist. In a stochastic setting, the argument is based on fixing the coder and controller and expanding out the closed-loop dynamics of the scalar LTI plant to write

$$x_k = a^k x_0 - a^k z_k \quad (11)$$

where $z_k := -a^{-k} \sum_{j=0}^{k-1} a^{k-j-1} u_j$. As z_k is a function of $s_0^{k-1} \in \mathcal{S}^k$, it can take at most M^k values. Furthermore, in the absence of noise, it is fully determined by x_0 , for a given coding and control policy (7) and (8). Thus z_k can be regarded as the output $Q'_k(x_0)$ of an M^k -valued quantizer. Substituting this into (11) yields

$$x_k = a^k (x_0 - Q'_k(x_0)).$$

From the asymptotic quantizer result (4), it then follows that for large k ,

$$\mathbb{E}[x_k^2] \geq c \frac{a^{2k}}{M^{2k}} \left(\int f_{X_0}(x)^{1/3} d\mu(x) \right)^3.$$

Thus a necessary condition for asymptotic mean-square stabilizability is that $M > |a|$ – see Nair and Evans (2000) for details.

The Data Rate Theorem

The discussions above emphasized the need for a more rigorous approach to quantized control. In the literature, the necessary condition $M > |a|$ was first derived in a nonrandom setting, where it was shown to be both sufficient and necessary to be able to ensure uniform stability (Wong and Brockett 1999; Baillieul 1999).

The sufficiency argument is constructive. Let Q be an M -level uniform quantizer on $[-1, 1]$, with cells formed by partitioning $[-1, 1]$ into M subintervals $R^1, \dots, R^M$ of equal length and

setting $Q(z)$ to be the midpoint of R^i when $z \in R^i$. Suppose that at time k , the unknown state x_k lies in a known interval $[-l, l]$, and set $u_k = -alQ(x_k/l)$. Thus

$$\begin{aligned} |x_{k+1}| &= |a||x_k - lQ(x_k/l)| \\ &= |a|l \left| \frac{x}{l} - Q\left(\frac{x}{l}\right) \right| \leq |a| \frac{l}{M}. \end{aligned}$$

When $M > |a|$, the right-hand side $< l$. Thus $x_{k+1} \in [-l, l]$ as well, and boundedness is achieved. Uniform asymptotic stability can be achieved by replacing the constant parameter l in the argument above with a time-varying bound l_k , updated as $l_{k+1} = |a|l_k/M \rightarrow 0$.

The necessity argument is based on volume partitioning. The basic idea is to fix an arbitrary coding and control policy and let m_k be the Lebesgue measure of the set of values that x_k can take at time $k \in \mathbb{Z}_{\geq 0}$. After k time steps, the plant dynamics expand this uncertainty volume m_0 by a factor $|a|^k$. However, the coder effectively divides this region into M^k disjoint, exhaustive pieces, each of which is shifted by the controller. As Lebesgue measure is translation invariant, it then follows that $m_k \geq |a|^k m_0 / M^k$. Consequently M must exceed $|a|$ if the closed loop is uniformly asymptotically stable.

The tight criterion $M > |a|$, or equivalently $R > \log_2 |a|$, was the first instance of the *data rate theorem*. Volume-partitioning arguments and Jordan canonical forms can be used to generalize it to LTI plants with vector-valued states, yielding the necessary and sufficient condition

$$R > \sum_{i: |\lambda_i| \geq 1} \log_2 |\lambda_i| =: H, \quad (12)$$

Where $\lambda_1, \dots, \lambda_n$ are the eigenvalues of A . This criterion is remarkably universal, having been shown to be tight for a variety of settings and objectives: e.g., for asymptotic r -th moment stabilizability with random, unbounded x_0 and no process or measurement noise (Nair and Evans 2003); uniform stabilizability with bounded x_0 and no process or measurement noise (Baillieul 2002); and uniform stabilizability with bounded initial state, process, and measurement noise

(Hespanha et al. 2002; Tatikonda and Mitter 2004).

When the plant noise is random and unbounded, mean-square stability can be achieved using a time-varying bit rate with average value arbitrarily close to but strictly exceeding H (Nair and Evans 2004). If in addition the bit rate is constrained to be constant, an interesting subtlety arises, and the minimum rate is given by $R_{\min} = \log_2 \left[1 + \prod_{i: |\lambda_i| \geq 1} |\lambda_i| \right]$ (Kostina et al. 2018).

The deep nature of (12) becomes even clearer when it is noted that the right-hand side of (12) coincides with the intrinsic entropy generation rate H of the (open-loop) plant, in both the Kolmogorov-Sinai and topological senses; that is, it describes the growth rate of the number of “distinguishable” state trajectories. Thus the data rate theorem states that stability is possible if the communication rate in the feedback loop exceeds the rate at which the plant generates uncertainty. This interpretation leads to the notion of *feedback entropy* (see cross-reference to article by C. Kawan).

Zooming Quantized Control

When the plant noise and initial state of the plant (5) are bounded, stability (in a uniform sense) can be guaranteed by applying a linear observer to track the plant states with bounded error and then applying a suitable static, memoryless coding and control policy on the observer states x_k^o .

However, if the noise or initial state has unbounded support – e.g., when they are Gaussian or when prior bounds on them are not known – then stability cannot be achieved by any such static memoryless scheme or indeed by any scheme where the control inputs (8) are bounded (Nair and Evans 2004). The explanation is simple: due to the infinite support, there is a nonzero probability that the propagated state Ax_t will be beyond reach of the control input at some time t . The unstable plant dynamics then amplify this shortfall, causing the same phenomenon to

occur with increasing probability at subsequent times and inevitably leading to instability.

One solution is to use a *zooming quantizer*, i.e., having a dynamic range $l_k > 0$ that is not bounded a priori but expands or contracts according to the most recent symbol (Brockett and Liberzon 2000). In the noiseless case, if this symbol corresponds to the “overload region” of the quantizer (as indicated by a special symbol), then the range is updated as $l_{k+1} := \phi_{\text{out}} l_k$, where $\phi_{\text{out}} > 1$ is the “zoom-out” factor. Otherwise $l_{k+1} := \phi_{\text{in}} l_k$, where $\phi_{\text{in}} < 1$ is the “zoom-in” coefficient.

In the communications literature, such schemes are called *adaptive quantizers* (Goodman and Gersho 1974). If ϕ_{out} is sufficiently large compared to the unstable open-loop eigenvalues, and if ϕ_{in} is not too small, then global asymptotic stability ensues. With unbounded noise in the plant, variants of this scheme guarantee mean-square stability at any data rate satisfying (12) (Nair and Evans 2004); other forms of stochastic stability (Yüksel 2010); or *input-to-state stability* (Liberzon and Nesic 2007).

Zooming quantization is an important example of a finite-dimensional coder-controller (9) and (10), with l_k playing the role of an internal state variable. As the range update is driven by the symbols, both coder and controller can each generate identical copies of l_k , provided that there are no errors in the channel and they both start from the same initial range l_0 .

Erroneous Digital Channels

The information-theoretic aspects of quantized control become especially pronounced when the channel is not error-free. In this case, the data rate theorem (12) can be extended, but in ways that are highly dependent on the precise setting and stability objective.

A common figure of merit for a stochastic discrete memoryless channel (DMC) is its *ordinary capacity* C . This is defined operationally as the largest block-code bit rate that can be transmitted across the channel with negligible probability of decoding error and also coincides with the largest

rate of Shannon information across the channel (Shannon 1948).

For a noiseless LTI plant with random initial state controlled over a DMC, the condition $C > H$ is a tight criterion for almost sure (a.s.) asymptotic stabilizability (Matveev and Savkin 2007a). This is a natural generalization of (12).

On the other hand, if the objective is to bound the state moments of a scalar LTI plant subject to bounded process noise, then the achievability of this goal is determined by the *anytime capacity* C_{any} (Sahai and Mitter 2006): this is essentially given by the fastest decay rate of the decoding error probability.

However, if the aim is a.s. boundedness of an LTI plant with random initial state and bounded, nonstochastic process noise, then the stabilizability criterion changes again to $C_{0f} > H$ (Matveev and Savkin 2007b). Here C_{0f} is the *zero-error feedback capacity* of the channel, defined as the largest block-code bit rate that can be transmitted across the channel with *exactly zero* probability of decoding error and with perfect channel feedback (Shannon 1956).

As $C_{0f} < C_{\text{any}} < C$ for most channels, these conditions do not coincide. This suggests that there is no universal, operationally relevant information theory for feedback control over error-prone channels: such a theory must instead be tailored to match the underlying objectives and assumptions. For systems with nonstochastic disturbances, preliminary steps in this direction have been taken in Nair (2015, 2013). The reader is also referred to You and Xie (2011) and Minero et al. (2013) for information-theoretic analyses of stochastic linear systems controlled via Markov channels.

Summary and Future Directions

This entry described the key elements of quantized control with finite data rates, emphasizing the interplay between coding and control. A great deal is now known about the fundamental limitations on stability in quantized control systems consisting a single feedback loop. Two major directions for future research suggest themselves:

- Little work has been done on designing optimal coding and control schemes or determining optimal performance at a given rate, apart from one or two special cases and structural results – see Nair et al. (2006, 2007) and the references therein. Lower and upper bounds on optimal performance have recently been derived in Kostina and Hassibi (2019). It is very unlikely that exact, closed-form solutions will be possible. However, numerical approaches based on the Lloyd-Max algorithm, particle filtering, and model-predictive control may prove fruitful.
- Networked control systems usually consist of a number of subsystems interconnected over a network. Furthermore, in multi-agent systems, the main objective may not be stability, but rather coordination or consensus to a common state. Comparatively little is known about the data rate requirements and information-theoretic aspects of these problems.

Cross-References

- [Data Rate of Nonlinear Control Systems and Feedback Entropy](#)

Bibliography

- Baillieul J (1999) Feedback designs for controlling device arrays with communication channel bandwidth constraints. In: ARO workshop on smart structures, Pennsylvania State University
- Baillieul J (2002) Feedback designs in information-based control. In: Pasik-Duncan B (ed) Stochastic theory and control: proceedings of a workshop held in Lawrence, Kansas. Springer, pp 35–57
- Brockett RW, Liberzon D (2000) Quantized feedback stabilization of linear systems. *IEEE Trans Autom Control* 45(7):1279–1289
- Curry RE (1970) Estimation and control with quantized measurements. MIT, Cambridge
- Delchamps DF (1990) Stabilizing a linear system with quantized state feedback. *IEEE Trans Autom Control* 35:916–924
- Gershon A, Gray RM (1993) Vector quantization and signal compression. Kluwer, Boston
- Goodman DJ, Gershon A (1974) Theory of an adaptive quantizer. *IEEE Trans Commun* 22:1037–1045

- Hespanha J, Ortega A, Vasudevan L (2002) Towards the control of linear systems with minimum bit-rate. In: Proceedings of 15th international symposium on mathematical theory of networks and systems (MTNS), U. Notre Dame
- Kostina V, Hassibi B (2019) Rate-cost tradeoffs in control. *IEEE Trans Autom Contr* 64(11):4525–4540
- Kostina V, Peres Y, Ranade G, Sellke M (2018) Exact minimum number of bits to stabilize a linear system. In: Proceedings of IEEE conference on decision and control, Miami Beach, pp 453–458. Full version at <https://arxiv.org/abs/1807.07686v1>
- Liberzon D, Nesic D (2007) Input-to-state stabilization of linear systems with quantized state measurements. *IEEE Trans Autom Control* 52:767–781
- Matveev AS, Savkin AV (2007a) An analogue of Shannon information theory for detection and stabilization via noisy discrete communication channels. *SIAM J Control Optim* 46(4):1323–1367
- Matveev AS, Savkin AV (2007b) Shannon zero error capacity in the problems of state estimation and stabilization via noisy communication channels. *Int J Control* 80:241–255
- Minero P, Coviello L, Franceschetti M (2013) Stabilization over Markov feedback channels: the general case. *IEEE Trans Autom Control* 58(2):349–362
- Nair GN (2013) A nonstochastic information theory for communication and state estimation. *IEEE Trans Autom Control* 58(6):1497–1510
- Nair GN (2015) Nonstochastic information concepts for estimation and control. In: Proceedings of IEEE conference on decision and control, Osaka, pp 45–56
- Nair GN, Evans RJ (2000) Stabilization with data-rate-limited feedback: tightest attainable bounds. *Syst Control Lett* 41(1):49–56
- Nair GN, Evans RJ (2003) Exponential stabilisability of finite-dimensional linear systems with limited data rates. *Automatica* 39:585–593
- Nair GN, Evans RJ (2004) Stabilizability of stochastic linear systems with finite feedback data rates. *SIAM J Control Optim* 43(2):413–436
- Nair GN, Huang M, Evans RJ (2006) Optimal infinite horizon control under a low data rate. In: Proceedings of 14th IFAC symposium on identification and system parameter identification (SYSID), Newcastle, pp 1115–1120
- Nair GN, Fagnani F, Zampieri S, Evans RJ (2007) Feedback control under data rate constraints: an overview. *Proc IEEE* 95(1):108–137. In special issue on Technology of Networked Control Systems
- Sahai A, Mitter S (2006) The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link part I: scalar systems. *IEEE Trans Inf Theory* 52(8):3369–3395
- Shannon CE (1948) A mathematical theory of communication. *Bell Syst Tech J* 27:379–423, 623–656
- Shannon CE (1956) The zero-error capacity of a noisy channel. *IRE Trans Inf Theory* 2:8–19
- Tatikonda S, Mitter S (2004) Control under communication constraints. *IEEE Trans Autom Control* 49(7):1056–1068

Wong WS, Brockett RW (1999) Systems with finite communication bandwidth constraints II: stabilization with limited information feedback. *IEEE Trans Autom Control* 44:1049–1053

You K, Xie L (2011) Minimum data rate for mean square stabilizability of linear systems with Markovian packet losses. *IEEE Trans Autom Control* 56(4):772–785

Yüksel S (2010) Stochastic stabilization of noisy linear systems with fixed-rate limited feedback. *IEEE Trans Autom Control* 55(12):2847–2853

Quantum Networks

Matthew R. James

Research School of Electrical, Energy and Materials Engineering, Australian National University, Canberra, Australia

Abstract

In this entry we discuss a symbolic tool for describing the interconnection of open quantum systems using boson fields. The tool called the series product is expressed in terms of physical parameters. Boson fields, such as free quantum optical fields, serve as “wires” connecting components. The framework is general and allows for the description of networks consisting of quantum and classical components in a consistent and unified manner.

Keywords

Quantum · Networks · Feedback · Control

Introduction

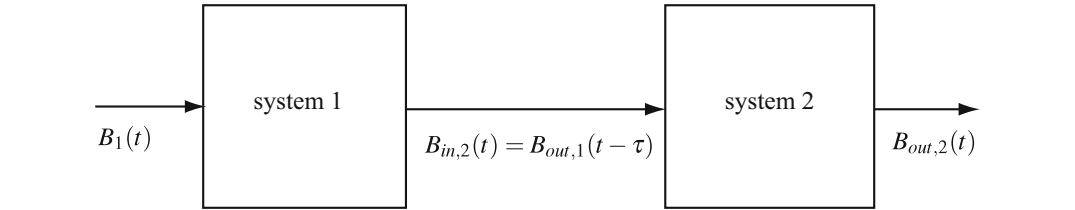
The purpose of this entry is to present some easy-to-use tools for constructing feedback networks involving quantum components and optical “wires.” What is presented in this entry is a simplification of a more general quantum feedback network theory (Yanagisawa and Kimura 2003; Gough and James 2009), which builds on earlier cascade theory (Gardiner 1993; Carmichael 1993), network quantization (Yurke and Denker 1984), and quantum control (Wiseman and Milburn 1994).

The basic idea is simple, Fig. 1. Take an output channel and connect it to an input channel. Such series or cascade connections are commonplace in classical electrical circuit theory. For instance, if the systems are resistors with resistances R_1 and R_2 , then the total series-connected system is equivalent to a single resistor with resistance $R = R_1 + R_2$. This use of simple parameters for devices, and rules for interconnecting devices in terms of these parameters, is a powerful feature of classical electrical circuit theory.

Specifically, what we wish to do is to take open quantum models G_1, G_2 for the two systems and combine them to form an equivalent open quantum model $G = G_2 \triangleleft G_1$, as in Fig. 2.

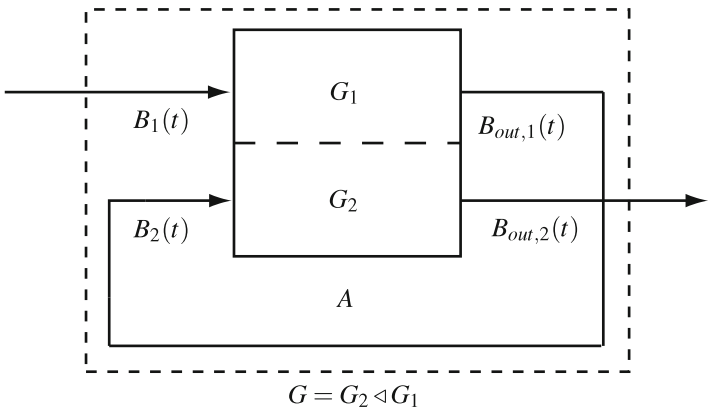
Open Quantum Systems

An atom interacting with an electromagnetic field is a fundamental example of an open quantum system, Fig. 3. Such systems were studied by Einstein which he described using his famous rate equations (Einstein 1916) for absorption and emission of photons. An atom in its ground state may absorb a quanta of energy from the field, while an excited atom may release quantized energy into the field. Modern models for open quantum systems (Gardiner and Zoller 2000) have an input-output structure, where the field is represented in terms of incoming and outgoing components.

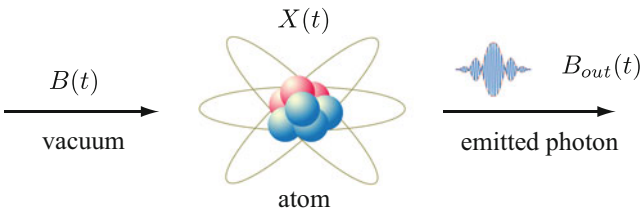


Quantum Networks, Fig. 1 A series connection between two systems

Quantum Networks,
Fig. 2 A coherent
feedback loop described by
a series connection
 $G = G_2 \triangleleft G_1$



Quantum Networks,
Fig. 3 An atom interacting
with an electromagnetic
field (e.g., photon
emission)



In the absence of any other influences, the behavior of the atom will evolve in time according to the laws of quantum mechanics, as determined by the self-energy of the atom, the nature of the fields, and how they are coupled to the atom. The field channels may be used to gather information about the atom via photodetection and to influence the behavior of the atom by directing light onto the atom. The fields may also be used for interconnection, where, for example, the emitted light from one system could be directed onto another.

In quantum mechanics, systems can be represented in terms of an underlying Hilbert space. States are unit vectors in this space, and physical quantities are represented as operators; see, for example, Nielsen and Chuang (2000, Chapter 2). Together, states and operators provide a quantum noncommutative probabilistic model, Bouten et al (2007, Section 2.2). Measurements are described in these probabilistic terms, and dynamical evolution of states and operators is unitary – Schrodinger’s equation.

A simple two-level model for an atom may be described using the Pauli matrices

$$\begin{aligned}\sigma_0 = I &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \\ \sigma_y &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.\end{aligned}\quad (1)$$

Any physical quantity of the atom can be expressed in terms of these matrices. In particular, the atomic energy levels are the eigenvalues $\pm \frac{1}{2}\omega$ of the Hamiltonian $H = \frac{1}{2}\omega\sigma_z$ describing the self-energy of the atom.

The atom interacts with the field channels by an exchange of energy that may be described by first principles in terms of an interaction Hamiltonian, Gardiner and Zoller (2000, Chapter 3). In this entry we use an idealized quantum noise model for the open atom-field system which is well justified theoretically and experimentally, Hudson and Parthasarathy (1984), Gardiner and Collett (1985), Parthasarathy (1992), and Gardiner and Zoller (2000). The input field, $B(t)$, drives an interaction-picture equation for

a unitary operator $U(t)$ governing the atom-field system. If $|\psi_a\rangle$ and $|\psi_f\rangle$ are initial atomic and field states, respectively, the state of the atom-field system at time t is $U(t)|\psi_a\psi_f\rangle$. Here, we are using the standard Dirac “ket” notation for state vectors, with the tensor product notation $|\psi_a\psi_f\rangle = |\psi_a\rangle \otimes |\psi_f\rangle$.

Let’s now look at the equations governing the driven atomic system. We suppose that the field is initially in the vacuum state, which we denote by $|\psi_f\rangle = |0\rangle$. In this case the input process $B(t)$ is a quantum Wiener process, for which the non-zero Ito product is $dB(t)dB^\dagger(t) = dt$. The atom-field coupling is determined by coupling operator $L = \sqrt{\kappa}\sigma_-$, where

$$\sigma_- = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

is the lowering operator (the raising operator is defined by $\sigma_+ = \sigma_-^\dagger$) and $\kappa > 0$ describes the strength of the coupling to the field. The evolution of the unitary $U(t)$ is given by the Schrodinger equation

$$dU(t) = \left\{ dB^\dagger(t)L - L^\dagger dB(t) - \frac{1}{2}(L^\dagger L + iH)dt \right\} U(t), \quad (2)$$

where $L = \sqrt{\kappa}\sigma_-$ and $H = \frac{1}{2}\omega\sigma_z$. Equation (2) is *quantum stochastic differential equation*, expressed in quantum-Ito differential form.

In the Heisenberg picture, atomic operators X evolve according to $X(t) = U^\dagger(t)XU(t)$, so that we have

$$\begin{aligned}dX(t) &= (-i[X(t), H(t)] + \mathcal{L}_{L(t)}(X(t)))dt \\ &\quad + dB^\dagger(t)[X(t), L(t)] \\ &\quad + [L^\dagger(t), X(t)]dB(t),\end{aligned}\quad (3)$$

and

$$\mathcal{L}_L(X) = \frac{1}{2}L^\dagger[X, L] + \frac{1}{2}[L^\dagger, X]L. \quad (4)$$

The output field is defined by $B_{\text{out}}(t) = U^\dagger(t)B(t)U(t)$, which in differential form is

given by the equation

$$dB_{\text{out}}(t) = L(t)dt + dB(t). \quad (5)$$

Explicitly, for the operators $X = \sigma_x$, σ_y , and σ_z , we have, with $L = \sqrt{\kappa} \sigma_-$ and $H = \frac{1}{2}\omega\sigma_z$,

$$\begin{aligned} d\sigma_x(t) = & (-\omega\sigma_y(t) - \frac{\kappa}{2}\sigma_x(t))dt \\ & + \sqrt{\kappa}(dB^\dagger(t)\sigma_z(t) + \sigma_z(t)dB(t)) \end{aligned} \quad (6)$$

$$\begin{aligned} d\sigma_y(t) = & (\omega\sigma_x(t) - \frac{\kappa}{2}\sigma_y(t))dt \\ & - i\sqrt{\kappa}(\sigma_z(t)dB^\dagger(t) - \sigma_z(t)dB(t)) \end{aligned} \quad (7)$$

$$\begin{aligned} d\sigma_z(t) = & (-\kappa\sigma_z(t) - \kappa)dt \\ & - 2\sqrt{\kappa}(dB^\dagger(t)\sigma_-(t) + \sigma_+(t)dB(t)), \end{aligned} \quad (8)$$

with output equation

$$dB_{\text{out}}(t) = \sqrt{\kappa} \sigma_-(t)dt + dB(t). \quad (9)$$

The equations of motion for the atomic state density operator ρ (convex combinations of outer products $|\psi\rangle\langle\psi|$) may be obtained by averaging out the noise in the Schrodinger picture, $\rho(t) = \text{tr}_B[U(t)(\rho \otimes |0\rangle\langle 0|)U^\dagger(t)]$; here, we take the field to be in the vacuum state. The differential equation for $\rho(t)$ is

$$\dot{\rho}(t) = i[\rho(t), H] + \mathcal{L}_L^\dagger(\rho(t)) \quad (10)$$

where

$$\mathcal{L}_L^\dagger(\rho) = \frac{1}{2}[L, \rho L^\dagger] + \frac{1}{2}[L\rho, L^\dagger]. \quad (11)$$

Equation (10) is called the *master equation*, and $i[\rho, H] + \mathcal{L}_L^\dagger(\rho)$ is called the *Lindblad super-operator* (in Schrodinger form). The equations of motion for the averages $x(t) = \text{tr}[\rho(t)\sigma_x]$, $y(t) = \text{tr}[\rho(t)\sigma_y]$, and $z(t) = \text{tr}[\rho(t)\sigma_z]$ are

$$\dot{x}(t) = -\frac{\kappa}{2}x(t) - \omega y(t), \quad (12)$$

$$\dot{y}(t) = -\frac{\kappa}{2}y(t) + \omega x(t), \quad (13)$$

$$\dot{z}(t) = -\kappa z(t) - \kappa, \quad (14)$$

In Equations (12), (13), and (14), we can see the effect of the field coupling, which causes $x(t) \rightarrow 0$, $y(t) \rightarrow 0$, and $z(t) \rightarrow -1$ as $t \rightarrow \infty$, i.e., $\rho(t) \rightarrow \frac{1}{2}(\sigma_z - I) = |-1\rangle\langle -1|$, the pure state of lowest energy.

Notice that once the physical parameters H and L and initial states are given, the equations of motion can readily be derived.

Series Network

Let's see how we can achieve a simple, symbolic rule involving the physical parameters $G = (L, H)$ analogous to $R = R_1 + R_2$ for the series connection of two open quantum systems, where the input and output channels are quantum fields. As mentioned in the example of the atom above, the physical parameters determining open quantum systems are a Hamiltonian H describing the self-energy of the system and a vector L of operators describing how the system is coupled to the field channels. Actually, there is a third parameter S , a self-adjoint matrix of operators describing scattering between field channels, that was introduced in Hudson and Parthasarathy (1984) and Parthasarathy (1992). While S does not appear in the Lindblad nor the master equation for a single system, it does have nontrivial use in several applications including quantum feedback networks (such as those including beamsplitters). Thus in general an open quantum system, call it G , is characterized by three parameters $G = (S, L, H)$. However, in this entry we do not use the scattering parameter, and so we set $S = I$; actually, we make the abbreviation $G = (L, H)$.

For example, the parameters for the atom, coupled to two field channels, call it A , are

$$A = \left(\begin{pmatrix} \sqrt{\kappa_1} \sigma_- \\ \sqrt{\kappa_2} \sigma_- \end{pmatrix}, \frac{1}{2}\omega\sigma_z \right). \quad (15)$$

Here,

$$L = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}, \quad (16)$$

where $L_1 = \sqrt{\kappa_1} \sigma_-$, $L_2 = \sqrt{\kappa_2} \sigma_-$, and $H = \frac{1}{2} \omega \sigma_z$.

In network modeling, and indeed in modeling in general, it can be helpful to decompose large systems into smaller pieces and to assemble large systems from components. In Gough and James (2009), the *concatenation product* $\boxplus$ was introduced to assist with this. The concatenation product is defined by

$$(L_1, H_1) \boxplus (L_2, H_2) = \left(\begin{pmatrix} L_1 \\ L_2 \end{pmatrix}, H_1 + H_2 \right). \quad (17)$$

For the atom, if we wish to decompose it with respect to field channels, we may write

$$A = (\sqrt{\kappa_1} \sigma_-, \frac{1}{2} \omega \sigma_z) \boxplus (\sqrt{\kappa_2} \sigma_-, 0). \quad (18)$$

Now suppose we have two systems $G_1 = (L_1, H_1)$ and $G_2 = (L_2, H_2)$, as in Fig. 1. Because the systems are separated spatially, the field segment connecting the output of G_1 to the input of G_2 has non-zero length, and so this means there is a small delay τ in the transmission of quantum information from G_1 to G_2 . That is, $B_{\text{in},2}(t) = B_{\text{out},1}(t - \tau)$. Now if the systems are sufficiently close, τ will be small compared with the timescales of the systems and may be neglected. In this zero-delay limit, the output of the second system is given by, using the output equations,

$$\begin{aligned} dB_{2,\text{out}}(t) &= L_2(t)dt + (L_1(t)dt + dB_1(t)), \\ &= (L_1(t) + L_2(t))dt + dB_1(t). \end{aligned} \quad (19)$$

(20)

This suggests an effective coupling $L = L_1 + L_2$.

In this way, a Markovian model for the series connection $G = G_2 \triangleleft G_1$ may be derived. In terms of the physical parameters, the *series product* (defined in Gough and James 2009) is given by

$$\begin{aligned} (L_2, H_2) \triangleleft (L_1, H_1) \\ = (L_1 + L_2, H_1 + H_2 + \text{Im}[L_2^\dagger L_1]). \end{aligned} \quad (21)$$

Thus the series connection $G_{\text{series}} = G_2 \triangleleft G_1$ has parameters $L = L_1 + L_2$ and $H = H_1 + H_2 + \text{Im}[L_2^\dagger L_1]$, analogous to the expression $R = R_1 + R_2$ for series-connected resistors.

Coherent Feedback

The series product serves very well for Markovian approximations to cascades of independent open systems. However, importantly for us, the series product may also be used to describe an important class of *quantum feedback networks*. This is because the two systems $G_1 = (L_1, H_1)$ and $G_2 = (L_2, H_2)$ need not be independent – they can be parts of the same system.

For example, suppose we take G_1 to be the first factor in the decomposition (18) of A , i.e., $G_1 = (\sqrt{\kappa_1} \sigma_-, \frac{1}{2} \omega \sigma_z)$, and $G_2 = (\sqrt{\kappa_2} \sigma_-, 0)$ the second. Before feedback, the Heisenberg evolution equations for the system $A = (\sqrt{\kappa_1} \sigma_-, \frac{1}{2} \omega \sigma_z) \boxplus (\sqrt{\kappa_2} \sigma_-, 0)$ are

$$\begin{aligned} d\sigma_x(t) &= (-\omega \sigma_y(t) - \frac{\kappa_1 + \kappa_2}{2} \sigma_x(t))dt \\ &\quad + \sqrt{\kappa_1} (dB_1^\dagger(t) \sigma_z(t) + \sigma_z(t) dB_1(t)) \\ &\quad + \sqrt{\kappa_2} (dB_2^\dagger(t) \sigma_z(t) + \sigma_z(t) dB_2(t)) \end{aligned} \quad (22)$$

$$\begin{aligned} d\sigma_y(t) &= (\omega \sigma_x(t) - \frac{\kappa_1 + \kappa_2}{2} \sigma_y(t))dt \\ &\quad - i\sqrt{\kappa_1} (dB_1^\dagger(t) \sigma_z(t) - \sigma_z(t) dB_1(t)) \\ &\quad - i\sqrt{\kappa_2} (dB_2^\dagger(t) \sigma_z(t) - \sigma_z(t) dB_2(t)) \end{aligned} \quad (23)$$

$$\begin{aligned} d\sigma_z(t) &= (-\kappa_1 - \kappa_2)(\sigma_z(t) + 1)dt \\ &\quad - 2\sqrt{\kappa_1} (dB_1^\dagger(t) \sigma_-(t) + \sigma_+(t) dB_1(t)) \\ &\quad - 2\sqrt{\kappa_2} (dB_2^\dagger(t) \sigma_-(t) + \sigma_+(t) dB_2(t)) \end{aligned} \quad (24)$$

with output equations

$$dB_{1,\text{out}}(t) = \sqrt{\kappa_1} \sigma_-(t)dt + dB_1(t), \quad (25)$$

$$dB_{2,\text{out}}(t) = \sqrt{\kappa_2} \sigma_-(t)dt + dB_2(t). \quad (26)$$

The series connection for the fields gives

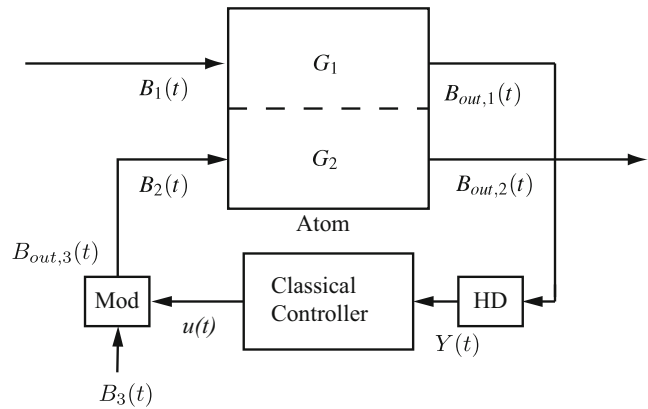
$$\begin{aligned} dB_2(t) &= dB_{1,\text{out}}(t) \\ &= \sqrt{\kappa_1} \sigma_-(t)dt + dB_1(t). \end{aligned} \quad (27)$$

One may now be tempted to substitute equation (27) into equations (22), (23), and (24). However, one must be careful because the Ito increments commute with system operators, e.g., $dB_2^\dagger(t)\sigma_z(t) = \sigma_z(t)dB_2^\dagger(t)$, whereas $\sigma_z\sigma_- \neq \sigma_-\sigma_z$. Fortunately, the substitutions are correct provided we preserve the normal ordering given in (22), (23), and (24), Gough and James (2017). An expression is normally ordered if the adjoints all appear on the left, e.g., $A^\dagger B + B^\dagger A$. For instance, performing the normally ordered substitutions in equation (22) gives

$$\begin{aligned} d\sigma_x(t) &= (-\omega\sigma_y(t) - \frac{\kappa_1 + \kappa_2}{2}\sigma_x(t))dt \\ &\quad + \sqrt{\kappa_1}(dB_1^\dagger(t)\sigma_z(t) + \sigma_z(t)dB_1(t)) \\ &\quad + \sqrt{\kappa_2}((\sqrt{\kappa_1}\sigma_+(t)dt + dB_1^\dagger(t))\sigma_z(t) \\ &\quad + \sigma_z(t)(\sqrt{\kappa_1}\sigma_-(t)dt + dB_1(t))) \end{aligned} \quad (28)$$

Quantum Networks,

Fig. 4 Atom in a measurement feedback loop with a classical controller



$$\begin{aligned} &= (-\omega\sigma_y(t) - \frac{(\sqrt{\kappa_1} + \sqrt{\kappa_2})^2}{2}\sigma_x(t))dt \\ &\quad + (\sqrt{\kappa_1} + \sqrt{\kappa_2})(dB_1^\dagger(t)\sigma_z(t) \\ &\quad + \sigma_z(t)dB_1(t)) \end{aligned} \quad (29)$$

This corresponds to the series connection

$$\begin{aligned} G_{fb} &= (\sqrt{\kappa_2}\sigma_-, 0) \triangleleft \left(\sqrt{\kappa_1}\sigma_-, \frac{1}{2}\omega\sigma_z \right) \\ &= \left((\sqrt{\kappa_1} + \sqrt{\kappa_2})\sigma_-, \frac{1}{2}\omega\sigma_z \right) \end{aligned} \quad (30)$$

which describes the coherent feedback arrangement shown in Fig. 2. The system G_{fb} is an open quantum system with Hamiltonian $H = \frac{1}{2}\omega\sigma_z$ that is coupled to a single field channel via the coupling operator $L = (\sqrt{\kappa_1} + \sqrt{\kappa_2})\sigma_-$.

Measurement Feedback

In general quantum feedback networks may include both quantum and classical components. For instance, in a measurement feedback system, the controller is a classical dynamical system that processes a signal made from measurements of the quantum system, and these measurement results are processed to determine control actions. An example is shown in Fig. 4. Here, an output channel of the atom discussed above is continuously monitored by homodyne detection

(HD). The classical measurement signal $Y(t)$ is fed as an input into a classical controller. The classical controller might, for example, be constructed from ordinary analog or digital electronics. The control signal $u(t)$ produced is used to modulate an input channel of the atom. In this way, a quantum-classical closed-loop system is formed.

Ideal homodyne detection of the first channel of the atom may be described by

$$Y(t) = B_{\text{out},1}(t) + B_{\text{out},1}^\dagger(t). \quad (31)$$

Here, the signal is self-adjoint and commutative $[Y(t), Y(s)] = 0$ and so by the spectral theorem (Bouten et al 2007, Theorem 2.4) is equivalent to a classical stochastic process. The classical control signal $u(t)$ modulates a vacuum field $B_3(t)$ to produce the atom's second input $B_2(t) = B_{\text{out},3}(t)$:

$$dB_{\text{out},3}(t) = u(t)dt + dB_3(t). \quad (32)$$

Both homodyne detection and modulation are standard procedures in quantum optics (Bachor and Ralph 2004).

We suppose (for simplicity) that the controller is a first-order linear system

$$dx(t) = -ax(t)dt + bdY(t) \quad (33)$$

$$u(t) = cx(t). \quad (34)$$

Equations (31), (32), (33), and (34) involve a mixture of quantum and classical variables. We may regard the classical variables as belonging to a commutative subsystem of a larger, composite noncommutative quantum system. One way to describe this is to represent the controller dynamics in terms of an open quantum $C = C_1 \boxplus C_2$, where

$$C_1 = (cq, -\frac{\alpha}{2}(qp + pq)) \quad (35)$$

and

$$C_2 = (-ibp, 0). \quad (36)$$

Here, q and p are the canonical position and momentum operators defined by $q\varphi(x) = x\varphi(x)$, $p\varphi(x) = -i\frac{d}{dx}\varphi(x)$, and the underlying Hilbert space for C is $\mathfrak{H}_C = L^2(\mathbf{R})$. The open quantum system C has inputs $B_3(t)$ and $B_4(t)$.

For $X = \varphi$ a smooth function, we have from (3)

$$d\varphi = (-a\varphi' + \frac{1}{2}b^2\varphi'')dt + b\varphi'd(B_4(t) + B_4^\dagger(t)), \quad (37)$$

and in particular when $\varphi(q) = q$ and $B_4(t) = B_{\text{out},1}(t)$, this reduces to

$$dq = -aqdt + bdY(t), \quad (38)$$

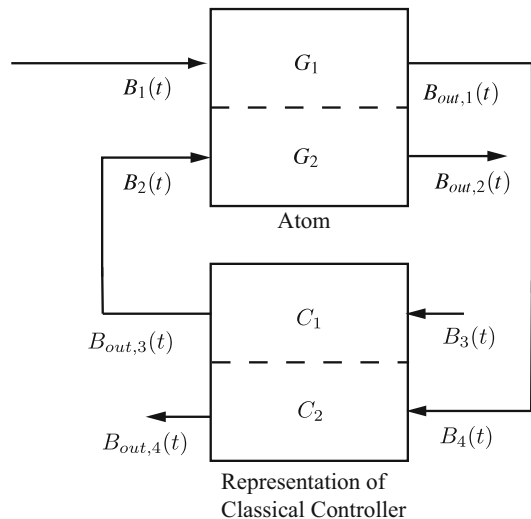
which represents the classical dynamics (33).

From the general expression (5), the first output of the system C is given by

$$dB_{\text{out},3}(t) = cq(t)dt + dB_3(t) \quad (39)$$

This corresponds to the modulator output (32) (Fig. 5).

Now that the classical and quantum components are represented at the same level, we may



Quantum Networks, Fig. 5 Quantum representation of an atom in a measurement feedback loop with a classical controller

apply the series product to determine the parameters of the measurement feedback system. From the above representations, the parameters for the measurement feedback system are

$$\begin{aligned}
 G_{mfb} &= (C_2 \triangleleft G_1) \boxplus (G_2 \triangleleft C_1) \\
 &= \left(-ibp + \sqrt{\kappa_1} \sigma_-, \frac{1}{2} \omega \sigma_z \right. \\
 &\quad \left. + \frac{1}{2} \sqrt{\kappa_1} bp \sigma_x \right) \\
 &\boxplus \left(\sqrt{\kappa_2} \sigma_- + cq, -\frac{\alpha}{2} (qp + pq) \right. \\
 &\quad \left. + \frac{1}{2} \sqrt{\kappa_2} cq \sigma_y \right). \quad (41)
 \end{aligned}$$

Summary and Future Directions

We have seen that idealized models for open quantum systems provide a powerful framework for describing feedback networks that include quantum as well as classical systems. This framework is based on quantum stochastic calculus, part of a general noncommutative quantum probability theory. The systematic exploitation of the concepts and methods from noncommutative quantum probability theory is seen as important for the future development of quantum technology, as it emerges from basic physics.

The series product is a fundamental symbolic tool for representing series connections of open quantum systems. The series product provides a recipe for determining the physical parameters of systems connected in series. Once the parameters are known, dynamical equations for the network can be easily derived. The derivation of the dynamical network equations using the series product is equivalent to algebraic substitutions in equations provided the normally ordered form is preserved. The series product is symbolic in nature, and indeed symbolic tools, such as the *Quantum Hardware Description Language (QHDL)*, are beginning to emerge, Tezak et al (2012).

Cross-References

- [Application of Systems and Control Theory to Quantum Engineering](#)
- [Control of Quantum Systems](#)
- [Learning Control of Quantum Systems](#)

References

- Bachor H, Ralph T (2004) A guide to experiments in quantum optics, 2nd edn. Wiley-VCH, Weinheim, Germany
- Bouten L, van Handel R, James M (2007) An introduction to quantum filtering. *SIAM J Control Optim* 46(6):2199–2241
- Carmichael H (1993) Quantum trajectory theory for cascaded open systems. *Phys Rev Lett* 70(15):2273–2276
- Einstein A (1916) Strahlungs-emission und -absorption nach der quantentheorie. *Verhandlungen der Deutschen Physikalischen Gesellschaft* 18:318–323
- Gardiner C (1993) Driving a quantum system with the output field from another driven quantum system. *Phys Rev Lett* 70(15):2269–2272
- Gardiner C, Collett M (1985) Input and output in damped quantum systems: quantum stochastic differential equations and the master equation. *Phys Rev A* 31(6):3761–3774
- Gardiner C, Zoller P (2000) *Quantum noise*. Springer, Berlin
- Gough JE, James M (2009) The series product and its application to quantum feedforward and feedback networks. *IEEE Trans Automatic Control* 54(11):2530–2544
- Gough JE, James MR (2017) The series product for gaussian quantum input processes. *Rep Math Phys* 79(1):111–133
- Hudson R, Parthasarathy K (1984) Quantum Ito's formula and stochastic evolutions. *Commun Math Phys* 93:301–323
- Nielsen M, Chuang I (2000) *Quantum computation and quantum information*. Cambridge University Press, Cambridge
- Parthasarathy K (1992) *An introduction to quantum stochastic calculus*. Birkhauser, Berlin
- Tezak N, Niederberger A, Pavlichin DS, Sarma G, Mabuchi H (2012) Specification of photonic circuits using quantum hardware description language. *Philos Trans R Soc A Math Phys Eng Sci* 370(1979):5270–5290
- Wiseman HM, Milburn GJ (1994) All-optical versus electro-optical quantum-limited feedback. *Phys Rev A* 49(5):4110–4125
- Yanagisawa M, Kimura H (2003) Transfer function approach to quantum control-part I: dynamics of quantum feedback systems. *IEEE Trans Autom Control* 48:2107–2120
- Yurke B, Denker J (1984) Quantum network theory. *Phys Rev A* 29(3):1419–1437

Quantum Stochastic Processes and the Modelling of Quantum Noise

Hendra I. Nurdin

School of Electrical Engineering and Telecommunications, University of New South Wales (UNSW), Sydney, NSW, Australia

Abstract

This brief entry gives an overview of quantum mechanics as a *quantum probability theory*. It begins with a review of the basic operator-algebraic elements that connect probability theory with quantum probability theory. Then quantum stochastic processes are formulated as a generalization of stochastic processes within the framework of quantum probability theory. Quantum Markov models from quantum optics are used to explicitly illustrate the underlying abstract concepts and their connections to the quantum regression theorem from quantum optics.

Keywords

Quantum feedback control · Quantum stochastic systems · Quantum stochastic control

Introduction

Many phenomena display dynamics that appear to be random and can often be accurately modeled as stochastic processes. The modern theory of stochastic processes is built upon the measure-theoretic axiomatization of probability theory as developed by Kolmogorov in the 1930s; for a historical overview, see, e.g., Bingham (2000); Shafer and Vovk (2006). These theories underpin the stochastic systems theory and stochastic control theory that find wide applications in disciplines such as engineering (in particular control engineering), economics, and mathematical finance. At the atomic size

and energy scales, where classical physics is superseded by quantum mechanics, randomness is inherent. Indeed, measurement of a quantum mechanical system yields a random outcome, as dictated by the measurement postulate of quantum mechanics. Research into the axiomatic foundation of quantum mechanics since the seminal work of von Neumann have led to the recognition that quantum mechanics is essentially a noncommutative generalization of probability theory. That is, *quantum mechanics is a quantum probability theory*.

Quantum probability theory provides a rigorous basis for studying quantum stochastic processes and developing a modern formulation of quantum stochastic systems and control theory. This facilitates the systematic formulation and solution of estimation and feedback control problems for quantum systems (Bouten et al. 2007; Bouten and van Handel 2008; Gough and James 2009b; Nurdin and Yamamoto 2017). This entry provides an overview of quantum probability theory and the formulation of quantum stochastic processes and quantum Markov processes based on this theory. The abstract mathematical concepts introduced will then be explicitly illustrated in the context of quantum Markov models for a large class of quantum optical devices.

We will use fairly standard notations, with additional notations introduced later in the text. $\mathbb{R}$, $\mathbb{C}$, and $\mathbb{R}_{0+}$ denote the real and complex numbers and $\mathbb{R}_{0+} = [0, \infty)$, respectively. For $c \in \mathbb{C}$, $\bar{c}$ is its complex conjugate. For a complex-valued function X , $\bar{X}(\cdot) = \overline{X(\cdot)}$. For a set S , S^n denotes the n -fold direct product $S^n = \underbrace{S \times S \times \cdots \times S}_{n \text{ times}}$.

The indicator function for a set A is denoted by $\mathbf{1}_A$. A Dirac ket $|x\rangle$ denotes a complex vector in a Hilbert space, while a bra $\langle x|$ denotes the conjugate transpose (or dual functional) of the vector. Thus $\langle x|y\rangle$ is the inner product of $|x\rangle$ and $|y\rangle$. For any operator X mapping a Hilbert space to another, $X^\dagger$ denotes the adjoint of X (if X maps a Hilbert space to itself, then $X^\dagger$ is also referred to as the Hermitian conjugate). $\text{Tr}(X)$ denotes the trace of a trace-class operator. For a self-adjoint (Hermitian) operator, X , $X \geq 0$

denotes a positive operator, $\langle x|X|x \rangle \geq 0$ for all elements x of the Hilbert space.

Quantum Probability Theory and Quantum Stochastic Processes

A (classical) (In this entry we will use the qualifier “classical” in brackets to emphasize that a theory is not quantum mechanical.) probability space is a tuple $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the sample space, $\mathcal{F}$ the event set as a σ -algebra of subsets of Ω , and $\mathbb{P}$ is a probability measure on the measure space $(\Omega, \mathcal{F})$. We endow $\mathbb{C}$ with its Borel σ -algebra $\mathcal{B}(\mathbb{C})$, making $(\mathbb{C}, \mathcal{B}(\mathbb{C}))$ a measurable space. A random variable $X : (\Omega, \mathcal{F}) \rightarrow (\mathbb{C}, \mathcal{B}(\mathbb{C}))$ is a measurable complex-valued map from Ω to $\mathbb{C}$ (for the sake of simplicity, we intentionally restrict ourselves only to random variables taking values in $\mathbb{C}$). The expectation value $\mathbb{E}[X]$ of X is given by $\mathbb{E}[X] = \int_{\Omega} X(\omega) P(d\omega)$. Given a sub- σ -algebra $\mathcal{G}$ of $\mathcal{F}$, the conditional expectation of a random variable X on $\mathcal{G}$, denoted by $\mathbb{E}[X|\mathcal{G}]$, is a random variable with the properties (i) $\mathbb{E}[X|\mathcal{G}]$ is $\mathcal{G}$ -measurable and (ii) $\mathbb{E}[\mathbb{E}[X|\mathcal{G}]K] = \mathbb{E}[XK]$ for any $\mathcal{G}$ -measurable random variable K . It follows from this definition that if $\mathcal{G}$ and $\mathcal{H}$ are sub- σ -algebras of $\mathcal{F}$ with $\mathcal{H} \subset \mathcal{G}$, then $\mathbb{E}[\mathbb{E}[X|\mathcal{G}]|\mathcal{H}] = \mathbb{E}[X|\mathcal{H}]$ (the tower property).

For the rest of the entry, let $T \subseteq \mathbb{R}$ denote a time index. For example, T can be discrete as in $T = \{0, 1, 2, \dots\}$, or it can be continuous as in $T = [0, \infty)$. Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, a classical stochastic process over T is a collection of random variables $\{X_t\}_{t \in T}$ on $(\Omega, \mathcal{F}, \mathbb{P})$ indexed by elements of T . From a practical perspective, one would not start with a given $(\Omega, \mathcal{F}, \mathbb{P})$ but with the specification of a countable collection of finite-dimensional probability distributions $F_{t_1, t_2, \dots, t_n}(C_1, C_2, \dots, C_n) = \mathbb{P}(X_{t_1} \in C_1, X_{t_2} \in C_2, \dots, X_{t_n} \in C_n)$ for any integer $n \geq 1$, any collection of distinct points $t_1, t_2, \dots, t_n \in T$ and any $C_j \in \mathcal{B}(\mathbb{C})$, $j = 1, 2, \dots, n$, which describe the stochastic process. As is well-known, under the Kolmogorov consistency conditions on the finite-dimensional distributions, a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ can

be reconstructed on which the stochastic process $\{X_t\}_{t \in T}$ satisfying all the specifications of the finite-dimensional distributions can be realized; see, e.g., Billingsley (1986).

An equivalent formulation of probability theory in terms of algebras of operators over a Hilbert space, attributed to Gelfand (Streater 2000, Theorem 3.2), is the key to connecting probability theory to quantum mechanics. Consider an essentially bounded random variable X on $(\Omega, \mathcal{F}, \mathbb{P})$, $\text{ess sup}_{\omega \in \Omega} |X(\omega)| < \infty$. Denote the class of all such random variables by $L^\infty(\Omega, \mathcal{F}, \mathbb{P})$. Each element of $L^\infty(\Omega, \mathcal{F}, \mathbb{P})$ may be viewed as a multiplication operator on elements of the Hilbert space $L^2(\Omega, \mathcal{F}, \mathbb{P})$ of square-integrable complex-valued functions on $(\Omega, \mathcal{F}, \mathbb{P})$, in the sense that $X : Y(\cdot) \in L^2(\Omega, \mathcal{F}, \mathbb{P}) \mapsto X(\cdot)Y(\cdot) \in L^2(\Omega, \mathcal{F}, \mathbb{P})$. The class $L^\infty(\Omega, \mathcal{F}, \mathbb{P})$ is closed under the “+” operation ($X_1, X_2 \in L^\infty(\Omega, \mathcal{F}, \mathbb{P}) \rightarrow X_1 + X_2 \in L^\infty(\Omega, \mathcal{F}, \mathbb{P})$) and under the commutative operator composition operation ($X_1, X_2 \in L^\infty(\Omega, \mathcal{F}, \mathbb{P}) \rightarrow X_1 X_2 = X_2 X_1 \in L^\infty(\Omega, \mathcal{F}, \mathbb{P})$). Moreover, an antilinear unary involution operator can be defined on $L^\infty(\Omega, \mathcal{F}, \mathbb{P})$ as $(*) : X \mapsto \bar{X}$. Thus, $L^\infty(\Omega, \mathcal{F}, \mathbb{P})$ forms a commutative $*$ -algebra of operators.

A state μ on a $*$ -operator algebra $\mathcal{A}$ with identity operator $I_{\mathcal{A}}$ (with respect to the composition operation) is a linear complex functional on $\mathcal{A}$ which is positive (i.e., $\mu(A) \geq 0$ for all $0 \leq A \in \mathcal{A}$) and unital, $\mu(I_{\mathcal{A}}) = 1$. A map M acting on $\mathcal{A}$ (such as the state μ) is said to be normal if $M(\sup_{\alpha} A_{\alpha}) = \sup_{\alpha} M(A_{\alpha})$ for any bounded increasing net $\{A_{\alpha}\}$ of positive elements in $\mathcal{A}$. The algebra $L^\infty(\Omega, \mathcal{F}, \mathbb{P})$ is a commutative $*$ -operator algebra on $L^2(\Omega, \mathcal{F}, \mathbb{P})$ (since the composition operation is commutative), and $\mathbb{E}[\cdot]$ is a normal state on this algebra; see, e.g., Bouten et al. (2007) and the references therein. More precisely, the $*$ -algebra is a commutative von Neumann algebra (Bratelli and Robinson 1979; Bouten et al. 2007).

Conversely, it can be shown that any commutative von Neumann algebra $\mathcal{A}$ and a unital normal state μ on $\mathcal{A}$ is $*$ -isomorphic (a bijective correspondence that preserves the involution

operation $*$) to $(L^\infty(\Omega, \mathcal{F}, \nu), \mathbb{E})$, where $\mathbb{E}[\cdot] = \int_\Omega (\cdot)(\omega) \mathbb{P}(d\omega)$ for some probability measure $\mathbb{P}$ on $(\Omega, \mathcal{F})$ which is absolutely continuous with respect to ν . This suggests that a natural generalization of probability theory would be a noncommutative von Neumann algebra $\mathcal{A}$ with identity operator $I_{\mathcal{A}}$ that is endowed with a unital normal state μ ; see Streater (2000) for a historical overview of quantum probability theory. The pair $(\mathcal{A}, \mu)$ is referred to as a *quantum probability space*. If $\mathcal{X} \subset \mathcal{A}$ is a Neumann algebra containing $I_{\mathcal{A}}$, then a positive linear map $E^{\mathcal{X}} : \mathcal{A} \rightarrow \mathcal{X}$ is a conditional expectation map if $\mu(x_1 A x_2) = \mu(x_1 E^{\mathcal{X}}[A] x_2)$ for any $A \in \mathcal{A}$ and $x_1, x_2 \in \mathcal{X}$.

The physical interpretation associated with a quantum probability space is as follows. The underlying Hilbert space of the von Neumann algebra corresponds to the Hilbert space of a quantum mechanical system. The Hermitian elements of the algebra represent physical observables. Orthogonal projection operators in the algebra (operators P satisfying $P^2 = P = P^\dagger$) represent events that can take place, such as the event that “observable X takes on the value c on measurement.” Two events P_1 and P_2 are compatible if they commute; these are events that can be assigned as a joint probability distribution in the classical sense. If the initial state is the density operator ρ , then $\mu(X) = \text{Tr}(\rho X)$ for any observable X . As will be seen next, in a quantum probability space, it is more natural to consider quantum dynamics in the Heisenberg picture, where operators evolve in time and the state is fixed to the initial one.

Let $\mathcal{B}$ and $\mathcal{A}$ be von Neumann algebras with identity operators $I_{\mathcal{B}}$ and $I_{\mathcal{A}}$, respectively, and $(\mathcal{A}, \mu)$ a quantum probability space. A *quantum stochastic process* over $\mathcal{B}$ indexed by $T \subseteq \mathbb{R}$ is a tuple $(\mathcal{A}, \{j_t\}_{t \in T}, \mu)$ where j_t is a $*$ -homomorphism of $\mathcal{B}$ into $\mathcal{A}$ for each $t \in T$ (i.e., $j_t(A^\dagger) = j_t(A)^\dagger$ and $j_t(AB) = j_t(A)j_t(B)$ for any $A, B \in \mathcal{B}$) such that $j_t(I_{\mathcal{B}}) = I_{\mathcal{A}}$ and $\mathcal{A} = \text{vN}(\{j_t(\mathcal{B}), t \in T\})$. Here $j_t(\mathcal{B}) = \{j_t(b), b \in \mathcal{B}\}$ and $\text{vN}(\mathcal{S})$ denotes the von Neumann algebra generated by the operators in $\mathcal{S}$. Since $j_t(X)$ need not commute at different times for any $X \in \mathcal{B}$, one cannot in general assign a joint probability distribution to the process $\{j_t(X)\}_{t \in T}$;

thus it does not have the exact analogue of a collection of finite-dimensional distributions that characterize a classical stochastic process. As a substitute for the finite-dimensional distributions, we introduce finite-dimensional correlation kernels w_{t_n} . For any integer $n \geq 1$, let $\mathbf{b}_n = (b_1, b_2, \dots, b_n) \in \mathcal{B}^n$ with $b_1, b_2, \dots, b_n \in \mathcal{B}$ and $\mathbf{t}_n = (t_1, t_2, \dots, t_n) \in \mathbb{R}^n$, and define $j_{t_n}(\mathbf{b}_n) = j_{t_n}(b_n) \cdots j_{t_2}(b_2) j_{t_1}(b_1)$. Then we define the finite-dimensional correlation kernels as:

$$w_{t_n}(\mathbf{a}_n, \mathbf{b}_n) = \mu(j_{t_n}(\mathbf{a}_n)^\dagger j_{t_n}(\mathbf{b}_n)). \quad (1)$$

One can define a notion of equivalence between two quantum stochastic processes $(\mathcal{A}_k, \{j_{k,t}\}_{t \in T}, \mu_k)$ for $k = 1, 2$ (Accardi et al. 1982, §1). Also, a reconstruction theorem analogous to reconstruction theorems for stochastic processes from finite-dimensional distributions in the classical setting, such as the Kolmogorov extension theorem, can be established (Accardi et al. 1982, Theorem 1.3). Roughly speaking, it states that under certain technical assumptions on w_{t_n} , there exists a quantum stochastic process with w_{t_n} as its correlation kernels, which is unique up to equivalence.

Markov processes are an important and large class of stochastic processes that are employed as models for many applications across diverse fields. Roughly speaking, it is a process where given the present, the future of the process is independent of its past. In the quantum setting, one encounters quantum stochastic processes with an analogous property but properly interpreted since the process generally involves non-commuting operators that do not have a joint probability distribution. They are prominent in quantum optics to model a wide range of situations involving the interaction of localized quantum systems with travelling optical fields and are used to accurately describe many quantum optical devices (Gardiner and Zoller 2004; Nurdin and Yamamoto 2017).

Consider a stochastic process $\{X_t\}$ with $t \in T \subseteq \mathbb{R}$. Then the process is Markov if for any integer $n \geq 2$ and any $t_1, t_2, \dots, t_n \in T$ satisfying $t_1 < t_2 < \dots < t_n$ we have that

$\mathbb{P}(X_{t_n} \in A \mid \sigma(\{X_{t_k}\}_{k=1, \dots, n-1})) = \mathbb{P}(X_{t_n} \in A \mid \sigma(\{X_{t_{n-1}}\}))$ for any $A \in \mathcal{B}(\mathbb{C})$, where $\sigma(Y)$ denotes the σ -algebra generated by the random variables in Y . That is, given a sequence $\{X_{t_k}\}_{k=1, 2, \dots, n-1}$, X_{t_n} only depends on the most recent past, $X_{t_{n-1}}$. Due to noncommutativity, this definition cannot be extended in an analogous way to define a quantum Markov process on a quantum probability space.

To introduce quantum Markov processes, we follow the treatment in Accardi et al. (1982). We start with the definition of canonical maps. Consider a quantum probability space $(\mathcal{A}, \mu)$ and let $\mathcal{A}_{[s,t]} = \text{vN}(\{j_u(\mathcal{B}), s \leq u \leq t \in T\})$, $\mathcal{A}_t = \mathcal{A}_{(-\infty, t]}$, $\mathcal{A}_t = \mathcal{A}_{[t, t]}$ and $\mathcal{A}_t = \mathcal{A}_{[t, \infty)}$. A two-parameter family $\{E_{s,t}\}_{s < t \in T}$ of completely positive identity preserving maps $E_{[s,t]} : \mathcal{A}_t \rightarrow \mathcal{A}_s$ for $s < t \in T$, which are compatible with μ in the sense that $\mu|_{\mathcal{A}_t} = \mu|_{\mathcal{A}_s} \circ E_{s,t}$, $\forall s < t \in T$, is said to be a family of *canonical maps* if they have the properties: (i) $E_{s,t}$ is a normal map and (ii) $E_{s,t}E_{t,u} = E_{s,u}$ for all $s < t < u \in T$. We have the following definition of a quantum Markov process.

Definition 1 (Quantum Markov process)

A quantum stochastic process $(\mathcal{A}, \{j_t\}_{t \in T}, \mu)$ over the von Neumann algebra $\mathcal{B}$ is a quantum Markov process if the canonical maps satisfy $E_{s,t}(\mathcal{A}_{[s,t]}) \subseteq \mathcal{A}_s \forall s < t \in T$.

The definition is slightly relaxed from the one given in Accardi et al. (1982) in that μ is not required to be faithful; only existence of the canonical maps is imposed.

Let j_t have the left inverse $j_t^* : \mathcal{A} \rightarrow \mathcal{B}$ for all $t \in T$ and define $E_{t,t}$ as the identity map on $\mathcal{A}_t$. Then we can define a two-parameter family $\{Z_{s,t}\}_{s < t \in T}$ of completely positive identity preserving maps of $\mathcal{B}$ to itself, defined by $Z_{s,t} = j_s^* E_{s,t} j_t$, $s \leq t \in T$, satisfying $Z_{s,t} Z_{t,u} = Z_{s,u} \forall s \leq t \in T$. In general, $\{E_{s,t}\}$ will not be conditional expectations of $\mathcal{A}_t$ to $\mathcal{A}_s$ unless additional technical conditions, such as from the Tomita-Takesaki theory, are satisfied (Accardi et al. 1982, p. 109). Note, however, that $E_{s,t}$ will be a conditional expectation when $\mathcal{A}$ is a commutative algebra. If we assume that the canonical maps are conditional expectations, then

we have that (Accardi et al. 1982, Theorem 2.1)

$$\begin{aligned} E_{t_0, t_n}(j_{t_1}(a_1)^\dagger \cdots j_{t_n}(a_n)^\dagger j_{t_1}(b_1) \cdots j_{t_n}(b_n)) \\ = j_{t_0} Z_{t_0, t_1}(a_1^\dagger Z_{t_1, t_2}(a_2^\dagger \cdots Z_{t_{n-1}, t_n} \\ (a_n^\dagger b_n) \cdots b_2) b_1), \end{aligned} \quad (2)$$

for any integer $n \geq 1$, any $t_0 \leq t_1 \leq \dots \leq t_n \in T$, and any $a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n \in \mathcal{B}$. When the canonical maps are also conditional expectations, there exists a one-parameter family $\{E_t\}_{t \in T}$ of conditional expectations mapping $\mathcal{A}$ to $\mathcal{A}_t$ for each $t \in T$, which are compatible with μ , such that $E_{s,t} = E_s|_{\mathcal{A}_t}$ for all $s < t \in T$. In terms of this one-parameter family, the Markov property can be stated as $E_s(\mathcal{A}_s) = \mathcal{A}_s$, $\forall s \in T$. The family $\{E_s\}_{s \in T}$ satisfies the following properties:

$$E_s(ab) = E_s(a)b, \forall a \in \mathcal{A}, b \in \mathcal{A}_s \quad (3)$$

$$\mu = \mu|_{\mathcal{A}_s} \circ E_s \quad (4)$$

$$E_s E_t = E_{s \wedge t}. \quad (5)$$

We can now define quantum Markov processes with conditional expectations, as follows:

Definition 2 $(\mathcal{A}, \{j_t\}_{t \in T}, \mu)$ is a quantum Markov process with conditional expectations if there exists a family of normal conditional expectations $\{E_t\}_{t \in T}$, mapping $\mathcal{A}$ to $\mathcal{A}_t$ for each $t \in T$, satisfying $E_s(\mathcal{A}_s) = \mathcal{A}_s \forall s \in T$ and (3)–(5).

The special structure of quantum Markov processes with conditional expectations entails that their associated multi-time correlation kernels (1) also have a special structure, which will be given in a theorem below. In a slightly different form (to be discussed in the next section), this result is known in the physics literature, in particular in quantum optics, as the “quantum regression theorem.”

Theorem 1 (Quantum regression theorem)

Let $t_n = (t_1, t_2, \dots, t_n) \in T^n$ with $t_1 \leq t_2 \leq \dots \leq t_n$. Then for any integer n , the time-ordered correlation kernels $w_{t_n}(a_n, b_n)$, with $a_n, b_n \in \mathcal{B}^n$, of a quantum Markov process with

conditional expectations are given by:

$$w_{t_n}(\mathbf{a}_n, \mathbf{b}_n) = \mu \circ j_{t_1}(a_1^\dagger Z_{t_1, t_2} a_2^\dagger \cdots Z_{t_{n-1}, t_n} (a_n^\dagger b_n) \cdots b_2) b_1). \quad (6)$$

We will now make the abstract notions presented above explicit by connecting them with concrete quantum Markov models from quantum optics. We consider a localized quantum system with a finite-dimensional Hilbert space $\mathfrak{h}$ coupled to a single propagating optical field. We assume the rotating wave approximation and Markov approximation on the coupling of the system and the field (Gardiner and Zoller 2004). The Hilbert space of the field is the boson (symmetric) Fock space $\mathfrak{F} = \Gamma_s(\mathcal{H})$ over the Hilbert space $\mathcal{H} = L^2(\mathbb{R}_{0+})$ of square-integrable complex functions on $\mathbb{R}_{0+}$; for details see (Hudson and Parthasarathy 1984; Parthasarathy 1992; Meyer 1995). We have the direct-sum decomposition $\mathfrak{F} = \mathbb{C} \oplus \bigoplus_{k=1}^{\infty} \mathcal{H}^{\otimes_s k}$ (where $\otimes_s$ denotes the symmetric tensor product), where $\mathcal{H}^{\otimes_s k}$ is the k -photon subspace. Also, for any positive integer n and $t_0 = 0 < t_1 < t_2 < \dots < t_n < t_{n+1} = \infty$, we have the tensor-product decomposition over time, $\mathfrak{F} = \bigotimes_{k=0}^n \mathfrak{F}_{[t_k, t_{k+1}]}$, where $\mathfrak{F}_{[t_k, t_{k+1}]} = \Gamma_s(\mathcal{H}_{[t_k, t_{k+1}]})$, and $\mathcal{H}_{[t_k, t_{k+1}]} = L^2([t_k, t_{k+1}])$ is the space of square-integrable functions on $[t_k, t_{k+1}]$.

The field has annihilation and creation densities, $b(t)$ and $b^\dagger(t)$, respectively, that satisfy the commutation relation $[b(t), b^\dagger(s)] = \delta(t-s)$ for all $s, t \geq 0$. For any $f \in \mathcal{H}$, let $B(f) = \int_0^\infty \overline{f(s)} b(s) ds$, $B^\dagger(f) = \int_0^\infty f(s) b^\dagger(s) ds$. Then we have the commutation relations, $[B(f), B(g)] = [B^\dagger(f), B^\dagger(g)] = 0$ and $[B(f), B^\dagger(g)] = \int_0^\infty \overline{f(s)} g(s) ds$. Let $|\Phi\rangle$ denote the vacuum state of the field, a state in which the field has no photons. In this state, $b(t)|\Phi\rangle = 0$ for all $t \geq 0$ and thus also $B(f)|\Phi\rangle = 0$. Define $B(t) = B(\mathbf{1}_{[0, t]})$ and $B^\dagger(t) = B^\dagger(\mathbf{1}_{[0, t]})$. Taking f informally as $f = \mathbf{1}_{[t, t+dt]}$, we can define the differentials $dB(t) = B(\mathbf{1}_{[t, t+dt]})$ and $dB^\dagger(t) = B^\dagger(\mathbf{1}_{[t, t+dt]})$. It follows that $dB(t)|\Phi\rangle = 0$, $\langle\Phi|dB(t)dB^\dagger(t)|\Phi\rangle = dt$, and $\langle\Phi|dB^\dagger(t)dB(t)|\Phi\rangle = \langle\Phi|dB(t)dB^\dagger(t)|\Phi\rangle =$

$\langle\Phi|dB^\dagger(t)dB^\dagger(t)|\Phi\rangle = 0$. This informally yields the Itô table in the vacuum state (see Hudson and Parthasarathy 1984; Parthasarathy 1992; Meyer 1995 for a rigorous treatment):

$\times$	dt	dB	dB^*
dt	0	0	0
dB	0	0	dt
dB^*	0	0	0

Take $T = \mathbb{R}_{0+}$ and let the system have Hamiltonian H . If the optical field is initialized in the vacuum state, the joint unitary propagator is given by the solution to a quantum stochastic differential equation (QSDE) (Here we do not consider a general QSDE as we do not include the so-called gauge or exchange process $\Lambda(t)$; see Hudson and Parthasarathy 1984 for details.) (Hudson and Parthasarathy 1984):

$$dU(t) = (-iH + (1/2)L^\dagger L)dt + dB^\dagger(t)L - L^\dagger dB(t)U(t),$$

with $U(0) = I$. Here $L : \mathfrak{h} \rightarrow \mathfrak{h}$ is a bounded system operator through which the system is coupled to the field. Let $\mathfrak{F}_t = \mathfrak{F}_{[0, t]}$ and $\mathfrak{F}_t = \mathfrak{F}_{[t, \infty)}$. Also, let $\mathcal{B}(h)$ denote the space of bounded linear operators over a Hilbert space h . The solution of the QSDE is *adapted*, meaning that for each $t \geq 0$ $U(t) = Z(t) \otimes I_{\mathfrak{F}_t}$ for some $Z(t) \in \mathcal{B}(\mathfrak{h}) \otimes \mathcal{B}(\mathfrak{F}_t)$, where $I_{\mathfrak{F}_t}$ is the identity operator on $\mathfrak{F}_t$. Note that $\mathcal{B}(\mathfrak{h}) \otimes \mathcal{B}(\mathfrak{F}_t)$ is a subalgebra of $\mathcal{B}(\mathfrak{h}) \otimes \mathcal{B}(\mathfrak{F})$ by identifying any element $W \in \mathcal{B}(\mathfrak{h}) \otimes \mathcal{B}(\mathfrak{F}_t)$ with its ampliation $W \otimes I_{\mathfrak{F}_t} \in \mathcal{B}(\mathfrak{h}) \otimes \mathcal{B}(\mathfrak{F})$.

Let $j_t(\cdot) = U(t)^\dagger(\cdot)U(t)$. For any system operator, X and $t \geq 0$, $j_t(X)$ is the evolution of X in the Heisenberg picture. Now, take $\mathcal{B} = \mathcal{B}(\mathfrak{h})$ and let $\mathcal{A}, \mathcal{A}_t$ be as defined previously (in terms of $\mathcal{B}$ and j_t). By the adaptedness of $U(t)$, $\mathcal{A}_t$ can be identified as a subalgebra of $\mathcal{B}(\mathfrak{h}) \otimes \mathcal{B}(\mathfrak{F}_t)$. We will next sketch how $(\mathcal{A}, \{j_t\}_{t \in T}, \mu)$ when μ is a normal state defined by $\mu(\cdot) = \text{Tr}(\rho \otimes |\Phi\rangle\langle\Phi| \cdot)$, where ρ is the initial density operator of the system, defines a quantum Markov process with conditional expectations.

Note that the vacuum state has the decomposition $|\Phi\rangle = |\Phi_t\rangle|\Phi_{[t]}\rangle$, with $|\Phi_t\rangle \in \mathfrak{F}_t$ and $|\Phi_{[t]}\rangle \in \mathfrak{F}_{[t]}$. For each $t \geq 0$, define the operator $E_t : \mathcal{A} \rightarrow \mathcal{B}(\mathfrak{h}) \otimes \mathcal{B}(\mathfrak{F}_{[t]})$ via the identity $\langle \eta' | E_t X | \eta \rangle = \langle \Phi_{[t]} | \langle \eta' | X | \eta \rangle | \Phi_{[t]} \rangle$ for any $X \in \mathcal{A}$ and $\eta, \eta' \in \mathfrak{h} \otimes \mathfrak{F}_{[t]}$ (Parthasarathy 1992, p. 214–215). Via the map E_t , we can then define $E_{[t]} : \mathcal{A} \rightarrow \mathcal{A}_{[t]}$ as $E_{[t]} = E_t \otimes I_{\mathfrak{F}_{[t]}}$. It can be verified that the family $\{E_{[t]}\}$ satisfies (3)–(5) (Parthasarathy 1992, p. 215) and is thus a family of conditional expectations. Furthermore, we can define $E_{s,t} = E_{s[} |_{\mathcal{A}_{[t]}}$ for all $s < t \in T$, and $\{E_{s,t}\}$ is a family of canonical maps.

In the quantum optics setting considered here, we can state the quantum regression theorem in a more explicit form, as it is usually found in the quantum optics literature. From the definition of $E_{s[}$, it is easily verified that $E_{s[}(B(t') - B(t)) = 0$ for any $t' > t > s$. Using this identity and the fact that $E_{s,t} j_t = E_{s[} |_{\mathcal{A}_{[t]}} j_t = E_{s[} j_t$, one can compute $Z_{s,t}(X) = j_s^* E_{s[} j_t(X)$ for any $X \in \mathcal{B}(\mathfrak{h})$ ($0 \leq s \leq t$). The main step is computing the differential (with respect to t , s fixed), $dZ_{s,t}(X) = j_s^* E_{s[} d j_t(X)$; see, e.g., Hudson and Parthasarathy (1984), Parthasarathy (1992) and Meyer (1995). This yields the differential equation

$$\frac{\partial}{\partial t} Z_{s,t}(X) = Z_{s,t}(\mathcal{L}(X)), \quad 0 \leq s \leq t,$$

with initial condition $Z_{s,s}(X) = X$. Here $\mathcal{L}$ the well-known Lindblad super-operator in the Heisenberg picture given by $\mathcal{L}(X) = \frac{1}{2} L^\dagger [X, L] + \frac{1}{2} [L^\dagger, X] L - i[X, H]$. Defining the adjoint map $Z_{s,t}^*$ via the duality $\text{Tr}(Y Z_{s,t}(X)) = \text{Tr}(Z_{s,t}^*(Y) X)$ for all $X, Y \in \mathcal{B}(\mathfrak{h})$, it follows that $Z_{s,t}^*$ satisfies the differential equation (for fixed s)

$$\frac{\partial}{\partial t} Z_{s,t}^*(X) = \mathcal{L}^*(Z_{s,t}^*(X)), \quad 0 \leq s \leq t,$$

with initial condition $Z_{s,s}^*(X) = X$, where $\mathcal{L}^*$ is the Lindblad super-operator in the Schrödinger picture, as the dual to $\mathcal{L}$, given by $\mathcal{L}^*(\rho) = \frac{1}{2} L \rho L_k^\dagger - \frac{1}{2} (L^\dagger L \rho + \rho L^\dagger L) + i[\rho, H]$. This equation has an explicit solution given by $Z_{s,t} = e^{\mathcal{L}^*(t-s)}$ for $t \geq s$. In terms of $Z_{s,t}^*$, the time-ordered correlation kernel (6) can be expressed as

$$w_{t_n}(a_n, b_n) = \text{Tr}(b_n Z_{t_{n-1}, t_n}^* (b_{n-1} \dots Z_{t_2, t_3}^* (b_2 Z_{t_1, t_2}^* (b_1 Z_{t_1, 0}^*(\rho) a_1^\dagger) a_2^\dagger) \dots) a_{n-1}^\dagger) a_n^\dagger). \quad (7)$$

The formula (7) is the familiar form of the quantum regression theorem found in the quantum optics literature (Gardiner and Zoller 2004, §5.2). Note the nested (or pyramidal) structure of (6) and (7) commensurate with the time ordering. This structure is fundamental in quantum theory and reflects the fact that probabilities for the outcomes of sequential measurements on a quantum system (by taking the operators a_k, b_k to be events in the algebra) generally depend on the order in which the measurements are performed; see Gardiner and Zoller (2004, §2.3) and the contribution of Gough in this volume. Also note that for a quantum Markov process with conditional expectations, the time-ordered correlations are completely determined by the reduced evolution on the system (with the field traced out), as given by (6) in the Heisenberg picture or (7) in the Schrödinger picture.

Summary and Future Directions

This review entry has provided a brief introduction to quantum mechanics as a quantum probability theory and the notion of quantum stochastic processes and quantum Markov processes in the quantum probabilistic setting. From the control engineering perspective, quantum probability theory provides a theoretical foundation for the analysis and feedback control of a large class of physical systems with well-defined input and outputs, which are ubiquitous in the field of quantum optics, quantum optomechanics, and superconducting circuits. Such systems are currently of interest as physical platforms for quantum technologies in sensing, communication, and computing.

Readers already familiar with probability theory can use this existing knowledge as a basis for understanding quantum probability theory and its applications. More recent efforts in quantum stochastics include analysis of the dynamics of quantum systems interacting with travelling

fields that are in highly non-Gaussian states such as single-photon states and multiphoton states, cat states, and a class of continuous matrix product states (Gough et al. 2011, 2013, 2012, 2014; Gough and Zhang 2015). An important quantum feedback operation in quantum information processing is quantum error correction (Gottesman 2009; Kerckhoff et al. 2010). Quantum feedback control will become more prominent as technology advances to a stage where more sophisticated quantum error correction codes can be experimentally demonstrated. Control of systems driven by fields in non-Gaussian states for applications such as quantum error correction will be an interesting direction for future research.

Cross-References

► [Conditioning of Quantum Open Systems](#)

Recommended Reading

For a historical overview of the development of quantum probability theory see Streater (2000). The text (Meyer 1995) provides an introduction to quantum probability theory for readers with a working knowledge of probability theory. Quantum stochastic calculus was developed in Hudson and Parthasarathy (1984) and comprehensive introductions can be found in Parthasarathy (1992) and Meyer (1995). The theory of quantum feedback networks based on quantum stochastic calculus can found in Gough and James (2009a, b), and for an overview that emphasizes physical perspectives of the network modelling see Combes et al. (2017). For an introduction to quantum filtering theory as a quantum version of stochastic filtering theory see the contribution by Gough in this volume and (Bouten et al. 2007) and the references therein. For optimal quantum feedback control and the separation principle see Bouten and van Handel (2008) and the references therein. For an introduction to linear quantum (stochastic) systems as a quantum analog of linear stochastic

systems and the modelling of linear quantum optical devices and their applications, see Nurdin and Yamamoto (2017).

References

- Accardi L, Frigerio A, Lewis JT (1982) Quantum stochastic processes. *Publ RIMS Kyoto Univ* 19:97–133
- Billingsley P (1986) Probability and measure, 2nd edn. Wiley series in probability and mathematical statistics. Wiley, New York
- Bingham NH (2000) Studies in the history of probability and statistics XLVI. Measure into probability: from Lebesgue to Kolmogorov. *Biometrika* 87(1):145–156
- Bouten L, van Handel R (2008) On the separation principle of quantum control. In: Belavkin VP, Guta M (eds) Quantum stochastic and information: statistics, filtering and control (University of Nottingham, 15–22 July 2006). World Scientific, Singapore, pp 206–238
- Bouten L, van Handel R, James MR (2007) An introduction to quantum filtering. *SIAM J Control Optim* 46:2199–2241
- Bratelli O, Robinson DW (1979) Operator algebras and quantum statistical mechanics. Texts and monographs in physics, vol I. Springer, New York
- Combes J, Kerckhoff J, Sarovar M (2017) The SLH framework for modeling quantum input-output networks. *Adv Phys X* 2(784)
- Gardiner CW, Zoller P (2004) Quantum noise: a handbook of Markovian and non-Markovian quantum stochastic methods with applications to quantum optics, 3rd edn. Springer, Berlin/New York
- Gottesman D (2009) An introduction to quantum error correction and fault-tolerant quantum computation, arXiv preprint: [Online] Available: <https://arxiv.org/abs/0904.2557>
- Gough J, James MR (2009a) Quantum feedback networks: Hamiltonian formulation. *Comm Math Phys* 287:1109–1132
- Gough J, James MR (2009b) The series product and its application to quantum feedforward and feedback networks. *IEEE Trans Automat Control* 54(11): 2530–2544
- Gough J, James MR, Nurdin HI (2011) Quantum master equation and filter for systems driven by field in a single photon state. In: Proceedings of IEEE conference on decision and control, pp 5570–5576
- Gough J, James MR, Nurdin HI, Combes J (2012) Quantum filtering for systems driven by fields in single-photon states or superposition of coherent states. *Phys Rev A* 86:043819
- Gough JE, Zhang G (2015) Generating nonclassical quantum input field states with modulating filters. *EPJ Quantum Technol* 2(15)
- Gough JE, James MR, Nurdin HI (2013) Quantum filtering for systems driven by fields in single pho-

- ton states and superposition of coherent states using non-markovian embeddings. *Quantum Inf Process* 12:1469–1499
- Gough JE, James MR, Nurdin HI (2014) Quantum trajectories for a class of continuous matrix product input states. *New J Phys* 16:075008
- Hudson RL, Parthasarathy KR (1984) Quantum Ito's formula and stochastic evolution. *Commun Math Phys* 93:301–323
- Kerckhoff J, Nurdin HI, Pavlichin D, Mabuchi H (2010) Designing quantum memories with embedded control: photonic circuits for autonomous quantum error correction. *Phys Rev Lett* 105:040502–1–040502–4
- Meyer PA (1995) *Quantum probability for probabilists*, 2nd edn. Springer, Berlin/Heidelberg
- Nurdin HI, Yamamoto N (2017) *Linear dynamical quantum systems: analysis, synthesis, and control*. Communications and control engineering. Springer, Cham/Switzerland
- Parthasarathy K (1992) *An introduction to quantum stochastic calculus*. Birkhauser, Berlin
- Shafer G, Vovk V (2006) The sources of Kolmogorov's Grundbegriffe. *Stat Sci* 21(1):70–98
- Streater RF (2000) Classical and quantum probability. *J Math Phys* 41(6):3556

Randomized Methods for Control of Uncertain Systems

Fabrizio Dabbene and Roberto Tempo†
CNR-IEIIT, Politecnico di Torino, Torino, Italy

Abstract

In this entry, we study the tools and methodologies for the analysis and design of control systems in the presence of random uncertainty. For analysis, the methods are largely based on the Monte Carlo simulation approach, while for design new randomized algorithms have been developed in the recent years by the systems and control community. These methods have been successfully employed in various application areas, which include systems biology; aerospace control; power systems; control of hard disk drives; high-speed networks; quantized, embedded, and electric circuits; structural design; and automotive and driver assistance.

Keywords

Chernoff bound · Hoeffding inequality · Monte carlo simulation · Randomized algorithms

Introduction

Randomized methods for control deal with the design of uncertain and complex dynamical systems. These methods have been originally developed for linear systems affected by structured uncertainty, usually expressed in the so-called $M - \Delta$ configuration. A similar approach may be followed when dealing with uncertainty in other contexts, such as uncertainty in the environment (random disturbances) or even when there is no uncertainty in the problem formulation, but the complexity of the problem is such that randomized methods may be the best approach, since these methods are known to break the curse of dimensionality; see Tempo et al. (2013) for details.

For the sake of simplicity, in this entry we consider an uncertain linear plant transfer function $P(s, q)$ affected by parametric uncertainty

$$q = [q_1 \dots q_\ell]^T$$

bounded in a set $\mathbb{Q} \subset \mathbb{R}^\ell$. The objective is to design the parameters $\theta \in \mathbb{R}^n$ of a controller transfer function $C(s, \theta)$ so to guarantee robustly some desired performance. This is reformulated as the problem of finding a design satisfying some *uncertain constraints* of the form

$$f(\theta, q) \leq 0 \text{ for all } q \in \mathbb{Q}.$$

In other words, the goal is to design a robust controller which satisfies the uncertain constraints.

Roberto Tempo: deceased.

Specific examples of these constraints include an $\mathcal{H}_\infty$ or $\mathcal{H}_2$ norm bound on the closed-loop sensitivity function or time-domain specifications.

Since this objective may be too hard to achieve in many situations, we are relaxing it as follows: we would like to design controller parameters $\theta \in \mathbb{R}^n$ such that a certain *violation* is allowed, i.e.,

$$\begin{cases} f(\theta, q) \leq 0 & \text{for all } q \in \mathbb{Q}_{\text{good}}; \\ f(\theta, q) > 0 & \text{for all } q \in \mathbb{Q}_{\text{bad}} \end{cases}$$

where the good and bad sets satisfy the equations

$$\begin{cases} \mathbb{Q}_{\text{good}} \cup \mathbb{Q}_{\text{bad}} = \mathbb{Q}; \\ \mathbb{Q}_{\text{good}} \cap \mathbb{Q}_{\text{bad}} = \emptyset, \end{cases}$$

and the goal is to guarantee that the bad set $\mathbb{Q}_{\text{bad}}$ is “small” enough. To state this concept more precisely, we assume that $q \in \mathbb{Q}$ is a random vector with given probability density function (pdf), and we introduce the probability of violation and the controller reliability.

Definition 1 (Probability of Violation and Reliability) The probability of violation for the controller parameters $\theta \in \mathbb{R}^n$ is defined as

$$V(\theta) \doteq \text{Prob} \{q \in \mathbb{Q} : f(\theta, q) > 0\}.$$

The reliability of the design $\theta \in \mathbb{R}^n$ is given by

$$R(\theta) = 1 - V(\theta).$$

In this context, we are satisfied if, given a violation level $\alpha \in (0, 1)$, the probability of violation is sufficiently small, i.e., $V(\theta) \leq \alpha$. We remark that relaxing the requirement of robust satisfaction of the uncertain constraints $f(\theta, q) \leq 0$ to a probabilistic one (by means of the probability of violation) does not help from a computational viewpoint because computing *exactly* the probability $V(\theta)$ is very hard in general, since it requires the solution of a multidimensional integral over the nonconvex domain defined by $f(\theta, q) > 0$, with $q \in \mathbb{Q} \subset \mathbb{R}^\ell$. The problem is then resolved introducing Monte Carlo randomized algorithms (formally defined in the next section). This is a

computational approach which leads to solutions which are often denoted as PAC (probably approximately correct) (Vidyasagar 2002).

More precisely, for fixed design $\theta \in \mathbb{R}^n$, to compute a Monte Carlo approximation based on N random simulations, we consider N independent identically distributed (iid) random samples of $q \in \mathbb{Q}$, called the *multisample*, of the uncertainty q according to the given probability density function

$$q^{(1\dots N)} = \{q^{(1)}, \dots, q^{(N)}\} \in \mathbb{Q}^N.$$

The cardinality N of the multisample $q^{(1\dots N)}$ is often referred to as the *sample complexity* (Vidyasagar 2001). The empirical violation of the design θ is then defined.

We note that in many cases, these samples are available as experimental data, and in this sense, randomized methods can be considered as a data-driven approach. In other situation, the multisample may be the outcome of a random sampling procedure.

Definition 2 (Empirical Violation) For given $\theta \in \mathbb{R}^n$, the empirical violation of $V(\theta, q)$ with respect to the multisample $q^{(1\dots N)} = \{q^{(1)}, \dots, q^{(N)}\} \in \mathbb{Q}^N$ is given by

$$\hat{V}_N(\theta, q^{(1\dots N)}) \doteq \frac{1}{N} \sum_{i=1}^N \mathbb{I}_f(\theta, q^{(i)})$$

where $\mathbb{I}_f(\theta, q^{(i)})$ is the indicator function

$$\mathbb{I}_f(\theta, q^{(i)}) \doteq \begin{cases} 0 & \text{if } f(\theta, q^{(i)}) \leq 0 \\ 1 & \text{otherwise.} \end{cases}$$

Monte Carlo Randomized Algorithms for Analysis

In this section, we study Monte Carlo randomized algorithms for analysis, i.e., when the controller parameters are fixed, and in particular we concentrate on a PAC computation of the probability of violation. In agreement with classical notions in computer science (Mitzenmacher and Upfal

2005; Motwani and Raghavan 1995), a randomized algorithm (RA) is formally defined as an algorithm that *makes random choices* during its execution to produce a result. This implies that, even for the same input data, the algorithm might produce different results at different runs, and, moreover, the results may be incorrect. Therefore, statements regarding properties of these algorithms are necessarily of probabilistic nature.

Formally, the probabilistic parameters ε , $\delta \in (0, 1)$, called *accuracy* and *confidence*, respectively, are introduced. For any θ , the PAC approach provides an empirical violation which is an approximation $\hat{V}_N(\theta, q^{(1\dots N)})$ to $V(\theta)$ within accuracy ε , and this event holds with confidence $1 - \delta$.

Monte Carlo Randomized Algorithm

Given a design $\theta \in \mathbb{R}^n$, a Monte Carlo randomized algorithm (MCRA) is a randomized algorithm that provides an approximation $\hat{V}_N(\theta, q^{(1\dots N)})$ to $V(\theta)$ based on the multisample $q^{(1\dots N)}$. Given accuracy ε and confidence δ , the approximation may be incorrect, i.e.,

$$\left| V(\theta) - \hat{V}(\theta, q^{(1\dots N)}) \right| > \varepsilon$$

but the probability of such an event is bounded, and it is smaller than δ .

In general, the results obtained by an MCRA as well as its running time would be different from one run to another since the algorithm is based on random sampling. As a consequence, the computational complexity of such an algorithm is usually measured in terms of its expected running times. MCRA are efficient because the expected running time is of polynomial order in the problem size (Tempo et al. 2013). One-sided and two-sided Monte Carlo randomized algorithms may also be defined (Tempo and Ishii 2007).

To derive the probabilistic properties of MCRA, we need to state the so-called Hoeffding inequality, which provides a bound on the error between the probability of violation and the empirical violation (Vidyasagar 2002).

Two-Sided Hoeffding Inequality

For fixed $\theta \in \mathbb{R}^n$ and $\varepsilon \in (0, 1)$, we have

$$\text{Prob} \left\{ q^{(1\dots N)} \in \mathbb{Q}^N : \left| V(\theta) - \hat{V}(\theta, q^{(1\dots N)}) \right| > \varepsilon \right\} \leq 2e^{-2N\varepsilon^2}.$$

For fixed accuracy ε , we observe that the right-hand side of this equation approaches zero exponentially. Furthermore, if we bound the right-hand side of this equation with confidence δ , we immediately obtain the classical (additive) Chernoff bound (Chernoff 1952) which is stated next.

Chernoff Bound

For any $\varepsilon \in (0, 1)$ and $\delta \in (0, 1)$, if

$$N \geq \frac{1}{2\varepsilon^2} \log \frac{2}{\delta}$$

then, with probability greater than $1 - \delta$, we have

$$\left| V(\theta) - \hat{V}(\theta, q^{(1\dots N)}) \right| \leq \varepsilon.$$

The Chernoff bound provides an indication of the required sample size, i.e., it provides the so-called sample complexity. More precisely, the sample complexity of a randomized algorithm is defined as the minimum cardinality of the multisample $q^{(1\dots N)}$ that needs to be drawn in order to achieve the desired accuracy ε and confidence δ . Notice that the confidence enters the Chernoff bound in a logarithmic fashion, while accuracy enters quadratically, and therefore, it is much more expensive computationally. Other large deviation inequalities and sample complexity bounds are discussed in the literature, including in particular the (multiplicative) Chernoff bound and the log-over-log bound for computing the so-called empirical maximum (Tempo et al. 1997). We refer to Vidyasagar (2002) for additional details.

Remark 1 (Las Vegas Randomized Algorithms) Las Vegas randomized algorithms (LVRA) are based on random samples generated according to a *discrete* probability density function, instead of a continuous pdf as in the case of Monte Carlo.

Therefore, contrary to MCRA, LVRA provide the “correct answer” with probability one because the entire search space can be fully explored. However, because of randomization, the running time of an LVRA is random (similarly to MCRA) and may be different in each execution. Hence, it is of interest to study the expected running time of the algorithm. It is noted that the expectation is with respect to the random samples generated during the execution of the algorithm and not to the problem data. Classical examples of LVRA within computer science include the well-known randomized quicksort (RQS) algorithm for ranking numbers, which is implemented in a C library of the UNIX operating system (Knuth 1998). Other more recent developments in systems and control regarding these algorithms are for the PageRank computation in the Google search engine (Ishii and Tempo 2010), consensus over large-scale networks (Fagnani and Zampieri 2008), localization and coverage control of robotic networks (Bullo et al. 2012), and opinion dynamics (Frasca et al. 2013). These problems are generally formulated in a graph theoretic setting consisting of nodes and links, and either the nodes or the links are randomly selected according to a given “local” protocol (often called gossip) based on a given discrete pdf.

Randomized Algorithms for Control Design

This section deals with control problems which require computing a design $\theta \in \mathbb{R}^n$ satisfying some probabilistic properties on the uncertain constraints $f(\theta, q)$. Two classes of problems, feasibility and optimization, are considered.

Feasibility Problem

Given uncertain constraints $f(\theta, q)$ and level $\alpha \in (0, 1)$, compute $\theta \in \mathbb{R}^n$ such that

$$V(\theta) = \text{Prob}\{q \in \mathbb{Q} : f(\theta, q) > 0\} \leq \alpha. \quad (1)$$

The second problem relates to the optimization of a linear function of the design parameters under probability constraints.

Optimization Problem

Given uncertain constraints $f(\theta, q)$, a linear objective function $c^T \theta$ and level $\alpha \in (0, 1)$, solve the constrained optimization problem

$$\begin{aligned} \min_{\theta} \quad & c^T \theta \\ \text{subject to} \quad & V(\theta) = \text{Prob}\{q \in \mathbb{Q} : f(\theta, q) > 0\} \\ & \leq \alpha. \end{aligned} \quad (2)$$

Optimization problems subject to constraints of the form $V(\theta) = \text{Prob}\{q \in \mathbb{Q} : f(\theta, q) > 0\} \leq \alpha$ are often called chance constraint optimization (Uryasev 2000).

Most of the algorithms that have been studied in the literature follow two main paradigms and are often based on the following convexity assumption.

Convexity Assumption

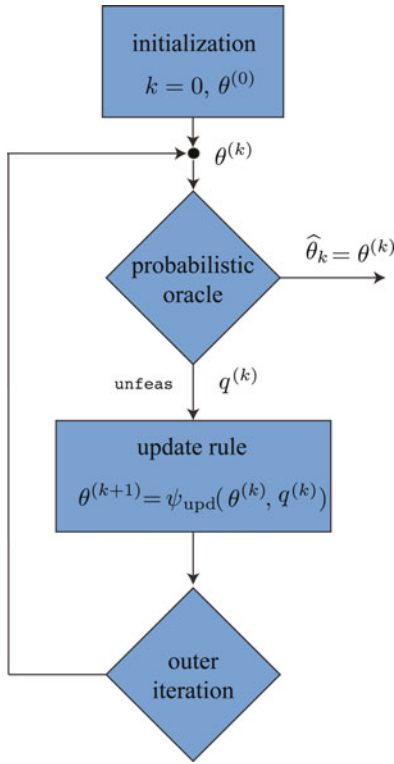
The uncertain constraint $f(\theta, q)$ is convex in θ for any fixed value of $q \in \mathbb{Q}$.

The two solution paradigms that have been proposed are now summarized. The algorithms have been implemented in the Toolbox RACT (Randomized Algorithms Control Toolbox) for probabilistic analysis and control design in the presence of uncertainty (Tremba et al. 2008).

Paradigm 1 (Sequential Approach)

Under the convexity assumption, we study the Feasibility Problem (1). The algorithms presented in the literature (see, e.g., Calafiore et al. (2011) for finding a probabilistic feasible design) follow a general iterative scheme (Fig. 1), which consists of successive randomization steps to handle uncertainty and optimization steps to update the design parameters. In particular, these algorithms share two fundamental ingredients:

1. A *probabilistic oracle* which performs a random check, with the objective to assess whether the probability of violation $V(\theta^{(k)})$ of the current candidate solution $\theta^{(k)}$, is smaller than a given level ρ and returns a *certificate* of unfeasibility, that is, a value $q^{(k)}$ such that $f(\theta^{(k)}, q^{(k)}) > 0$, when the candidate solution is found unfeasible



Randomized Methods for Control of Uncertain Systems, Fig. 1 Paradigm for sequential design consisting of probabilistic oracle and update rule

2. An *update rule* ψ_{upd} which exploits the convexity of the problem for constructing a new candidate solution $\theta^{(k+1)}$ based on the probabilistic oracle outcome

In this paradigm, the algorithm returns a design $\hat{\theta}_k$ such that

$$V(\hat{\theta}_k) = \text{Prob} \left\{ q \in \mathbb{Q} : f(\hat{\theta}_k, q) > 0 \right\} \leq \rho$$

is larger than $1 - \delta$. That is, the violation probability associated to the design $\hat{\theta}_k$ is smaller than the level ρ , and this event holds with large confidence $1 - \delta$.

Paradigm 2 (Scenario Approach)

Under the convexity assumption, we study the optimization problem (2). We remark that, even under these assumptions, solving this problem

is very hard computationally because the probabilistic constraint is nonconvex. To alleviate this difficulty, we reformulate problem (2) as a so-called scenario problem introduced in Calafiore and Campi (2006), which is now described.

For randomly extracted scenarios $q^{(1 \dots N)}$, this approach requires computing $\theta \in \mathbb{R}^n$ that solves the convex optimization problem subject to a finite number of sampled constraints

$$\hat{\theta}_N = \min_{\theta} c^T \theta \quad \text{subject to } f(\theta, q^i) \leq 0, \quad i = 1, \dots, N \quad (3)$$

In this paradigm, the algorithm returns in one-shot a design $\hat{\theta}_N$ and the sample complexity N such that

$$V(\hat{\theta}_N) = \text{Prob} \left\{ q \in \mathbb{Q} : f(\hat{\theta}_N, q) > 0 \right\} \leq \rho$$

is larger than $1 - \delta$. That is, the violation probability associated to the design $\hat{\theta}_N$ is smaller than the level ρ , and this event holds with large confidence $1 - \delta$.

Summary and Future Directions

Other probabilistic approaches have been proposed in the literature for control design, which are not based on the convexity assumption. A noticeable example is the strategy based on statistical learning theory (Valiant 1984; Vapnik 1998) which has the objective to design a controller without any convexity assumptions (Alamo et al. 2009). In particular, in Alamo et al. (2013), the general class of sequential probabilistic validation (SPV) algorithms has been introduced. A specific SPV algorithm tailored to scenario problems, providing a sequential scheme for dealing with the optimization problem, has been recently studied in Chamanbaz et al. (2013).

Cross-References

- [Markov Chains and Ranking Problems in Web Search](#)

- **Stability and Performance of Complex Systems Affected by Parametric Uncertainty**
- **Uncertainty and Robustness in Dynamic Vision**

Bibliography

- Alamo T, Tempo R, Camacho E (2009) A randomized strategy for probabilistic solutions of uncertain feasibility and optimization problems. *IEEE Trans Autom Control* 54:2545–2559
- Alamo T, Tempo R, Luque A, Ramirez D (2013) The sample complexity of randomized methods for analysis and design of uncertain systems. *arXiv:13040678* (accepted for publication)
- Bullo F, Carli R, Frasca P (2012) Gossip coverage control for robotic networks: dynamical systems on the space of partitions. *SIAM J Control Optim* 50(1):419–447
- Calafiore G, Campi M (2006) The scenario approach to robust control design. *IEEE Trans Autom Control* 51(1):742–753
- Calafiore G, Dabbene F, Tempo R (2011) Research on probabilistic methods for control system design. *Automatica* 47:1279–1293
- Chamanbaz M, Dabbene F, Tempo R, Venkataramanan V, Wang Q (2013) Sequential randomized algorithms for convex optimization in the presence of uncertainty. *arXiv:1304.2222*
- Chernoff H (1952) A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann Math Stat* 23:493–507
- Fagnani F, Zampieri S (2008) Randomized consensus algorithms over large scale networks. *IEEE J Sel Areas Commun* 26(4):634–649
- Frasca P, Ravazzi C, Tempo R, Ishii H (2013) Gossips and prejudices: ergodic randomized dynamics in social networks. In: *Proceedings of the 4th IFAC workshop on distributed estimation and control in networked systems*, Koblenz
- Ishii H, Tempo R (2010) Distributed randomized algorithms for the PageRank computation. *IEEE Trans Autom Control* 55:1987–2002
- Knuth D (1998) *The art of computer programming. Sorting and searching*, vol 3. Addison-Wesley, Reading
- Mitzenmacher M, Upfal E (2005) *Probability and computing: randomized algorithms and probabilistic analysis*. Cambridge University Press, Cambridge
- Motwani R, Raghavan P (1995) *Randomized algorithms*. Cambridge University Press, Cambridge
- Tempo R, Ishii H (2007) Monte Carlo and Las Vegas randomized algorithms for systems and control: an introduction. *Eur J Control* 13:189–203
- Tempo R, Bai EW, Dabbene F (1997) Probabilistic robustness analysis: explicit bounds for the minimum number of samples. *Syst Control Lett* 30:237–242
- Tempo R, Calafiore G, Dabbene F (2013) *Randomized algorithms for analysis and control of uncertain systems, with applications*. Communications and control engineering series, 2nd edn. Springer, London
- Tremba A, Calafiore G, Dabbene F, Gryazina E, Polyak B, Shcherbakov P, Tempo R (2008) RACT: randomized algorithms control toolbox for MATLAB. In: *Proceedings 17th IFAC world congress*, Seoul, pp 390–395
- Uryasev SP (ed) (2000) *Probabilistic constrained optimization: methodology and applications*. Kluwer Academic, New York
- Valiant L (1984) A theory of the learnable. *Commun ACM* 27(11):1134–1142
- Vapnik V (1998) *Statistical learning theory*. Wiley, New York
- Vidyasagar M (2001) Randomized algorithms for robust controller synthesis using statistical learning theory. *Automatica* 37:1515–1528
- Vidyasagar M (2002) *Learning and generalization: with applications to neural networks*, 2nd edn. Springer, New York

Realizations in Linear Systems Theory

Panos J. Antsaklis¹ and Alessandro Astolfi^{2,3}

¹Department of Electrical Engineering,
University of Notre Dame, Notre Dame, IN,
USA

²Department of Electrical and Electronic
Engineering, Imperial College London, London,
UK

³Dipartimento di Ingegneria Civile e Ingegneria
Informatica, Università di Roma Tor Vergata,
Rome, Italy

Abstract

When a state variable description of a linear system is known, then its input–output behavior can be easily realized by interconnecting simpler components. The problem of realization is the inverse of this process and can be formulated as follows: given an input–output description, such as the impulse response or the transfer function in the case of time-invariant systems, find a state variable description the impulse response or transfer function of which is the given one. Existence and minimality conditions are discussed. We are interested in realizations of minimum order, which is the case when the

realization is both reachable and observable. Realizations for both continuous-time and discrete-time systems are discussed.

review the key relations between the internal state variable and the external impulse response, or transfer function, descriptions.

Keywords

Reachability · Observability ·
Irreducibility · Minimality · Realizations

Introduction

The problem of system realization is described as follows. Given an external description of a linear system, specifically its impulse response (or its transfer function in the case of a time-invariant system), determine an internal state variable description that generates the given impulse response (or the transfer function). The name reflects the fact that if a (continuous-time) state variable description is known, an operational amplifier circuit can be easily built to realize (i.e., simulate) the system response. Before discussing the realization problem we

Continuous-Time Linear Systems

Consider a continuous-time system described by the equations

$$\dot{x} = A(t)x + B(t)u, \quad y = C(t)x + D(t)u, \quad (1)$$

where $x(t)$, the state vector, is a column vector of dimension n (i.e., $x(t) \in \mathbb{R}^n$), and $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ are the inputs and outputs of the system. The matrices $A(t) \in \mathbb{R}^{n \times n}$, $B(t) \in \mathbb{R}^{n \times m}$, $C(t) \in \mathbb{R}^{p \times n}$, and $D(t) \in \mathbb{R}^{p \times m}$ have entries which are continuous functions of t . For such a system the output response is given by

$$y(t) = C(t)\Phi(t, t_0)x_0 + \int_{t_0}^t H(t, \tau)u(\tau)d\tau, \quad (2)$$

where $\Phi(t, t_0)$ is the $n \times n$ transition matrix of $\dot{x} = A(t)x$, given by the Peano-Baker formula

$$\Phi(t, t_0) = \begin{cases} I + \int_{t_0}^t A(\sigma_1)d\sigma_1 + \int_{t_0}^t A(\sigma_1) \int_{t_0}^{\sigma_1} A(\sigma_2)d\sigma_2 d\sigma_1 + \cdots & t > t_0, \\ I & t = t_0, \end{cases} \quad (3)$$

$x(t_0) = x_0$ is the initial condition, and $H(t, \tau)$ is the $p \times m$ impulse response matrix given by

$$H(t, \tau) = \begin{cases} C(t)\Phi(t, \tau)B(\tau) + D(t)\delta(t - \tau) & t \geq \tau, \\ 0 & t < \tau, \end{cases} \quad (4)$$

where $\delta(t - \tau)$ is the impulse (delta or Dirac) function applied at time $t = \tau$. Recall that $H(t, \tau)$ denotes the response at time t when an impulse input is applied at time $\tau \leq t$ assuming zero initial conditions.

In the time-invariant case, the equations (1) become

$$\dot{x} = Ax + Bu, \quad y = Cx + Du, \quad (5)$$

and the output response is given by

$$y(t) = Ce^{At}x_0 + \int_0^t H(t, \tau)u(\tau)d\tau, \quad (6)$$

where, without loss of generality, the initial time t_0 has been set to zero. In this case, the impulse response is

$$H(t, \tau) = \begin{cases} Ce^{A(t-\tau)}B + D\delta(t-\tau) & t \geq \tau, \\ 0 & t < \tau. \end{cases} \quad (7)$$

Note that time invariance implies that $H(t, \tau) = H(t - \tau, 0)$, hence τ , which is the time the impulse input is applied to the system, can be set to zero, again without loss of generality, to define $H(t, 0)$. The transfer function of the system is the (one-sided or unilateral) Laplace transform of $H(t, 0)$, namely

$$H(s) = \mathcal{L}[H(t, 0)] = C(sI - A)^{-1}B + D. \quad (8)$$

A realization of $H(t, \tau)$ is any state variable description (1), that is, any quadruple $\{A(t), B(t), C(t), D(t)\}$, the impulse response of which is $H(t, \tau)$, that is, such that (4) is satisfied. Similarly, in the time-invariant case a realization is any quadruple $\{A, B, C, D\}$ such that (7) is satisfied.

In the time-invariant case, a realization is commonly defined in terms of the transfer function matrix $H(s)$. Then a realization of $H(s)$ is any state variable description (5), the transfer function of which is $H(s)$, that is, (8) is satisfied. There are alternative conditions under which a quadruple $\{A, B, C, D\}$ gives a realization of some $H(s)$. To this end, expand $H(s)$ in a Laurent series to obtain

$$H(s) = H_0 + H_1s^{-1} + H_2s^{-2} + \dots \quad (9)$$

The matrices H_i , $i = 0, 1, 2, \dots$ are called the *Markov parameters* of the system and can be determined as follows:

$$H_0 = \lim_{s \rightarrow \infty} H(s), \quad H_1 = \lim_{s \rightarrow \infty} s(H(s) - H_0),$$

and, in general,

$$H_k = \lim_{s \rightarrow \infty} s^k \left(H(s) - \sum_{i=0}^{k-1} H_i s^{-i} \right), \quad k \geq 1.$$

From the definition of Markov parameters, one can conclude that a quadruple $\{A, B, C, D\}$ is a realization of $H(s)$ if and only if

$$H_0 = D, \quad H_i = CA^{i-1}B, \quad i = 1, 2, \dots \quad (10)$$

We now provide conditions for the existence of a realization given either $H(t, \tau)$ or $H(s)$. Note that if a realization does exist then there are infinitely many realizations, as detailed in what follows.

It can be shown that $H(t, \tau)$ is realizable as the impulse response of a system described by the equations (1) if and only if $H(t, \tau)$ can be decomposed as

$$H(t, \tau) = M(t)N(\tau) + D(t)\delta(t - \tau), \quad (11)$$

for $t \geq \tau$, where $M(t)$, $N(t)$, and $D(t)$ are $p \times n$, $n \times m$, and $p \times m$ matrices, respectively, with continuous real-valued entries and with n finite. If in addition to (11), $M(t)$ and $N(t)$ are differentiable and

$$H(t, \tau) = H(t - \tau, 0), \quad (12)$$

then $H(t, \tau)$ is realizable as the impulse response of a system described by the equations (5).

In the time-invariant case, it is more common to work with a given transfer function $H(s)$. Then $H(s)$ is realizable as the transfer function matrix of a time-invariant system described by the equations (5) if and only if $H(s)$ is a matrix of rational functions and satisfies

$$\lim_{s \rightarrow \infty} H(s) < \infty, \quad (13)$$

that is, if and only if $H(s)$ is a *proper rational matrix* or, equivalently, if and only if

$$\lim_{s \rightarrow \infty} H(s) = D, \quad (14)$$

with D constant.

We are interested in realizations (5) of a given transfer function matrix $H(s)$ of least order n (i.e., $A \in \mathbb{R}^{n \times n}$) called minimal or irreducible realizations of $H(s)$. The following two results completely solve the minimal realization problem.

Theorem 1 *An n -dimensional realization $\{A, B, C, D\}$ of $H(s)$ is minimal (irreducible, or of least order) if and only if it is both reachable and observable.*

Note that if (A, B) is not reachable then by separating the reachable and unreachable parts of the system by an equivalence transformation and taking only the reachable part one can still obtain $H(s)$, because the unreachable part of the system does not contribute to $H(s)$. Similar considerations apply with respect to the observability property.

Hence, reachability and observability are necessary conditions for minimality. It can be shown that they are also sufficient.

Theorem 2 *If a minimal realization of order n is found, then any other minimal realization may be obtained via an equivalence transformation.*

Specifically, if $\{A, B, C, D\}$ and $\{\bar{A}, \bar{B}, \bar{C}, \bar{D}\}$ are realizations of $H(s)$ and $\{A, B, C, D\}$ is minimal, then $\{\bar{A}, \bar{B}, \bar{C}, \bar{D}\}$ is also minimal if and only if there exists a nonsingular matrix T such that

$$\begin{aligned}\bar{A} &= TAT^{-1}, & \bar{B} &= TB, \\ \bar{C} &= CT^{-1}, & \bar{D} &= D.\end{aligned}\quad (15)$$

Discrete-Time Linear Systems

The definitions and results for the discrete-time case are completely analogous to the ones for the continuous-time case. They are summarized below for completeness.

Consider a discrete-time system described by the equations

$$\begin{aligned}x(k+1) &= A(k)x(k) + B(k)u(k), \\ y(k) &= C(k)x(k) + D(k)u(k).\end{aligned}\quad (16)$$

Its output response is given by

$$y(k) = C(k)\Phi(k, k_0)x_0 + \sum_{i=k_0}^{k-1} H(k, i)u(i), \quad k > k_0, \quad (17)$$

where $x_0 = x(k_0)$, $\Phi(k, k_0)$ is the $n \times n$ transition matrix, and $H(k, i)$ is the $p \times m$ discrete impulse (pulse) response, that is

$$\Phi(k, k_0) = \begin{cases} A(k-1) \cdots A(k_0) & k > k_0, \\ I & k = k_0, \end{cases} \quad (18)$$

$$H(k, i) = \begin{cases} C(k)\Phi(k, i+1)B(i) & k > i, \\ D(k) & k = i, \\ 0 & k < i. \end{cases} \quad (19)$$

Note that the definition of $\Phi(k, k_0)$ for $k < k_0$ requires invertibility of $A(k)$ for all k .

In the time-invariant case, equations (16) and (17) become

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k) + Du(k),\end{aligned}\quad (20)$$

and

$$y(k) = CA^k x_0 + \sum_{i=0}^{k-1} H(k, i)u(i), \quad k > 0, \quad (21)$$

where, without loss of generality, k_0 has been set to zero.

The discrete impulse (pulse) response is now given by the equations

$$H(k, i) = \begin{cases} CA^{k-(i+1)}B & k > i, \\ D & k = i, \\ 0 & k < i. \end{cases} \quad (22)$$

Since the system is time invariant, $H(k, i) = H(k-i, 0)$ and i , the time the pulse input is applied, can be set to zero. The transfer function

matrix for (20) is the (one-sided or unilateral) z -transform of $H(k, 0)$, namely

$$H(z) = \mathcal{Z}\{H(k, 0)\} = C(zI - A)^{-1}B + D. \quad (23)$$

It can be shown that a given $p \times m$ matrix $H(k, i)$, with $k \geq i$, is realizable as the pulse response of a system described by the equations (16) if and only if $H(k, i)$ can be decomposed as

$$H(k, i) = \begin{cases} M(k)N(i) & k > i, \\ D(k) & k = i. \end{cases} \quad (24)$$

If, in addition, $H(k, i) = H(k - i, 0)$, then it is realizable via a time-invariant description of the form (20). Similarly to the continuous-time case, $H(z)$ is realizable as the transfer function matrix of a system described by the equations (20) if and only if it is a matrix of rational functions and satisfies

$$\lim_{z \rightarrow \infty} H(z) < \infty. \quad (25)$$

A realization (20) of $H(z)$ is minimal if and only if it is reachable and observable. Finally, if (20) is a minimal realization of $H(z)$, then any other minimal realization is equivalent to (20).

Realization Algorithms

Given a transfer function $H(s)$ (or $H(z)$), we are interested in finding a minimal (irreducible, or of least order) realization of the form (5) (or (20)). First note that there are methods to determine the order n of a minimal realization directly from $H(s)$, via the characteristic polynomial and notions such as the McMillan degree of $H(s)$ or via the Markov parameters of $H(s)$ and the Hankel matrix. This can be done without finding a minimal realization. Knowing the order of a minimal realization in advance is useful as it provides a guide as to what we should expect when we determine an actual realization. Details may be found in the references provided.

In special cases it is possible that the realization algorithm results directly in a reachable and observable, and therefore minimal, realization. It is more common, however, for the algorithm to result in a realization that is either reachable or observable, but not both. In such cases, an extra step is needed to isolate the unreachable/unobservable part of the realization and take only the part that it is both reachable and observable. The reader should consult any of the references for detailed descriptions of several realization algorithms.

To illustrate the above discussion we present a simple example. Consider the transfer function

$$H(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

and assume that numerator and denominator are coprime, that is, they do not have common roots. A realization is given by the matrices

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$$C = [b_0, \quad b_1, \quad b_2], \quad D = [0].$$

This realization is reachable (the pair (A, B) has a form called *controller form*) and observable and, therefore, it is a minimal realization of $H(s)$. This algorithm easily generalizes to the case in which the degree of the denominator of $H(s)$ is n (in this example it is 3). Note that if $\lim_{s \rightarrow \infty} H(s) = D \neq 0$, then one has to apply the previous algorithm to $\hat{H}(s) = H(s) - D$ to obtain A , B , and C .

Summary

The state variable realization of impulse responses and transfer functions was one of the early problems addressed by system theory. Its solution provides clear understanding of the relations between external (input-output) and internal descriptions of systems. A key result is that any minimal order realization is controllable and observable. Many realization algorithms

may be found in the literature. Extensions to polynomial matrix descriptions can also be found in the literature, as well as extensions to partial realizations.

Cross-References

- ▶ [Linear Systems: Continuous-Time Impulse Response Descriptions](#)
- ▶ [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)
- ▶ [Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions](#)
- ▶ [Linear Systems: Discrete-Time Impulse Response Descriptions](#)
- ▶ [Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions](#)
- ▶ [Linear Systems: Discrete-Time, Time-Varying, State Variable Descriptions](#)

Recommended Reading

A clear understanding of the relationship between external and internal descriptions of systems is one of the principal contributions of systems theory. This topic was developed in the early 1960s with original contributions from Gilbert (1963) and Kalman (1963). The role of controllability and observability in minimal realization is due to Kalman (1963); see also Kalman et al. (1969). For extensive historical comments, see Kailath (1980). The time-varying case is treated in Brockett (1970), Antsaklis and Michel (2006), and Rugh (1996).

Bibliography

- Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston
- Brockett RW (1970) Finite dimensional linear systems. Wiley, New York
- Gilbert E (1963) Controllability and observability in multivariable control systems. *SIAM J Control* 1:128–151
- Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs
- Kalman RE (1963) Mathematical description of linear systems. *SIAM J Control* 1:152–192

- Kalman RE, Falb PL, Arbib MA (1969) Topics in mathematical system theory. McGraw-Hill, New York
- Moore BC (1981) Principal component analysis in linear systems: controllability, observability and model reduction. *IEEE Trans Autom Control* AC-26:17–32
- Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs

Real-Time Optimization of Industrial Processes

Jorge Otávio Trierweiler

Group of Intensification, Modelling, Simulation, Control and Optimization of Processes (GIMSCOP), Department of Chemical Engineering, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, RS, Brazil

Abstract

RTO aims to optimize the operation of the process taking into account economic terms directly. There are several fundamental gears for smooth operating of an RTO solution. The RTO loop is an extension of feedback control system and consists of subsystems for (a) steady-state detection, (b) data reconciliation and measurement validation, (c) process model updating, and (d) model-based optimization followed by solution validation and implementation. There are several alternatives for each one of these subsystems. This contribution introduces some of the currently used approaches and gives some perspectives for future works in this area.

Keywords

Data reconciliation · Model updating · Online optimization · Parameter selection · Steady-state detection

Synonyms

[RTO](#)

Introduction

Effectiveness, efficiency, product quality, process safety, and low environmental impact are the main driving forces for the improvement of the operation of industrial processes. Real-time (or online) optimization (RTO) is one of the options that are available to achieve these goals and is attracting considerable industrial interest due to its direct and indirect benefits.

RTO systems are model-based, closed-loop process control systems whose objective is to maintain the process operation as nearly as possible to the optimal operating point. Such RTO systems use rigorous process models and current economic information to predict the optimal process operating conditions. Additionally, RTO can mitigate and reject long-term disturbances and performance losses (e.g., due to fouling of heat exchangers or deactivation of catalysts).

The direct benefit from applying RTO is improving the economic performance in terms of increasing the profit of the plant and reducing energy consumption and pollutant emissions. These are also called the *online benefits*. The indirect benefits result from the tools used in the implementation of RTO. For instance, a better understanding of the processes can be employed to debottleneck the plant and to reduce operating difficulties. In addition, abnormal measurement information obtained from gross error detection can help instrumentation and process engineers to troubleshoot the plant instrument errors. Parameter estimation is very useful for process engineers to evaluate the equipment conditions and to identify decreasing efficiencies and other sources of problems. Furthermore, the detailed process simulation of the model used in online optimization can be used for process monitoring and serve as a training tool for new operators. Finally, the rigorous process model can be used for process maintenance, advanced process control, process design, facility planning, and process monitoring.

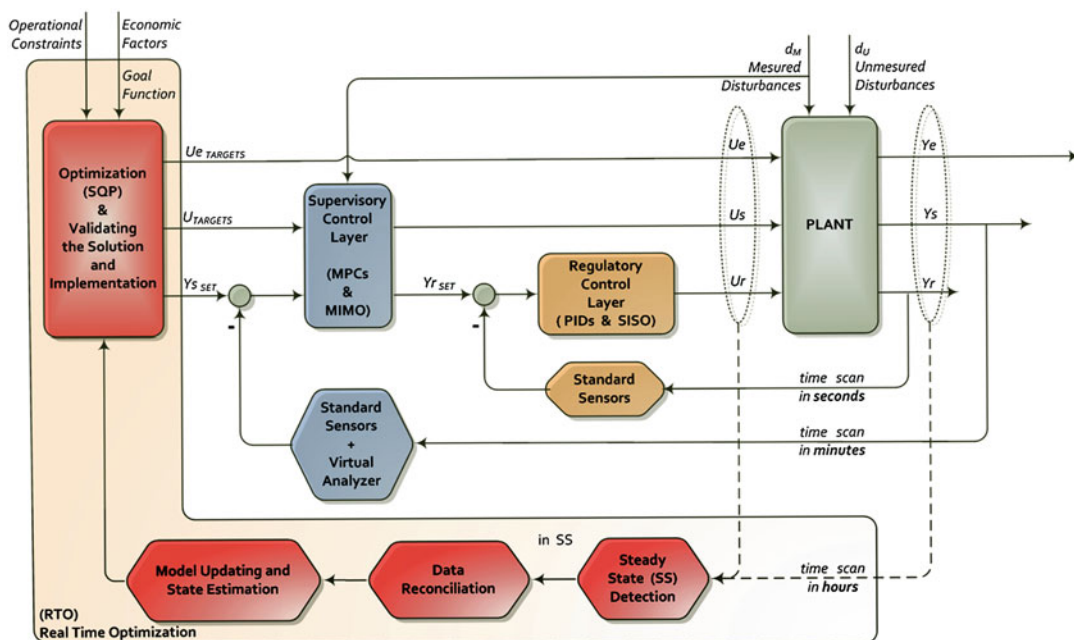
Real-time optimization (RTO) solutions have been developed since the early 1980s, and nowadays there are many petrochemical and chemical applications, especially in the production of

ethylene and propylene in fluid catalytic cracking units (FCCUs) (Darby et al. 2011). Other successful industrial applications are mentioned in Alkaya et al. (2009) with the respective economic returns.

Control Layers and the RTO Concept

Usually the process control is stratified into several layers, which have different response times and control objectives. RTO is located in an intermediate layer that provides the connection between plant scheduling (medium-term planning) and the control system (short-term process performance). In a plant control hierarchy, process disturbances are controlled using process controllers, whereas the RTO system must track changes in the optimum operating conditions caused by low-frequency process changes (e.g., raw material quality and composition, catalyst deactivation).

The typical structure of an RTO system is shown in Fig. 1, which depicts the elements of the closed-loop system. The RTO loop is an extension of the feedback control system and consists of subsystems for (a) steady-state detection, (b) data reconciliation and measurement validation, (c) process model updating, and (d) model-based optimization followed by solution validation and implementation. Once the plant operation has reached a steady state, the plant data ($y = [Y_e, Y_s, Y_r]$) are gathered and validated to detect and correct gross errors in the process measurements, and at the same time the measurements may be reconciled using material and energy balances to ensure that the data set used for model updating is consistent. These validated measurements are used to estimate model parameters (θ) to ensure that the model represents the plant faithfully at the current operating point. Then, the optimum controller set points (Y_{sSET}) and manipulated targets ($U_{TARGETS}$ and $U_{eTARGETS}$) are calculated using the updated model and are transferred to the advanced process controllers after they have been validated to be effectively applied.



Real-Time Optimization of Industrial Processes, Fig. 1 Basic structure of the traditional RTO

Each layer in Fig. 1 has its own specific tasks as discussed in the following:

1. **Regulatory layer.** This layer is focused on basic (e.g., temperature, flow rate) and inventory (e.g., level and pressure) control ensuring safety and operational stability for the industrial plant. The holdups of vapors and gases are measured by pressure sensors, while the holdups of liquids and solids are measured by level and weighing sensors. In the case of unstable processes, the regulatory layer is also responsible for their stabilization, e.g., by temperature control of industrial reactors. No industrial process can operate without this control layer. The typical operation time scan is in the order of seconds. For its design, typical questions that have to be answered are the following: “How to ensure safe unit operation?” “How to quickly meet the demands coming from the supervisory layer or from the operators?” “How to prevent disturbances to propagate throughout the plant?” The control technology that prevails in this layer is SISO (single-input-single-output) PID controllers, with very few cases

where the derivative action is effectively employed. In Fig. 1, Y_r are the controlled variables of this layer (e.g., levels, flow rates, pressures, temperatures, pH), and U_r are the corresponding manipulated variables, typically control valves.

2. **Supervisory layer.** This layer is concerned with the quality of the final product. The goal is to ensure the specifications without infringing the operating limits of the equipment. Typically, in this layer there is a strong interaction between the controlled variables, requiring tailored multi-look control structures or the use of multivariate control techniques. The dominant advanced technology in this layer is model predictive control (MPC). In this layer the calculations and updates are performed on the time scale of minutes and the typical associated question is “how to ensure the quality of the final product while satisfying the operating constraints and improving the profitability by reducing the variability of the product parameters?” Here the controlled variables (Y_s) are usually related to the product quality and the manipulated variables are the set points for the regulatory layer Y_{rSET} and additional

manipulated variables (Us) not used by the regulatory layer (e.g., variable frequency drive).

3. **RTO layer.** Here the main focus is the profitability of the process. Specifications and operating points (i.e., set points and targets for the manipulated variables) are determined by solving an optimization problem that aims at maximizing the profitability of the process under stationary conditions. When the optimal operating point is close to the operational limits, the real-time optimization is quite straightforward, since it is enough to take the process to these limits, which is usually done by solving a linear programming (LP) optimization problem. Such simple solutions are effective especially in cases where it is known that to maximize or to minimize the flow rate of a given stream will maximize the profitability. As this kind of solution can be easily implemented, most commercial predictive controllers already have an LP or QP layer integrated, using as a model the gain of the dynamic model used in the MPC. However, for processes with large recycling streams and pronounced nonlinearity, this type of solution is not enough to bring the system to its optimal operating conditions. In this case, it is essential to use a nonlinear optimizer that aims at driving the system to operate in the best operating region. When the industrial process works essentially in steady state, the problem can be solved using stationary models. The solutions offered on the market typically involve the use of a stationary process simulator (e.g., Aspen Plus, PRO II). The RTO sampling times are in the time scale of hours and the questions to be answered are the following: “What is the best way of operation?” “How to increase the profitability of the process?” “How to decrease energy consumption and to increase the process efficiency?”

Four Elements of Classical Real-Time Optimization

A standard RTO solution requires that all four calculation blocks illustrated in Fig. 1 work

together smoothly. In fact, each block can be formulated as an optimization problem by itself. Sometimes these optimization problems are combined together. Below the alternative techniques that can be applied to each of these subsystems are discussed.

Steady-State Detection (SSD)

As indicated in Fig. 1, the RTO loop execution begins with the detection of a steady state. Identifying a steady state may be difficult because process variables are noisy and measurements do not settle at a constant value. Being at a steady state can be defined as an acceptable constancy of the measurements over a given period of time. Therefore, tests for stationarity are commonly based on checking the constancy of the measured quantities.

Mejía et al. (2010) compared 6 different approaches to SSD using 5,760 simulated data sets. They concluded that the method based on the estimation of the absolute value of the first and the second derivatives defined by

$$I_{DER} = \left| \frac{\widehat{dy}}{dx} \right| + 10 \left| \frac{\widehat{d^2y}}{dt^2} \right| \quad (1)$$

gives the best results. Although this idea is quite simple, as being at a steady state means zero derivatives by definition, it has some implementation issues, due to signal noise and outliers. These problems can be reduced by smoothing the plant data using smoothing splines, noncausal Butterworth filters, or wavelet decompositions. The second best compared approach was the local autocorrelation (Mejía et al. 2010) followed by the two statistical nonparametric tests of independence hypothesis proposed by Bebar (2005) and by the method of Cao and Rhinehart (1995).

Data Reconciliation (DR)

Within the mathematical models of industrial processes, the balance equations that result from conservation laws of mass, energy, etc., are the core that cannot be subject to debate. If the measured data do not satisfy the balance equations, this fact must be attributed to measurement errors or to fundamental model inadequacies. Ruling

out the latter, as measurement errors are always present, before using the measured data, they should be adjusted to obey the conservation laws and other constraints, e.g., of their ranges. The adjustment using optimization techniques combined with the statistical theory of errors is called data reconciliation. Unfortunately the adjustment of all variables can be greatly affected by “gross errors” in one variable, so such errors must be detected.

The relationship between a measurement of a variable and its true value can be represented by

$$y_m = y + \overbrace{e_r + e_g}^{\text{error}} \quad (2)$$

where y_m and y are the measured and true values, while e_r and e_g are the random and gross errors, respectively. The random errors (e_r) are assumed to be zero mean and normally distributed (Gaussian), since they are the result of the simultaneous effect of several causes. The gross errors (e_g) are caused by large nonrandom events. They can be subdivided into measurement-related errors, such as malfunctioning sensors (e.g., incorrect calibration, sensor degradation, or damage to the electronics), and processes-related errors, such as process leaks.

In the absence of gross errors, the simplest version of data reconciliation can be stated as a quadratic programming (QP) problem

$$\min_y \frac{1}{2} (y - y_m)^T Q^{-1} (y - y_m) \quad (3)$$

subject to the linear or linearized constraint related to the process model, written as

$$A \cdot y - c = 0.$$

The covariance matrix (Q), which is usually diagonal, captures the variance of the sensors and is responsible to distribute the errors among the measurements (y_m). The solution of this problem is the reconciled value that for this simple case is given analytically by

$$y = \left[I - QA^T(AQA)^{-1}A \right] y_m + QA^T(AQA)^{-1}c.$$

A rigorous formulation of the reconciliation problem is possible even with nonlinear constraints; only the general existence and uniqueness of a solution is not warranted theoretically.

Several statistical tests have been constructed for the detection of gross errors. Some of them are based in the distribution of the constraint residuals, i.e., $r_c = A \cdot y_m - c$, and others are based on the distribution of the estimated error after the reconciliation procedure, i.e., $\hat{e} = y_m - y$. The evaluation of r_c does not require solving previously the associated data reconciliation problem. For a complete discussion and review, see Narasimhan and Jordache (1999) and Sequeira et al. (2002).

Model Updating

A key, yet difficult, decision in model parameter adaptation is to select the parameters that are adapted. These parameters should be identifiable and represent actual changes in the process, and their adaptation should help to approach the true process optimum. Clearly, the smaller the subset of parameters, the better the confidence in the parameter estimates and the lower the required excitation (also the better the regularization effect). But too few adjustable parameters can lead to misleading models and thus wrong proposals for operational changes.

In general, the parameter estimation and updating are limited not only by the lack of information available from experimental data but also by the correlation between the parameters that are identified. The estimation of correlated parameters leads to a high degree of uncertainty in the model, since different combinations of parameter values lead to the same value of the objective function in the estimation problem.

The selection of the right number of parameters to be identified can be done by the analysis of the sensitivity matrix (S). The elements of S , s_{ij} , are the partial derivatives of the measurement y_i

with respect to the parameter θ_j evaluated at the current value of the parameter (θ_0), i.e.,

$$s_{ij} = \left(\frac{\partial y_i}{\partial \theta_j} \right) \Big|_{\theta=\theta_0}. \quad (4)$$

In general, different parameters and measurements have distinct magnitudes. Therefore, scaling is a key issue that has a strong impact on the results. Traditionally each element of the matrix S is scaled by the initial guess θ_{0j} of the parameter and by the average value of the measurement i ($\bar{y}_{si}$). The scaled elements $s_{s,ij}$ are then given by

$$s_{s,ij} = \left(\frac{\theta_{0j}}{\bar{y}_{si}} \right) \left(\frac{\partial y_i}{\partial \theta_j} \right) \Big|_{\theta=\theta_0} \quad (5)$$

This scaling procedure has some problems, once it requires both a good initial guess for the parameters and representative average values for the measurements. But the main drawback is that it does not consider the multivariable nature existing among all parameters and outputs. To solve these drawbacks, Botelho et al. (2012) proposed to apply diagonal scaling matrices L and R that result from the solution of the convex optimization problem to find the minimized condition number of the sensitivity matrix, $\gamma(LSR)$, i.e.,

$$\begin{aligned} \min_{L,R} \gamma(LSR) \\ \text{s.t. } L \in \mathbb{R}^{n_y \times n_y}, \text{ diagonal and nonsingular} \\ R \in \mathbb{R}^{n_\theta \times n_\theta}, \text{ diagonal and nonsingular} \end{aligned} \quad (6)$$

This convex optimization problem can be solved using the LMI (linear matrix inequality) approach as described by Boyd et al. (1994). With the optimized scaling matrices L and R , the scaled sensitivity matrix S_s is given by

$$S_s = LSR \quad (7)$$

With S_s , the best subset of parameters to be estimated can be determined using the non-square relative gain array matrix (NSRGA) as also proposed by Botelho et al. (2012). The NSRGA

can be easily calculated for the scaled sensitivity matrix by

$$NSRGA(S_s) \stackrel{\text{def}}{=} S_s \circ (S_s^\dagger)^T \quad (8)$$

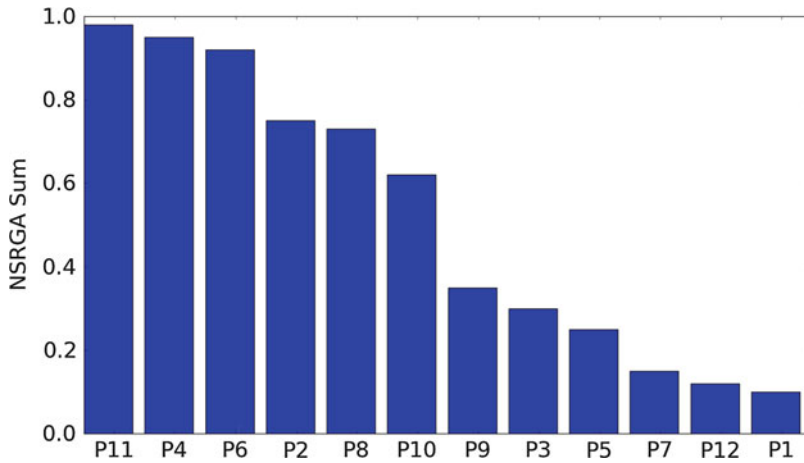
where $(S_s^\dagger)^T$ is the transpose of the pseudo-inverse of S_s and $\circ$ is the entrywise product (also known as the Hadamard or element-wise product). The rows of $NSRGA(S_s)$ are related to the output measurements, whereas the columns are related to the parameters. The sum of the values in each column, whose values can vary between 0 and 1, reflects the relevance of each parameter, and it can be used to sort in descending order their influence on the outputs. When the sum of a column is close to 1, the corresponding parameter has a small correlation with the other ones and a strong influence on the output measurements.

Figure 2 illustrates the typical ordering produced by sorting the $NSRGA(S_s)$ in descending order. Thus, it is possible to have an idea of which parameters should be selected for estimation. The values presented in this figure suggest that the parameters $P11$ and $P4$ have very small correlation with the others and should be selected as updated parameters, whereas the parameters $P12$ and $P1$ show the opposite behavior and should not be updated.

Solving the Optimization Problem

Nonlinear programs for RTO can be formulated using models of different complexities. For example, RTO can be based on process models similar to those used for design and analysis, using commercial simulators (e.g., Aspen Plus, PRO II, HYSYS, etc.). On the other hand, because these problems need to be solved at regular intervals (at least every few hours), detailed simulation models can be partly replaced by correlations or operating curves that are fitted to the process and updated on a longer time scale.

If a rigorous process model is used, the number of nonlinear equations can be very large. The model is usually built by linking smaller sub-models. The optimization problem can be formulated as the following nonlinear programming problem (NLP):



Real-Time Optimization of Industrial Processes, Fig. 2 Illustrative example of using $NSRGA(S_s)$ to rank the parameters

$$\begin{aligned}
 & \min_{x_M, S_M} f(x_M, S_M) \\
 & \quad s.t. \\
 & M_1(x_{M1}, S_{M1}; \theta_{M1}) = 0 \\
 & \quad \vdots \\
 & M_n(x_{Mn}, S_{Mn}; \theta_{Mn}) = 0 \\
 & OC(x_M, S_M) \leq 0 \\
 & x_M = [x_{M1}, \dots, x_{Mn}], S_M = [S_{M1}, \dots, S_{Mn}],
 \end{aligned} \tag{9}$$

where M_i are the unit modules that can be solved by a tailored procedure in the modular approach or all together in the equation-oriented approach. Each unit model M_i has internal variables x_{Mi} and parameters θ_{Mi} . These unit models are connected by the input and output streams S_{Mi} . Additionally, there are operating constraints OC to capture the possible lower and upper bounds and other equipment constraints. The objective function $f(x_M, S_M)$ is based on an economic model that involves the raw materials, products, and operating costs.

Successive quadratic programming (SQP) has become the most popular method for solving these nonlinear constrained optimization problems. SQP converges the equality and inequality constraints simultaneously with the optimality conditions. This strategy requires relatively few function evaluations and often performs efficiently for process optimization problems. The NLP solver can be implemented in a nonintrusive

way, similar to recycle convergence modules that are already in place. As a result, the structure of the simulation environment and the unit operation blocks does not need to be modified in order to include the optimization, so that SQP can be easily incorporated within existing modular simulators and therefore be applied directly to flow sheets modeled in these commercial simulators. However, in this case derivative information must be obtained by numerical differentiation which increases the effort and slows down convergence near the optimum.

For fully equation-oriented models with the exact first and second derivatives for all constraints and objective functions, efficient NLP algorithms were developed. For instance, large equation-based models can be solved efficiently with structured barrier NLP solvers (see Biegler 2010 for a detailed overview). But for problems where function evaluations are expensive, and gradients and Hessians are difficult to obtain, it is not clear that large-scale NLP solvers should be applied. Black-box optimization models with inexact (or approximated) derivatives and few decision variables are poorly served by large-scale NLP solvers, and derivative-free optimization algorithms should be considered instead. For the standard RTO problems formulated using modular process simulator model, SQP and

reduced-space SQP methods are expected to perform well (see Biegler 2010; Alkaya et al. 2009 for detailed discussion).

After solving the RTO optimization problem, it is necessary to decide if the solution can be implemented. For this, it is necessary to verify if the dominant cause of the plant changes is noise, since in this case implementing these changes could lower the profit. Thus, an important challenge in RTO results analysis is to determine when to implement the calculated changes (Miletic and Marlin 1996).

Summary and Future Directions

RTO aims at optimizing the operation of the process taking into account economic terms directly. There are several fundamental needs for a smooth operation of an RTO solution. The central point is the mathematical modeling which can be a complex first principle model or be based on simple operating curves. If a good model is available, it is necessary to have a good characterization of the inlet streams (properties and composition), to employ data reconciliation and gross error detection and steady-state identification. Finally, the efficiency of the optimizer is a key issue. Due to the time and resources needed to implement and maintain an RTO solution, a full RTO project involves a certain high risk. Therefore, in cases where simpler and easier approaches can be applied with equivalent economic benefits, they should be used instead. For processes with large recycle streams, it is worthwhile to apply the classical RTO strategy, i.e., the one discussed in the last section. In this case the optimal solution is not trivial, once it is not simply the maximal capacity of the plant. For the cases where the optimal operating constraint is a direct consequence of the operating process capacity, the economic optimization can be easily included in the LP or QP layer implemented usually within a model predictive controller.

In the previous section, the so-called two-step approach, where the measurements are used to refine the process model, which is then used to

repeat the optimization, was described. Several RTO schemes have emerged since the development of this two-step approach in the 1970s. Recently, it has been proposed to update the model differently. Instead of adjusting the model parameters, one updates correction terms that are added to the cost and constraint functions of the optimization problem. The technique, labeled as modifier adaptation (RTO-MA), forces the modeled cost and constraints to match the plant values (Marchetti et al. 2009; Gao and Engell 2005). The main advantage of RTO-MA compared to the two-step approach lies in its ability to converge to the true plant optimum, even in the presence of structural plant-model mismatch. RTO-MA is a static optimization method which means that its application to a continuous process requires waiting for reaching the steady state before taking measurements, updating the correction terms, and repeating the numerical optimization. Hence, several iterations are generally required before convergence can be achieved. In contrast, implicit methods, such as self-optimizing control (Skogetstad 2000) and NCO tracking (François et al. 2005), propose to adjust the inputs online in a control-inspired manner. Especially simple to be implemented is the “self-optimizing” approach, where a feedback control structure is chosen so that maintaining some function of the measured variables constant automatically maintains the process near an economically optimal steady state in the presence of disturbances. The problem is posed from a plantwide perspective, since the economics are determined by overall plant behavior.

The classical steady-state RTO has some drawbacks related to its low frequency of execution. It is normally run twice or three times per day and one does not consider the cost of transiting from one operating condition to another. Some plants need to respond to market changes very quickly, like grade changes in polymerization and petroleum process. In these processes, market competition requires the capability to accommodate fast and cost-effective transitions so that companies can produce and sell on demand at favorable prices. To provide this capability, dynamic RTO is

being developed and implemented in industrial processes. The largest difference between steady-state and dynamic RTOs is that traditional RTO only provides optimal operating conditions at the steady state, while dynamic RTO provides a trajectory of changes of operating conditions. Dynamic RTO does not require steady-state conditions to be applied. The formulation and solution of the problem DRTO are very similar to the approach used to solve nonlinear predictive controllers (NMPC), with the primary difference the inclusion of economic aspects in the objective function (Engell 2007).

Cross-References

- [Control Structure Selection](#)
- [Industrial MPC of Continuous Processes](#)
- [Model-Based Performance Optimizing Control](#)
- [Model Predictive Control in Practice](#)

Recommended Reading

As a number of design decisions must be made in the construction of a RTO system, there is no single approach how to implement it. The elements of the solution were discussed here which should be viewed as a starting point for further reading. The review paper by Engell (2007) discusses and compares several approaches for RTO and DRTO giving a quite general and broad perspective of the area. For the reader interested more in the solution and formulation of the optimization problems, the book by Biegler (2010) is a very good starting point and gives a complete discussion about the solvers currently used, illustrating the application with several examples. For data reconciliation and gross error detection the book by Narasimhan and Jordache (1999) is a good starting point. Finally, an industrial discussion about RTO and alternative approaches that have been used in the industry can be found in Darby et al. (2011).

Bibliography

- Alkaya D, Vasamtharajan S, Biegler LT (2009) Successive quadratic programming: applications in the process industry. In: Floudas CA, Pardalos PM (eds) *Encyclopedia of optimization*. Springer, Berlin, pp 3853–3864
- Bebar M (2005) Regelgütebewertung in kontinuierlichen verfahrenstechnischen Anlagen. Ruhr-Universität Bochum
- Biegler L (2010) *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. MOS-SIAM series on optimization. Society for Industrial and Applied Mathematics: Mathematical Optimization Society, Philadelphia
- Botelho VR, Trierweiler LF, Trierweiler JO (2012) A new approach for practical identifiability analysis applied to dynamic phenomenological models. Paper presented at the International symposium on advanced control of chemical processes – ADCHEM 2012, Singapura
- Boyd S, El Ghaoui L, Feron E, Balakrishnan V (1994) *Linear matrix inequalities in system and control theory*. Studies in applied and numerical mathematics. Society for Industrial and Applied Mathematics, Philadelphia
- Cao S, Rhinehart RR (1995) An efficient method of on-line identification of steady state. *Rev J Process Control* 5:11
- Darby ML, Nikolaou M, Jones J, Nicholson D (2011) RTO: an overview and assessment of current practice. *Rev J Process Control* 21(6):874–884. doi:<http://doi.org/10.1016/j.jprocont.2011.03.009>
- Engell S (2007) Feedback control for optimal process operation. *Rev J Process Control* 17(3):203–219. doi:<http://doi.org/10.1016/j.jprocont.2006.10.011>
- François G, Srinivasan B, Bonvin D (2005) Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. *Rev J Process Control* 15(6):701–712. doi:<http://doi.org/10.1016/j.jprocont.2004.11.006>
- Gao W, Engell S (2005) Iterative set-point optimization of batch chromatography. *Rev Comput Chem Eng* 29(6):1401–1409. doi:<http://doi.org/10.1016/j.compchemeng.2005.02.035>
- Marchetti A, Chachuat B, Bonvin D (2009) Modifier-adaptation methodology for real-time optimization. *Rev Ind Eng Chem Res* 48(13):6022–6033. doi:10.1021/ie801352x
- Mejía RIG, Duarte MB, Trierweiler JO (2010) Avaliação do desempenho e ajuste automático de métodos de identificação de estado estacionário. In: ABEQ (ed) COBEQ 2010 Congresso Brasileiro de Engenharia Química, 10, Foz do Iguaçu
- Miletic I, Marlin T (1996) Results analysis for real-time optimization (RTO): deciding when to change the plant operation. *Rev Comput Chem Eng* 20(Supplement 2):S1077–S1082. doi:[http://doi.org/10.1016/0098-1354\(96\)00187-1](http://doi.org/10.1016/0098-1354(96)00187-1)
- Narasimhan S, Jordache C (1999) The importance of data reconciliation and gross error detection-I. In: *Data reconciliation and gross error detection*. Gulf Professional, Burlington, pp 1–31

- Sequeira SE, Graells M, Puigjaner L (2002) Real-time evolution for on-line optimization of continuous processes. *Rev Ind Eng Chem Res* 41(7):1815–1825. doi:10.1021/ie010464l
- Skogestad S (2000) Plantwide control: the search for the self-optimizing control structure. *Rev J Process Control* 10(5):487–507. doi:[http://doi.org/10.1016/S0959-1524\(00\)00023-8](http://doi.org/10.1016/S0959-1524(00)00023-8)

Redundant Robots

Stefano Chiaverini

Dipartimento di Ingegneria Elettrica e dell'Informazione "Maurizio Scarano",
Università degli Studi di Cassino e del Lazio
Meridionale, Cassino (FR), Italy

Abstract

Redundancy may occur in different ways in a robotic system. This article focuses on the resolution of kinematic redundancy, i.e., on the techniques for exploiting the redundant degrees of freedom in the solution of the inverse kinematics problem; this is indeed an issue of major relevance for motion planning and control purposes.

Keywords

Redundancy · Task-oriented kinematics · Kinematic singularity · Algorithmic singularity · Optimization · Task-space augmentation

Introduction

Redundant robots possess more resources than those strictly required to execute their task; this provides the robot with an increased capacity of facing real-world applications by allowing to handle performance issues besides the mere achievement of a given motion trajectory.

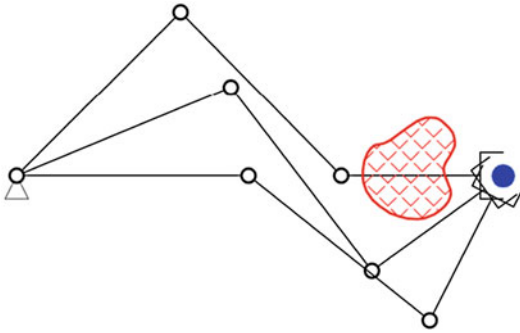
Redundancy may occur in the sensory system, in the mechanical structure, and/or in the actuation system, thus allowing, e.g., fault accommodation, multisensory perception,

dexterous motion, and load sharing. Nevertheless, unless otherwise specified, by *redundant robot* it is meant a robot that has a kinematically redundant mechanical structure, i.e., provided with more degrees of freedom than those strictly required to execute its task; this also typically leads to a redundancy in the number of actuators and sensors. Noticeably, *kinematic redundancy* is usually the key to handle the avoidance of singular configurations, the occurrence of joint limits, the engagement of obstacles in the workspace, and the minimization of joint torques or energy. In practice, if properly managed, the increased dexterity characterizing kinematically redundant robots may allow them to achieve a higher degree of autonomy.

In principle, no robot is inherently redundant; rather, there are certain tasks with respect to which it may become redundant. Nevertheless, since most papers in the classical literature on the topic have dealt with robotic manipulators (for which a general task consists in tracking an end-effector motion trajectory requiring six degrees of freedom), a robot arm with seven or more joints is often considered as a typical example of an inherently redundant manipulator. However, even robot arms with fewer degrees of freedom, like conventional six-joint industrial manipulators, may become kinematically redundant for specific tasks, such as simple end-effector positioning without constraints on the orientation.

In the case of traditional industrial applications involving non-redundant mechanical structures, the occurrence of singular configuration and/or the presence of obstacles in the workcell resulted in the need of a carefully structured (and static) working space where the motion of the manipulator could be planned in advance.

On the other hand, the presence of redundant degrees of freedom allows motions of the manipulator that do not displace the end effector (the so-called *self-motions* or *internal motions*); this implies that the same end-effector task can be executed with several different joint motions, giving the possibility of better exploiting the workspace of the manipulator and ultimately resulting in a more versatile robotic arm (see Fig. 1). Such feature is a key to allow



Redundant Robots, Fig. 1 A self-motion of the arm that keeps the end-effector position at the blue spot. It is possible to choose configurations that both take the blue spot and avoid the red obstacle

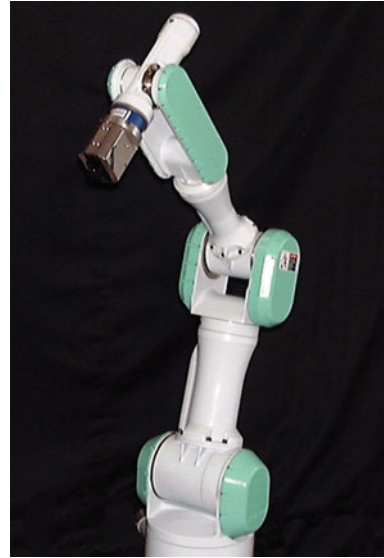
operation in unstructured and/or dynamically varying environments that characterize advanced industrial applications and service robotics scenarios.

The biological archetype of a robotic manipulator is the human arm, which, not surprisingly, also inspires the terminology used to characterize the serial-chain structure of a robot *arm*. Remarkably, a simple look at the human arm kinematics from the torso to the hand allows to recognize seven degrees of freedom (three at the *shoulder*, one at the *elbow*, and three at the *wrist*) that make a manipulator kinematically redundant.

The kinematic arrangement of the human arm has been replicated in a number of robots often termed as *human-arm-like manipulators* (see, e.g., Fig. 2). Manipulators with a larger number of joints are often called *hyperredundant robots* and include – among others – snakelike robots (Fig. 3).

The use of two or more robotic structures to execute a task (as in the case of cooperating manipulators or multifingered hands or multiarm/multilegged robots) also gives rise to kinematic redundancy. A headed multilimb structure is typical of a humanoid robot (Fig. 4). Redundant mechanisms also include vehicle-manipulator systems (Fig. 5).

Although the realization of a kinematically redundant structure raises a number of issues from the point of view of mechanical design, this article focuses on the techniques for exploiting



Redundant Robots, Fig. 2 The Mitsubishi PA-10 manipulator



Redundant Robots, Fig. 3 The SnakeRobots.com S7 snake robot prototype

R

the redundant degrees of freedom in the solution of the inverse kinematics problem. This is an issue of major relevance for motion planning and control purposes.

Task-Oriented Kinematics

The relationship between the N variables representing the configuration $\mathbf{q}$ of an articulated manipulator in the *joint space* and the M variables describing an assigned task $\mathbf{t}$ in an



Redundant Robots, Fig. 4 The Honda Asimo



Redundant Robots, Fig. 5 The KUKA youBot

appropriate *task space* constitutes a task-oriented kinematics; this can be established at the position, velocity, or acceleration level. Typically, one has $N \geq M$, so that the joints can provide at least the degrees of freedom required for the end-effector task. If $N > M$ strictly, the manipulator is *kinematically redundant*.

At the position level, the *direct kinematics* equation takes on the form

$$\mathbf{t} = \mathbf{k}_t(\mathbf{q}), \quad (1)$$

where $\mathbf{k}_t$ is a nonlinear vector function.

Besides the direct kinematics expressed at the position level, it is useful to consider the *first-order differential kinematics* (Whitney 1969)

$$\dot{\mathbf{t}} = \mathbf{J}_t(\mathbf{q}) \dot{\mathbf{q}}, \quad (2)$$

that can be obtained by differentiating Eq. (1) w.r.t. time. In (2), the mapping between the task-space and the joint-space velocities is held by the $M \times N$ *task Jacobian* matrix $\mathbf{J}_t(\mathbf{q}) = \partial \mathbf{k}_t / \partial \mathbf{q}$ (also called *analytic Jacobian*).

Remarkably, $\dot{\mathbf{t}}$ expresses the rate of change of the variables adopted to describe the task and thus does not necessarily have the meaning of an end-effector velocity. In general, by denoting the end-effector spatial velocity $\mathbf{v}_N$ as the stack of the 3D translational and angular end-effector velocities, the following relationship holds

$$\dot{\mathbf{t}} = \mathbf{T}(\mathbf{t}) \mathbf{v}_N, \quad (3)$$

where $\mathbf{T}$ is an $M \times 6$ transformation matrix.

For a given manipulator, the mapping

$$\mathbf{v}_N = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad (4)$$

relates a joint-space velocity to the corresponding end-effector velocity through the $6 \times N$ *geometric Jacobian* matrix $\mathbf{J}$.

By comparing (2), (3), and (4), the relation between the geometric Jacobian and the task Jacobian can be found as

$$\mathbf{J}_t(\mathbf{q}) = \mathbf{T}(\mathbf{t}) \mathbf{J}(\mathbf{q}). \quad (5)$$

Further differentiation of (2) w.r.t. time provides the following relationship between the acceleration variables

$$\ddot{\mathbf{t}} = \mathbf{J}_t(\mathbf{q}) \ddot{\mathbf{q}} + \dot{\mathbf{J}}_t(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}. \quad (6)$$

This equation is also known as *second-order differential kinematics*.

Singularities

A robot configuration $\mathbf{q}$ is *singular* if the task Jacobian matrix is rank-deficient at it. Considering the role of $\mathbf{J}_t$ in (2) and (6), it is easy to realize

that at a singular configuration, it is impossible to generate end-effector task velocities or accelerations in certain directions. Further insight can be gained by looking at (5), which indicates that a singularity may be due to a loss of rank of the transformation matrix T and/or of the geometric Jacobian matrix J .

Rank deficiencies of T are only related to the mathematical relationship between v_N and $t, \dot{t}$; for this reason, a configuration at which T is singular is referred to as a *representation singularity*. A representation singularity is not directly related to the true motion capabilities of the manipulator structure, which can be instead inferred by the analysis of the geometric Jacobian matrix. Rank deficiencies of J are in fact related to loss of mobility of the manipulator end effector; indeed, end-effector velocities exist in this case that are unfeasible for any velocity commanded at the joints. A configuration at which J is singular is referred to as a *kinematic singularity*.

Since redundancy resolution methods involve the inversion of the task differential kinematics (2) and (6), the handling of singularities through proper treatment of the Jacobian matrix is very important. However, due to space limitations, this topic is out of the scope of this article, and in the following we will assume that the Jacobian matrices at issue are all full rank.

Null-Space Velocities

With a full-rank task Jacobian, at each configuration, a $N - M$ dimensional null space of J_t exists made of the set of joint-space velocities that yield zero task velocity; these are thus called *null-space velocities* in short.

Remarkably, the components of $\dot{q}$ in the null space of J_t produce a change in the configuration of the manipulator without affecting its task velocity. This can be exploited to achieve additional goals – like obstacle or singularity avoidance – in addition to the realization of a desired task motion and constitutes the core of redundancy resolution approaches.

Inverse Differential Kinematics

The inverse kinematics problem can be solved by inverting the direct kinematics equation (1),

the first-order differential kinematics (2) or the second-order differential kinematics (6). With a time-varying desired task reference, it is convenient to solve the differential kinematic relationships because these represent linear equations with the task Jacobian as the coefficient matrix.

For a kinematically redundant manipulator, the general solution of (2) or (6) can be expressed by resorting to the pseudoinverse $J_t^\dagger$ of the task Jacobian matrix (Whitney 1969).

The general solution of (2) can be written as

$$\dot{q} = J_t^\dagger(q) \dot{t} + N_{J_t}(q) \dot{q}_0, \quad (7)$$

where

$$N_{J_t}(q) = I - J_t^\dagger(q) J_t(q)$$

is an orthogonal projection matrix into the null space of J_t , and $\dot{q}_0$ is an arbitrary joint-space velocity; the second part of the solution is therefore a null-space velocity. The particular solution obtained by setting $\dot{q}_0 = \mathbf{0}$ in (7) is known as the *pseudoinverse solution*.

As for the second-order kinematics (6), its solution can be expressed in the general form

$$\ddot{q} = J_t^\dagger(q)(\ddot{t} - \dot{J}_t(q, \dot{q})\dot{q}) + N_{J_t}(q) \ddot{q}_0, \quad (8)$$

where $\ddot{q}_0$ is an arbitrary joint-space acceleration.

In summary, for a kinematically redundant manipulator, the inverse kinematics problem admits an infinite number of solutions, so that a methodology to select one of them is needed.

Redundancy Resolution via Optimization

An approach to redundancy resolution is based on the optimization of suitable performance criteria.

Performance Criteria

The availability of redundant degrees of freedom can be used to improve the value of performance criteria during the motion. These criteria may depend on the robot joint configuration only or involve also velocities and/or accelerations.

The most frequently considered performance objective for trajectory tracking tasks is singularity avoidance. In fact, singularities lead to decreased mobility, and adding kinematic redundancy allows to reduce the extension of the workspace region where the manipulator is necessarily at a singular configuration (*unavoidable* singularities Baillieul et al. 1984). Possible performance criteria to drive the manipulator motion out of avoidable singularities are configuration-dependent functions that characterize the distance from singular configurations, i.e., the manipulability measure, the condition number, and the smallest singular value of $\mathbf{J}_t$.

Since kinematic inversion produces very high joint velocities in the vicinity of singular configurations, a conceptually different possibility is to minimize the norm of the joint velocity generated by the redundancy resolution scheme.

Redundancy can be also used to keep a robot away from undesired regions of the joint space or of the task space. For example, it might be desired that a manipulator avoids reaching mechanical joint limits (Liégeois 1977). Another interesting application is obstacle avoidance, which can be enforced by minimizing suitable artificial potential functions defined on the basis of the image of the obstacle region in the configuration space.

Many other performance criteria can be found in the literature.

Local Optimization

Equation (7) provides least-squares solutions to the end-effector task constraint (2), so that it minimizes $\|\dot{\mathbf{t}} - \mathbf{J}\dot{\mathbf{q}}\|$.

The simplest form of local optimization is represented by the pseudoinverse solution that provides the joint velocity with the minimum norm among those which realize the task constraint. Clearly, the joint movement generated by this locally optimal solution does not provide global velocity minimization along the entire manipulator motion; therefore, singularity avoidance is not guaranteed (Baillieul et al. 1984).

In terms of the inverse differential kinematics problem, the least-squares property may quantify the accuracy of the end-effector task realization,

while the minimum norm property may be relevant for the feasibility of the joint-space velocities.

Another possibility is to use the general solution (7), choosing $\dot{\mathbf{q}}_0$ as

$$\dot{\mathbf{q}}_0 = -k_H \nabla H(\mathbf{q}), \quad (9)$$

where k_H is a scalar stepsize and $\nabla H(\mathbf{q})$ denotes the gradient of a scalar configuration-dependent performance criterion H which is desired to minimize (Liégeois 1977).

As for the second-order solution (8), choosing $\ddot{\mathbf{q}}_0 = \mathbf{0}$ gives the minimum norm acceleration solution.

Global Optimization

Local optimization algorithms can lead to unsatisfactory performance over long-duration tasks. It is therefore natural to consider the possibility of selecting $\dot{\mathbf{q}}_0$ in (7) so as to minimize integral criteria of the form

$$\int_{t_i}^{t_f} H(\mathbf{q}) dt$$

defined over the whole duration of the task. Unfortunately, the solution of these problems (naturally formulated within the Calculus of Variations framework) may not exist and does not admit a closed form in general. One way to make the problem solvable is to use an integral criterion in quadratic form in the joint velocities or accelerations. However, this is more easily done at the second-order kinematic level (see section “[Second-Order Redundancy Resolution](#)”).

Redundancy Resolution via Task Augmentation

Another approach to redundancy resolution consists in augmenting the task vector so as to tackle additional objectives expressed as constraints.

Extended Jacobian

The extended Jacobian technique (Baillieul 1985), enforces an appropriate number of functional constraints to be fulfilled along with the original end-effector task.

Given an objective function $g(\mathbf{q})$, if $\mathbf{J}_t$ has full rank, a set of $N - M$ independent constraints can be obtained from the equation

$$\left. \frac{\partial g(\mathbf{q})}{\partial \mathbf{q}} \right|_{\mathbf{q}=\hat{\mathbf{q}}} N \mathbf{J}_t(\hat{\mathbf{q}}) = \mathbf{0}^T,$$

where $\hat{\mathbf{q}}$ is the current joint configuration such that the function $g(\mathbf{q})$ is at an extreme; these $N - M$ independent constraints can be written in vector form as

$$\mathbf{h}(\hat{\mathbf{q}}) = \mathbf{0}.$$

For a motion that tracks a trajectory $\mathbf{t}(t)$ by keeping $g(\mathbf{q})$ extremized at each time, it is

$$\begin{bmatrix} \mathbf{t}(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{k}_t(\mathbf{q}(t)) \\ \mathbf{h}(\mathbf{q}(t)) \end{bmatrix},$$

that, similarly to (1) and (2), leads to define an *extended Jacobian* matrix as

$$\mathbf{J}_{\text{ext}}(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_t(\mathbf{q}) \\ \frac{\partial \mathbf{h}(\mathbf{q})}{\partial \mathbf{q}} \end{bmatrix}.$$

Therefore, if the initial joint configuration extremizes $g(\mathbf{q})$ and provided that $\mathbf{J}_{\text{ext}}$ does not become singular, the time integral of the inverse mapping

$$\dot{\mathbf{q}} = \mathbf{J}_{\text{ext}}^{-1}(\mathbf{q}) \begin{bmatrix} \dot{\mathbf{t}} \\ 0 \end{bmatrix} \quad (10)$$

tracks the assigned end-effector trajectory $\mathbf{t}(t)$ propagating joint configurations that extremize $g(\mathbf{q})$.

The extended Jacobian method has a major advantage over the pseudoinverse solution in that it is *cyclic*, i.e., it generates repetitive joint motion from a repetitive task motion. Moreover, solution (10) can be made equivalent to (7) via suitable choice of the vector $\dot{\mathbf{q}}_0$ (Baillieul 1985).

Augmented Jacobian

The task-space augmentation approach is based on the direct definition of a constraint task to be fulfilled along with the end-effector task (Sciavicco and Siciliano 1988).

In detail, let $\mathbf{t}_c$ collect P variables that describe the additional tasks to be fulfilled besides the end-effector task $\mathbf{t}$. In the general case, it is $P \leq N - M$ although full redundancy exploitation suggests to consider exactly $P = N - M$.

The relation between the joint-space and the *constraint-task* coordinates can be considered as a direct kinematics equation

$$\mathbf{t}_c = \mathbf{k}_c(\mathbf{q}),$$

where $\mathbf{k}_c$ is a continuous nonlinear vector function. At this point, an *augmented task* can be defined by stacking the end-effector task with the constraint task as

$$\mathbf{t}_a = \begin{bmatrix} \mathbf{t} \\ \mathbf{t}_c \end{bmatrix} = \begin{bmatrix} \mathbf{k}_t(\mathbf{q}) \\ \mathbf{k}_c(\mathbf{q}) \end{bmatrix}.$$

According to this definition, finding a joint configuration $\mathbf{q}$ that brings $\mathbf{t}_a$ at some desired value means to satisfy both the end-effector and the constraint task at the same time.

A solution to this problem can be found at the differential level by inverting the mapping

$$\dot{\mathbf{t}}_a = \mathbf{J}_a(\mathbf{q}) \dot{\mathbf{q}} \quad (11)$$

where the matrix

$$\mathbf{J}_a(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_t(\mathbf{q}) \\ \mathbf{J}_c(\mathbf{q}) \end{bmatrix}$$

is termed *augmented Jacobian* and $\mathbf{J}_c(\mathbf{q}) = \partial \mathbf{k}_c / \partial \mathbf{q}$ is the $P \times N$ *constraint-task Jacobian* matrix.

A particular choice for the constraint-task vector is $\mathbf{t}_c = \mathbf{h}(\mathbf{q})$, with $\mathbf{h}$ defined as explained in section “[Extended Jacobian](#)” that allows the augmented Jacobian method to embed the extended Jacobian one.

Algorithmic Singularities

The specification of additional goals besides tracking the end-effector task raises the possibility that configurations exist at which the augmented kinematics problem is singular, while the sole end-effector task kinematics is not; these configurations are termed *algorithmic singularities* (Baillieul 1985). With reference to the velocity mappings (10) and (11), an algorithmically singular configuration is one at which the extended and the augmented Jacobians, respectively, are singular while $\mathbf{J}_t$ is full rank.

Remarkably, algorithmic singularities arise from the way in which the constraint task conflicts with the end-effector task and is not a problem of the specific inverse kinematic technique (Baillieul 1985). This is easily understandable in simple situations such as that of a desired trajectory passing through an obstacle; either the trajectory is tracked or the obstacle is avoided, so that both tasks cannot be achieved together. If the origin of the conflict between the two tasks has a clear meaning, the algorithmic singularity may be avoided by keenly specifying the constraint task case-by-case; otherwise, analytical tools must be adopted.

Task Priority

Conflicts between the end-effector task and the constraint task are handled in the framework of the task-priority strategy by suitably assigning an order of priority to the given tasks and then satisfying the lower-priority task only in the null space of the higher-priority task (Maciejewski and Klein 1985; Nakamura et al. 1987). The idea is that, when an exact solution does not exist, the reconstruction error should only affect the lower-priority task.

With reference to solution (7), the task-priority method consists in computing $\dot{\mathbf{q}}_0$ so as to suitably achieve the P -dimensional constraint-task velocity $\dot{\mathbf{i}}_c$. Remarkably, the projection of $\dot{\mathbf{q}}_0$ onto the null space of $\mathbf{J}_t$ ensures lower priority of the constraint task with respect to the end-effector task since it results in a null-space velocity.

Consistently with the defined order of priority between the two tasks, a reasonable choice is then to guarantee exact tracking of the

primary-task velocity while minimizing the constraint-task velocity reconstruction error $\dot{\mathbf{i}}_c - \mathbf{J}_c \dot{\mathbf{q}}$; this gives (Maciejewski and Klein 1985)

$$\begin{aligned} \dot{\mathbf{q}} = & \mathbf{J}_t^\dagger(\mathbf{q})\dot{\mathbf{i}} \\ & + (\mathbf{J}_c(\mathbf{q})\mathbf{N}_{\mathbf{J}_t}(\mathbf{q}))^\dagger (\dot{\mathbf{i}}_c - \mathbf{J}_c(\mathbf{q})\mathbf{J}_t^\dagger(\mathbf{q})\dot{\mathbf{i}}). \end{aligned} \quad (12)$$

It can be recognized that the problem of algorithmic singularities still remains; in fact, the matrix $\mathbf{J}_c \cdot \mathbf{N}_{\mathbf{J}_t}$ may lose rank with full-rank $\mathbf{J}_t$ and $\mathbf{J}_c$. However, differently from the task-space augmentation approach, correct primary-task solutions are expected as long as the sole primary-task Jacobian matrix is full rank. On the other hand, out of the algorithmic singularities, the task-priority strategy gives the same solution as the task-space augmentation approach; this implies that close to an algorithmic singularity, the solution becomes ill-conditioned and large joint velocities may result.

Another approach is to relax minimization of the secondary-task velocity reconstruction constraint and simply pursue tracking of the components of $\mathbf{J}_c^\dagger \dot{\mathbf{i}}_c$ that do not conflict with the primary task (Chiaverini 1997), namely:

$$\dot{\mathbf{q}} = \mathbf{J}_t^\dagger(\mathbf{q})\dot{\mathbf{i}} + \mathbf{N}_{\mathbf{J}_t}(\mathbf{q})\mathbf{J}_c^\dagger(\mathbf{q})\dot{\mathbf{i}}_c. \quad (13)$$

A nice property of solution (13) is that algorithmic singularities are decoupled from the singularities of $\mathbf{J}_c$.

Second-Order Redundancy Resolution

Redundancy resolution at the acceleration level allows the consideration of dynamic performance along the manipulator motion. Moreover, the obtained acceleration profiles (together with the corresponding positions and velocities) can be directly used as reference signals of task-space dynamic controllers.

The simplest scheme operating at the acceleration level is represented by (8) with $\ddot{\mathbf{q}} = \mathbf{0}$. Similar to the velocity-level pseudoinverse solution,

the joint motion generated by this locally optimal solution does not result in global acceleration minimization. Remarkably, provided that the appropriate boundary conditions are satisfied, this solution leads to the minimization of the integral of $\dot{\mathbf{q}}^T \dot{\mathbf{q}}$ (Kazerounian and Wang 1988).

More flexibility in the choice of performance criteria is obviously obtained by considering the full second-order solution (8). Let the manipulator dynamic model be expressed as

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\tau}_g(\mathbf{q}),$$

where $\boldsymbol{\tau}$ is the vector of actuator torques, $\mathbf{H}$ is the manipulator inertia matrix, $\mathbf{c}$ is the vector of centrifugal/Coriolis terms, and $\boldsymbol{\tau}_g$ is the gravitational torque vector. Choosing the null-space acceleration in (8) as

$$\begin{aligned} \ddot{\mathbf{q}}_0 = & -(\mathbf{H}(\mathbf{q})\mathbf{N}_{J_t}(\mathbf{q}))^\dagger \\ & \cdot (\mathbf{H}(\mathbf{q})\mathbf{J}_t^\dagger(\mathbf{q})(\ddot{\mathbf{t}} - \dot{\mathbf{J}}_t(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}) \\ & + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\tau}_g(\mathbf{q})) \end{aligned}$$

leads to the local minimization of the actuator torque norm $\boldsymbol{\tau}^T \boldsymbol{\tau}$ (Hollerbach and Suh 1987).

Another interesting inverse solution, which minimizes the integral of the manipulator kinetic energy, is the following (Kazerounian and Wang 1988)

$$\begin{aligned} \ddot{\mathbf{q}} = & \mathbf{J}_{t,H}^\dagger(\mathbf{q})(\ddot{\mathbf{t}} - \dot{\mathbf{J}}_t(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}) \\ & + (\mathbf{I} - \mathbf{J}_{t,H}^\dagger(\mathbf{q})\mathbf{J}_t(\mathbf{q}))\mathbf{H}^{-1}(\mathbf{q})\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}), \end{aligned}$$

where the *inertia-weighted task Jacobian pseudoinverse* can be computed as

$$\mathbf{J}_{t,H}^\dagger(\mathbf{q}) = \mathbf{H}^{-1}(\mathbf{q})\mathbf{J}_t^T(\mathbf{q})\left(\mathbf{J}_t(\mathbf{q})\mathbf{H}^{-1}(\mathbf{q})\mathbf{J}_t^T(\mathbf{q})\right)^{-1}.$$

Once again, the correct boundary conditions must be used.

Summary and Future Directions

To discuss kinematic redundancy, the concept of task-oriented kinematics has been first recalled with the basic methods for its inversion at the velocity and acceleration level. Next, different methods to solve kinematic redundancy at the velocity level have been arranged in two main categories, namely, those based on the optimization of suitable performance criteria and those relying on the augmentation of the task space. Finally, redundancy resolution methods at the acceleration level have been considered in order to take into account dynamics issues, e.g., torque or kinetic energy minimization.

Besides the classical linear algebra methods and optimization tools still ever under investigation, new methodological approaches to redundancy resolution recently include learning algorithms (Rolf et al. 2010) and soft computing techniques (Liu and Li 2006). Active fields of new applications are in sensorial redundancy for data fusion (Luo and Chang 2012) and in systems (like the one in Fig. 6) with a large number of degrees of freedom, namely, hyperredundant robots (Salviati et al. 2009), humanoids (Kanoun and Laumond 2010), and multirobot systems (Antonelli et al. 2010).



Redundant Robots, Fig. 6 The DLR Rollin' Justin

Cross-References

- [Cooperative Manipulators](#)
- [Optimal Control and Mechanics](#)
- [Robot Motion Control](#)

Recommended Reading

Because of space and scope limitations, in drawing an overview of such a mature and well-developed topic, there are a number of techniques and details that go neglected in any case; a slightly more extensive treatment of kinematic redundancy, including a touch on singularity robustness, cyclicity, and hyperredundant manipulators with related first-reading bibliography, can be found in Chiaverini et al. (2008, 2016). Other major issues of interest that could not be covered here are in the use of kinematic redundancy for fault tolerance, for improved grasping and for motion/force control (see, e.g., (Roberts et al. 2008; Ben-Gharbia et al. 2013; Prats et al. 2011; Khatib 1990; Ficuciello et al. 2015), respectively).

Bibliography

- Antonelli G, Arrichiello F, Chiaverini S (2010) Flocking for multi-robot systems via the null-space-based behavioral control. *Swarm Intell* 4(1):37–56
- Baillieul J (1985) Kinematic programming alternatives for redundant manipulators. In: *Proceedings of 1985 IEEE international conference on robotics and automation*, St. Louis, pp 722–728
- Baillieul J, Hollerbach J, Brockett R (1984) Programming and control of kinematically redundant manipulators. In: *Proceedings of 23th IEEE conference on decision and control*, Las Vegas, pp 768–774
- Ben-Gharbia K, Maciejewski A, Roberts R (2013) Kinematic design of redundant robotic manipulators for spatial positioning that are optimally fault-tolerant. *IEEE Trans Robot* 29:1300–1307
- Chiaverini S (1997) Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Trans Robot Autom* 13:398–410
- Chiaverini S, Oriolo G, Walker I (2008) Kinematically redundant manipulators. In: B Siciliano, O Khatib (eds) *Springer handbook of robotics*. Springer, Berlin, pp 245–268
- Chiaverini S, Oriolo G, Maciejewski A (2016) Redundant robots. In: B Siciliano, O Khatib (eds) *Springer handbook of robotics*, 2nd edn. Springer, Berlin, pp 221–242
- Ficuciello F, Villani L, Siciliano B (2015) Variable impedance control of redundant manipulators for intuitive human–robot physical interaction. *IEEE Trans Robot* 31:850–863
- Hollerbach J, Suh K (1987) Redundancy resolution of manipulators through torque optimization. *IEEE J Robot Autom* 3:308–316
- Kanoun O, Laumond JP (2010) Optimizing the stepping of a humanoid robot for a sequence of tasks. In: *Proceedings of 10th IEEE-RAS international conference on humanoid robots*, Nashville, pp 204–209
- Kazerounian K, Wang Z (1988) Global versus local optimization in redundancy resolution of robotic manipulators. *Int J Robot Res* 7(5):3–12
- Khatib O (1990) Motion/force redundancy of manipulators. In: *Proceedings of Japan-USA symposium on flexible automation*, Kyoto, pp 337–342
- Liégeois A (1977) Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Trans Syst Man Cybern* 7:868–871
- Liu Y, Li Y (2006) A new method of executing multiple auxiliary tasks by redundant nonholonomic mobile manipulators. In: *Proceedings of 2006 IEEE/RSJ international conference on intelligent robots and systems*, Beijing, pp 1–6
- Luo R, Chang CC (2012) Multisensor fusion and integration: a review on approaches and its applications in mechatronics. *IEEE Trans Ind Inf* 8:49–60
- Maciejewski A, Klein C (1985) Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *Int J Robot Res* 4(3):109–117
- Nakamura Y, Hanafusa H, Yoshikawa T (1987) Task-priority based redundancy control of robot manipulators. *Int J Robot Res* 6(2):3–15
- Prats M, Sanz P, del Pobil A (2011) The advantages of exploiting grasp redundancy in robotic manipulation. In: *Proceedings of 5th international conference on automation, robotics and applications*, Wellington, pp 334–339
- Roberts R, Hyun G, Maciejewski A (2008) Fundamental limitations on designing optimally fault-tolerant redundant manipulators. *IEEE Trans Robot* 24:1224–1237
- Rolf M, Steil J, Gienger M (2010) Goal babbling permits direct learning of inverse kinematics. *IEEE Trans Auton Ment Dev* 2:216–229
- Salviati G, Zhang H, Gonzalez-Gomez J, Prattichizzo D, Zhang J (2009) Task priority grasping and locomotion control of modular robot. In: *Proceedings of 2009 IEEE international conference on robotics and biomimetics*, Guilin, pp 1069–1074
- Sciavicco L, Siciliano B (1988) A solution algorithm to the inverse kinematic problem for redundant manipulators. *IEEE J Robot Autom* 4:403–410
- Whitney D (1969) Resolved motion rate control of manipulators and human prostheses. *IEEE Trans Man Mach Syst* 10(2):47–53

Regulation and Tracking of Nonlinear Systems

Lorenzo Marconi

C.A.SY. - DEI, University of Bologna, Bologna, Italy

Abstract

A classical problem in control theory is the design of feedback laws such that the effect of exogenous inputs on selected output variables is asymptotically rejected. This includes problems of asymptotic tracking and disturbance rejection. In this entry, the fundamentals of the theory are presented, as well as constructive procedures for the design of a controller, which embeds an “internal model” of the generator of the exogenous inputs. Current and future research directions are also discussed.

Keywords

Nonlinear output regulation · Robust control · Stabilization of nonlinear systems · Tracking

Introduction

The problem of controlling a dynamical systems in such way that a “regulated” output tracks reference signals or rejects exogenous disturbances is ubiquitous in control theory. Among various possible different approaches to the solution of this problem, in this entry we present the so-called theory of nonlinear output regulation. A distinctive feature of this theory is that reference/disturbance signals to be tracked/rejected are thought of as unknown functions of time, which belong to the set of all trajectories generated by an autonomous nonlinear system (the so-called *exosystem*). Fundamental in this setting is the concept of *internal model*, developed in the early 1970s for linear systems by Francis and Wonham (1976) and subsequently extended, beginning with the work (Isidori and Byrnes

1990), to the case nonlinear systems. Since these early contributions, nonlinear output regulation has been an active research domain, in which constant improvements have brought the theory to a stage of full maturity. In this entry we introduce the fundamental principles of the nonlinear output regulation theory and the associated design tools. The entry ends with an overview of actual research trends and future research directions.

The Generalized Tracking Problem for Nonlinear Systems

We consider the class of time-invariant smooth nonlinear systems described in the form

$$\begin{aligned}\dot{x} &= f(w, x, u) \\ e &= h(w, x) \\ y &= k(w, x)\end{aligned}\tag{1}$$

in which $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control input, $y \in \mathbb{R}^q$ is the measured output, and $e \in \mathbb{R}^p$ is the regulation *error*. The input $w \in \mathbb{R}^s$ models exogenous signals that might represent references to be tracked, exogenous disturbances to be rejected, or also parametric uncertainties. In this framework the problem is to design a controller of the form

$$\begin{aligned}\dot{\xi} &= \varphi(\xi, y) & \xi &\in \mathbb{R}^v \\ u &= \gamma(\xi, y)\end{aligned}\tag{2}$$

such that the associated closed-loop system (1)–(2) has bounded trajectories $(x(t), \xi(t))$ and the resulting error $e(t) = h(w(t), x(t))$ is asymptotically vanishing, i.e., $\lim_{t \rightarrow \infty} e(t) = 0$. The previous framework encompasses several standard control problems, such as the problem in which a system of the form $\dot{x} = f(x, u)$, with measured output $y = k(x)$ and regulated output $y_r = h_r(x)$, must be controlled in such a way that $y_r(t)$ asymptotically tracks a reference signal $y^*(t)$. This is the case, in fact, if we set $w(t) = y^*(t)$, define $e = h(w, x) = w - h_r(x)$, and drop the dependence from w in the functions $f(\cdot)$ and $k(\cdot)$ in (1). Similarly, the previous framework lends itself to capture a scenario of disturbance suppression, in which, in a system of

the form $\dot{x} = f(d, x, u)$, with measured output $y = k(x)$ and regulated output $y_r = h(x)$, the effect of a disturbance $d(t)$ on the regulated output $y_r(t)$ must be asymptotically rejected. This is the case if we set $w(t) = d(t)$ and drop the dependence on w in $h(\cdot)$ and $k(\cdot)$ in (1). Similarly, by letting $h(x) = x$ and interpreting the variable w as parametric uncertainty, the previous setting captures the problem of robust output feedback stabilization, at the origin, of an uncertain system of the form $\dot{x} = f(w, x, u)$ with measured output $y = k(w, x)$. Of course, the general case of tracking reference signals in presence of exogenous disturbances can be cast in a similar manner.

The ability of solving the problem in question strongly depends on the amount of knowledge one assumes about the exogenous variable w in the design of the controller (2). Among the different options available, in this entry we present the so-called theory of output regulation, in which the exogenous variable is assumed to be an *unknown member* of a *known family* of functions of time. Specifically, it is assumed that $w(t)$ is an unspecified member of the set of all trajectories generated by an autonomous nonlinear system of the form

$$\dot{w} = s(w) \quad (3)$$

as its initial condition $w(0)$ ranges on a prescribed set $W \subset \mathbb{R}^s$. In this framework, system (3), usually referred to as the “exosystem,” is assumed to be known and its knowledge potentially exploitable in the design of (2). The specific “member” $w(t)$ of the family, however, is unspecified as the initial condition $w(0)$ is not known. The fact of regarding $w(t)$ as unknown member of a known family seems to be the right trade-off between the favorable but unrealistic situation in which $w(t)$ is assumed to be perfectly known and the opposite realistic but conservative situation in which $w(t)$ is regarded as a totally unknown signal. An elementary, and yet meaningful, example is given by the case in which $w(t)$ belongs to the family of periodic functions of time with an unspecified frequency, phase, and amplitude. In this case the exosystem (3) is a *nonlinear* system

of the form ($w \in \mathbb{R}^3$)

$$\dot{w}_1 = w_2 \quad \dot{w}_2 = -w_3^2 w_1 \quad \dot{w}_3 = 0.$$

Solutions of the previous system, in fact, are periodic functions, with frequency $w_3(t) \equiv w_3(0)$ and amplitude and phase depending on the specific initial condition $(w_1(0), w_2(0))$. Other situations, such as exosystems modeling nonlinear oscillators, can be dealt with in a similar fashion.

In the previous context, the problem of output regulation can be formally cast as follows. Let $X \subset \mathbb{R}^n$ be a set of initial conditions for (1). Then, the problem consists in finding a controller of the form (2), with initial conditions in a set $\mathcal{E} \subset \mathbb{R}^v$, such that the trajectories of the closed-loop system (1)–(2) augmented with (3), originating from an initial condition $(w(0), x(0), \xi(0)) \in W \times X \times \mathcal{E}$, are bounded and $\lim_{t \rightarrow \infty} e(t) = 0$ *uniformly* in the initial conditions (The property of “uniformity” is relevant in the context of output regulation. It reflects the requirement that the time needed for the error $e(t)$ to reach an ϵ -neighborhood of the origin only depends on the set $W \times X \times \mathcal{E}$, where the initial conditions are supposed to range, and on ϵ but not on the particular value of the initial condition within $W \times X \times \mathcal{E}$.) Depending on the assumptions on the set X where the initial conditions of the plant are assumed to range, the problem is further classified in *semiglobal output regulation*, if the set X is a known *compact* but otherwise arbitrary set of $\mathbb{R}^n$, or *global output regulation* if $X = \mathbb{R}^n$.

Output Regulation Principles

Steady State for Nonlinear Systems and Internal Model Principle

Since the objective is to design the controller such that the effect of the exogenous variable is *asymptotically* rejected by the regulation error, it is apparent that any approach to the solution of the problem of output regulation must be

necessarily grounded on a precise characterization of the notion of “steady state” for a nonlinear system. As it is the case for the familiar version of this concept in linear systems theory, a notion of “steady state,” for the system consisting of (1)–(3), should be able to capture the “limiting behavior” – if any – that such system asymptotically approaches when the “transient behavior,” due to the effect of specific initial conditions of plant and controller, fades out and a “persistent behavior,” induced only by the specific exogenous input, emerges. In this respect, the mathematical tool that has been shown to be at the core of a rigorous notion of steady state for nonlinear systems is the one of ω -limit set of a set. We refer the reader to Hale et al. (2002) for a definition of this notion and to Byrnes and Isidori (2003) for a description of its use in the characterization of the steady-state behavior of a nonlinear system. In this entry, we simply observe that if the trajectories of the system (3)–(1)–(2) that originate from the set of initial conditions $W \times X \times \mathcal{E}$ are bounded (which, in turn, is one of the requirements of the problem in question), then there exists a compact set $\mathcal{A} \subset \mathbb{R}^s \times \mathbb{R}^n \times \mathbb{R}^v$, which is precisely the ω -limit set of the set $W \times X \times \mathcal{E}$ under the dynamics of (3)–(1)–(2), that is *invariant* for the closed-loop system and that *uniformly* attracts its trajectories. The set $\mathcal{A}$ is usually referred to as *steady-state locus*, while the restriction of the closed-loop dynamics to the set $\mathcal{A}$ are the *steady-state dynamics* of the closed-loop system. The latter characterize the “limiting behavior” of the system towards which all the closed-loop trajectories converge to. Unlike the case of linear systems, though, in a nonlinear context we cannot expect, in general, that the steady-state behavior is only governed by the exogenous w , namely, that the asymptotic behavior of the closed-loop system is totally independent of the initial conditions of the plant and of the regulator. Assuming that the set W is compact and invariant for (3), it can be proven (see Isidori and Byrnes 2008) that the set $\mathcal{A}$ is the graph of a *set-valued* map defined on W , namely, that there exists a map $\sigma : W \rightarrow \mathbb{R}^n \times \mathbb{R}^v$, which is set-valued in general, such that

$$\mathcal{A} = \{(w, x, \xi) \in W \times \mathbb{R}^n \times \mathbb{R}^v : (x, \xi) \in \sigma(w)\}.$$

Clearly the steady-state locus and the associated steady-state dynamics of the closed-loop system depend on the design of the controller (2). The role of the latter is not only to enforce the existence of a steady state, which is equivalent to enforce bounded closed-loop trajectories, but also to guarantee that the error converges asymptotically to zero uniformly in the initial conditions. In this respect, by bearing in mind the asymptotic properties of the set $\mathcal{A}$, it can be seen that a sufficient condition under which a regulator of the form (2) solves the problem of output regulation is that the steady-state locus is “shaped” in such a way that the regulation error is zero on it, namely,

$$\mathcal{A} \subset \{(w, x, \xi) : h(w, x) = 0\}. \quad (4)$$

In fact, it can be proved that condition (4) is not only sufficient but also necessary (see Byrnes and Isidori 2003) as a consequence of the requirement that the error converges to zero uniformly in the initial conditions. That is, any regulator that solves the problem in question necessarily enforces a steady state such that the steady-state locus fulfills (4).

In view of the previous considerations, a crucial property required to any regulator is to induce a steady-state locus $\mathcal{A}$ fulfilling (4). This key feature can be further elaborated by highlighting two necessary conditions, involving separately the plant and the regulator, leading to the notions of *regulator equations* and *internal model property*. To this purpose, consider the simplified, yet relevant, case in which the map $\sigma(\cdot)$ is single-valued and smooth, and let $\pi(w)$ and $\tau(w)$ be the two components of $\sigma(w)$ associated to x and ξ , respectively. By letting

$$c(w) = \gamma(\tau(w), k(w, \pi(w))) \quad (5)$$

it is immediately realized that the fact that $\mathcal{A}$ is invariant for (1)–(3) implies that the functions $\pi(\cdot)$ and $\tau(\cdot)$ necessarily fulfill

$$\frac{\partial \pi(w)}{\partial w} s(w) = f(w, \pi(w), c(w)) \quad (6)$$

and

$$\frac{\partial \tau(w)}{\partial w} s(w) = \varphi(\tau(w), k(w, \pi(w))) \quad (7)$$

for all $w \in W$. Furthermore, the fact that e must be zero on $\mathcal{A}$ (see (4)) implies that necessarily

$$h(w, \pi(w)) = 0 \quad (8)$$

for all $w \in W$. Equations (6) and (8), interpreted as equations in the unknown $\pi(w)$ and $c(w)$, involve only the regulated plant (1) and are known as *regulator equations* (see Isidori and Byrnes 1990; Isidori 1995). The functions $c(w(t))$ and $\pi(w(t))$, with $w(t)$ solution of (3), represent, respectively, the desired steady-state control input and state towards which the actual control input u and state x of (1) should converge in order to have the regulation goal fulfilled. On the other hand, Eqs. (5) and (7), interpreted as equations in the unknown $\tau(w)$, point out the so-called internal model property required to any regulator solving the output regulation problem, that is, the ability of the regulator to reproduce the ideal steady-state input $c(w(t))$, for all possible $w(t)$ solution of (3), once it is driven by the measured output of the plant in the ideal steady state (namely, by the function $k(w(t), \pi(w(t)))$). In fact, this property can be achieved by incorporating in the controller an appropriate “internal model” of the exogenous dynamics (3).

Regulator Design

As emphasized in the previous discussion, the design of the regulator involves the fulfillment of two crucial properties. The first is the internal model property, namely, the ability of generating, by means of the regulator outputs, all possible “feedforward inputs” which force an identically zero regulation error and, in turn, to guarantee the existence of an invariant steady-state set on which the error is identically zero. The second property asks that such a steady-state set is asymptotically stable for the

closed-loop system with a domain of attraction including the set of initial conditions. A systematic design procedure of regulators simultaneously fulfilling the previous two properties can be found under sufficient conditions that essentially restrict the class of regulated plants (1). In particular, in the following, we consider the class of single input-single output systems that are affine in the input u , with a measurable error variable (i.e., $e = y$) and that after an appropriate change of coordinates can be written in the form

$$\begin{aligned} \dot{z} &= f_z(w, z, e) & z &\in \mathbb{R}^{n-1} \\ \dot{e} &= a(w, z, e) + b(w, z, e)u & e, u &\in \mathbb{R} \end{aligned} \quad (9)$$

with $f_z(\cdot, \cdot, \cdot)$, $a(\cdot, \cdot, \cdot)$, and $b(\cdot, \cdot, \cdot)$ smooth functions with $b(w, z, e) \neq 0$ for all (w, z, e) . Systems of this kind possess a well-defined unitary relative degree (The restriction to systems with unitary relative degree is just made for sake of simplicity. Higher relative degree can be equally dealt with, Isidori (1995).) between the input u and the output e , and the Eqs. (9) are said to be in *normal form* (see Isidori 1995, 2013). In these coordinates an easy calculation shows that the solution of the regulator Eqs. (6) and (8) takes the form $\pi(w) = (\pi_z(w), 0)$, where $\pi_z(\cdot) : W \rightarrow \mathbb{R}^{n-1}$ is a solution of

$$\frac{\partial \pi_z(w)}{\partial w} s(w) = f_z(w, \pi_z(w), 0),$$

and

$$c(w) = -\frac{a(w, \pi_z(w), 0)}{b(w, \pi_z(w), 0)}.$$

In addition, we further restrict the class of systems by asking for a minimum-phase property (Isidori 1995, 2013). In the present context, the property in question amounts to asking that the set $\mathcal{B} = \{(w, z) \in W \times \mathbb{R}^{n-1} : z = \pi_z(w)\}$ is *asymptotically stable* for the system

$$\begin{aligned} \dot{w} &= s(w) \\ \dot{z} &= f_z(w, z, 0) \end{aligned}$$

with a domain of attraction containing $W \times Z$, with Z the set where the initial condition of z is expected to range.

Existence of the relative degree and the property of minimum-phase are all what is needed to design a regulator. The regulator takes the form

$$\begin{aligned}\dot{\xi} &= F\xi + G(\gamma(\xi) + \kappa(e)) \quad \xi \in \mathbb{R}^v \\ u &= \gamma(\xi) + \kappa(e)\end{aligned}\quad (10)$$

in which (F, G) is a controllable pair and $\gamma(\cdot)$ and $\kappa(\cdot)$ are real-valued functions to be properly designed. In particular, it can be shown (see Marconi et al. 2007) that if v , the dimension of the regulator is taken sufficiently large relative to the dimension of w (specifically, $v \geq 2s + 2$) and if F is any matrix whose eigenvalues have negative real part, there exist a continuously differentiable function $\tau(\cdot)$ and a continuous function $\gamma(\cdot)$ such that

$$\begin{aligned}\frac{\partial \tau(w)}{\partial w} s(w) &= F\tau(w) + Gc(w) \\ c(w) &= \gamma(\tau(w))\end{aligned}\quad (11)$$

for all $w \in W$. This being the case, it is seen that the regulator (10) fulfills conditions (5)–(7) and therefore has the internal model property, regardless of how $\kappa(\cdot)$ is chosen, provided that $\kappa(0) = 0$. In particular, in the closed-loop system (3), (9), and (10), the invariant set $\mathcal{A} = \{(w, z, e, \xi) \in W \times \mathbb{R}^{n-1} \times \mathbb{R} \times \mathbb{R}^v : z = \pi_z(w), e = 0, \xi = \tau(w)\}$ fulfills (4) regardless of how $\kappa(\cdot)$ is chosen. The function $\kappa(\cdot)$ is a degree of freedom that can be chosen to make the steady-state set $\mathcal{A}$ asymptotically stable. In this respect the minimum-phase assumption and the fact that F is a Hurwitz matrix play a role. In fact, the closed-loop system (3), (9), and (10), interpreted as a system with state (w, z, e) , input $\kappa(\cdot)$, and output e , have relative degree one and it is minimum-phase. This fact makes it possible to use standard high-gain arguments to show that there exists a function $\kappa(\cdot)$ such that the set $\mathcal{A}$ is asymptotically stable for the closed-loop systems with a domain of attraction containing any (arbitrarily large) compact set of initial conditions (see Teel and Praly 1995; Marconi et al. 2007).

The delicate part in the procedure illustrated above is the design of the function $\gamma(\cdot)$ that is

required to fulfill (11) for a suitable $\tau(\cdot)$. Exact, although hardly implementable in practice, expressions for the function $\gamma(\cdot)$ can be found in Marconi and Praly (2008). More constructive design procedures can be found at the price of restricting the class of systems and exosystems that can be dealt with. Such procedures require that the autonomous dynamical system with “output” u^*

$$\begin{aligned}\dot{w} &= s(w) \\ u^* &= c(w),\end{aligned}$$

namely, the system characterizing all possible ideal steady-state inputs, is “immersed” into a system exhibiting certain structural properties (Loosely speaking, the autonomous system with output Σ is immersed into the autonomous system with output $\tilde{\Sigma}$ if the set of all possible functions of time generated as outputs of Σ is a subset of the set of all possible functions generated as outputs of $\tilde{\Sigma}$). In this respect a number of alternative solutions have been proposed in literature that differ for the kind of underlying immersion assumption and consequent regulator design procedure. Immersion into a linear known observable system (see Huang and Lin 1994; Khalil 1994; Byrnes et al. 1997; Serrani et al. 2000), immersion into a linear unknown (but linearly parameterized) system (Serrani et al. 2001), immersion into a linear system having a nonlinear output map (Chen and Huang 2004), immersion into a nonlinear system linearizable by output injection (Delli Priscoli 2004), immersion into a system in canonical observability form Byrnes and Isidori (2004), and immersion into a system in a nonlinear adaptive observability form Delli Priscoli et al. (2006a, b) are a few examples of approaches proposed in literature.

Summary and Future Directions

The theory of output regulation for nonlinear systems is an active area of investigation. Research efforts are, in particular, addressed to the problems of weakening the minimum-

phase assumption and of identifying *robust* design procedures, to asymptotically stabilize the steady-state locus, not necessarily based on high-gain principles. Recently, the problem of output regulation for multivariable systems has been also addressed (Isidori and Marconi 2012). In this case a paradigm shift in the design of the regulator and of the stabilizer is expected to deal with the problem in its full generality. Finally, it is worth mentioning that the theory of output regulation and internal model-based design methods are being used for the problem of reaching a consensus between the outputs of a network of nonlinear systems exchanging relative information over a communication graph. In this case it has been proved the necessity of internal model-based regulators (Wieland 2010) and the research activity is now conveyed to identify constructive design strategies for classes of nonlinear systems and network topologies.

Cross-References

- [Differential Geometric Methods in Nonlinear Control](#)
- [Lyapunov's Stability Theory](#)
- [Nonlinear Zero Dynamics](#)
- [Tracking and Regulation in Linear Systems](#)

Bibliography

- Byrnes CI, Isidori A (2003) Limit sets, zero dynamics and internal models in the problem of nonlinear output regulation. *IEEE Trans Autom Control* 48:1712–1723
- Byrnes CI, Isidori A (2004) Nonlinear internal models for output regulation. *IEEE Trans Autom Control* 49:2244–2247
- Byrnes CI, Delli Priscoli F, Isidori A, Kang W (1997) Structurally stable output regulation of nonlinear systems. *Automatica* 33:369–385
- Chen Z, Huang J (2004) Global robust servomechanism problem of lower triangular systems in the general case. *Syst Control Lett* 52:209–220
- Delli Priscoli F (2004) Output regulation with nonlinear internal models. *Syst Control Lett* 53:177–185
- Delli Priscoli F, Marconi L, Isidori A (2006a) A new approach to adaptive nonlinear regulation. *SIAM J Control Optim* 45:829–855
- Delli Priscoli F, Marconi L, Isidori A (2006b) Nonlinear observers as nonlinear internal models. *Syst Control Lett* 55:640–649
- Francis BA, Wonham WM (1976) The internal model principle of control theory. *Automatica* 12:457–465
- Hale JK, Magalhães LT, Oliva WM (2002) Dynamics in infinite dimensions. Springer, New York
- Huang J (2004) Nonlinear output regulation: theory and applications. SIAM, Philadelphia
- Huang J, Lin CF (1994) On a robust nonlinear multivariable servomechanism problem. *IEEE Trans Autom Control* 39:1510–1513
- Isidori A (1995) Nonlinear control systems, 3rd edn. Springer, Berlin/New York
- Isidori A (2013) Nonlinear zero dynamics. In: Encyclopedia of systems and control. Springer
- Isidori A, Byrnes CI (1990) Output regulation of nonlinear systems. *IEEE Trans Autom Control* 25:131–140
- Isidori A, Byrnes CI (2008) Steady-state behaviors in nonlinear systems with an application to robust disturbance rejection. *Ann Rev Control* 32:1–16
- Isidori A, Marconi L (2008) System regulation and design, geometric and algebraic methods'. In: Encyclopedia of complexity and system science. Section control and dynamical systems. Springer, Heidelberg
- Isidori A, Marconi L (2012) Shifting the internal model from control input to controlled output in nonlinear output regulation. In: Proceedings of the 51th IEEE conference on decision and control, Maui
- Isidori A, Marconi L, Serrani A (2003) Robust autonomous guidance: an internal model-based approach. Limited series advances in industrial control. Springer, London
- Khalil H (1994) Robust servomechanism output feedback controllers for feedback linearizable systems. *Automatica* 30:587–1599
- Marconi L, Praly L (2008) Uniform practical output regulation. *IEEE Trans Autom Control* 53(5):1184–1202
- Marconi L, Praly L, Isidori A (2007) Output stabilization via nonlinear luenberger observers. *SIAM J Control Optim* 45(6):2277–2298
- Pavlov A, van de Wouw N, Nijmeijer H (2006) Uniform output regulation of nonlinear systems: a convergent dynamics approach. Birkhauser, Boston
- Serrani A, Isidori A, Marconi L (2000) Semiglobal output regulation for minimum-phase systems. *Int J Robust Nonlinear Control* 10:379–396
- Serrani A, Isidori A, Marconi L (2001) Semiglobal nonlinear output regulation with adaptive internal model. *IEEE Trans Autom Control* 46:1178–1194
- Teel AR, Praly L (1995) Tools for semiglobal stabilization by partial state and output feedback. *SIAM J Control Optim* 33:1443–1485
- Wieland P (2010) From static to dynamic couplings in consensus and synchronization among identical and non-identical systems. PhD thesis, Universität Stuttgart

Rehabilitation Robots

N. Vitiello^{1,2,3}, E. Trigili^{1,3}, and S. Crea^{1,2,3}

¹The Biorobotics Institute, Pisa, Italy

²IRCCS Fondazione Don Carlo Gnocchi,
Milan, Italy

³Department of Excellence in Robotics and AI,
Pisa, Italy

Abstract

In this entry, common robotic architectures employed in rehabilitation robotics are presented, focusing on their specific strengths and drawbacks, with examples of commercially available devices for each architecture. The main control strategies currently implemented to promote brain plasticity and functional recovery via robot-aided rehabilitation are illustrated. Finally, current trends and future directions in the field are discussed.

Keywords

Stroke · Robotic architectures · Impedance control

Introduction

Among cardiovascular diseases, stroke is one of the main causes of long-term disability in modern countries. Although statistics suggest that the global incidence of stroke is decreasing, the absolute number of strokes is expected to increase dramatically in few years, leading to 1.5 million European people suffering a stroke each year by 2025 (Béjot et al. 2016). This trend can be associated to the improvement of the quality of life and the consequent increase of life expectation. Such phenomena are gradually shaping our modern society, which is called upon to tackle the issues of population aging and its sustainability. A growing number of people will be at higher risk of developing age-associated chronic diseases, seriously impacting the healthcare system and related services. As

an example, common consequences of a stroke arise as cognitive or physical impairments. In about 85% of the cases, stroke results in muscle weakness affecting one side of the body (hemiparesis), limiting independence of affected people in performing common activities of daily living. Few months after stroke occurs, some conditions might emerge due to immobility, as different expressions of the upper motor neuron syndrome (UMNS). These include modifications of physiological stretch reflex, whose hyper-excitability results in involuntary, spastic muscle contractions. Spasticity of the shoulder girdle muscles occurs in most cases in the hemiplegic arm, forcing to unnatural and painful postures. Starting during acute phases, high-intensity and long-duration physical therapy becomes essential to mitigate the onset of spasticity, relearn motor skills, and maximize functional recovery. Traditional therapy performed in rehabilitation centers consists of one-to-one sessions between patient and therapist, who needs to perform manual mobilization of the affected limb typically for 1 hour, requiring significant physical effort and time.

For this reason, robot-aided rehabilitation has soon emerged as a powerful tool to increase the efficacy of the rehabilitation process, coming in aid of physical therapists for delivering repetitive, measurable, and effective treatments. The general trend of modern clinical rehabilitation favors the employment of a robotic device, provided with sensors able to register a large amount of quantitative data (e.g., range of motion, forces, speed, residual voluntary action), to allow physicians a more objective and repeatable evaluation of the patient's condition. Rehabilitation robotics can lead to disruptive advantages in tackling patient's impairments, reducing the therapists' workload, allowing them to follow more than one patient simultaneously, with the possibility to improve and adapt the therapy based on experimental observations. Although the benefits of robot-aided rehabilitation in terms of recovery of motor functions over traditional physical therapy are debated, the former is gradually introduced by physicians into the standards of care because of the aforementioned advantages.

Robot Architectures

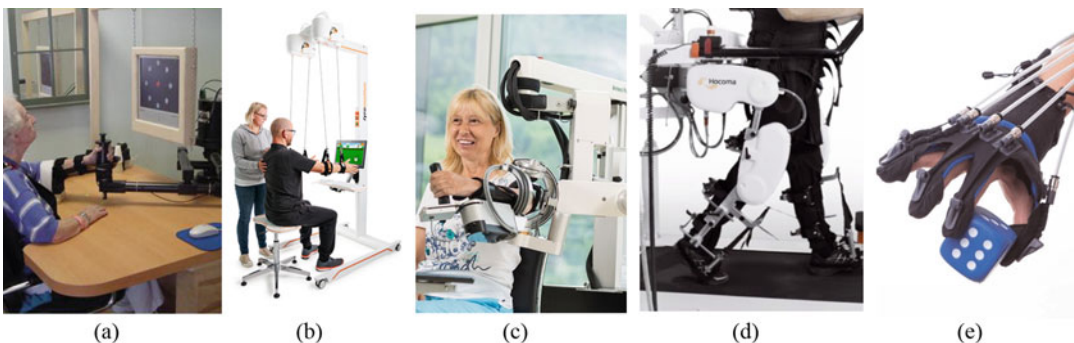
Rehabilitation robots can be classified in four main architectures, based on their physical human-robot interface. On account of their specific strengths and weaknesses, there are no preferred solutions in the state of the art; indeed the choice is strongly dependent on the targeted end users and the need for balancing and prioritizing between power, costs, ease of use, and actual deployment in clinical settings.

Endpoint manipulators were the first to be introduced in the state of the art, by virtue of their low mechanical complexity and ease of setup. They interface with the human limb only at a distal portion, usually by means of a handle (in the case of devices for upper-limb physical rehabilitation), resulting in an intrinsically safe operation. Nevertheless, they lack the possibility to address human joints in a selective manner; thus they are less effective in preventing abnormal compensatory movements occurring often in poststroke patients. The MIT-MANUS (Fig. 1a) is a planar endpoint manipulator, which has been involved in the highest number of clinical trials and the first to underline the importance of engaging and challenging patients with game-like visual feedbacks to guide the user's movements (Krebs et al. 2004). It is currently commercialized by Bionik Laboratories (Toronto, Canada) under the

name of InMotion™, in different configurations for arm, hand, and wrist rehabilitation.

Cable suspensions respond to the needs of reconfigurable, light-weighted robots, where freedom of movement is maximized whereas inertial effects due to the device presence are minimized. Forces or torques are transmitted by means of remotely actuated tendon cables, attached to the user at a single point or at different points around the addressed body segment. The Freebal is endowed with spring mechanisms for weight compensation of the elbow and wrist, which provide constant vertical forces at the sling attachments by means of cable pulleys placed on an overhead fixed beam guide (Stienen et al. 2007). With a similar structure, DIEGO® (Tyromotion, Graz, Austria – Fig. 1b) is a commercially available cable suspension, designed to treat one or both arms via a computer-assisted virtual reality system. Slings attached to the user's arm and/or hand provide active training options for the respective proximal or distal joints or the entire upper extremity, via gravity compensation, enabling treatment of conditions such as hemiparesis, spasticity, or restricted joint mobility.

Powered “rigid” exoskeletons are considered by many researchers as a suitable solution to provide movement assistance in a more physiological



Rehabilitation Robots, Fig. 1 Common architectures for robot-aided rehabilitation: (a) endpoint manipulator MIT-Manus. (Adapted from Krebs et al. 2004); (b) cable suspension DIEGO® (Tyromotion, Austria); (c) Armeo

Power for upper-limb rehabilitation. (Picture: Hocoma, Switzerland); (d) Lokomat gait trainer for lower-limb rehabilitation. (Picture: Hocoma, Switzerland); (e) Gloreha Sinfonia exoglove (Idrogenet Srl, Italy)

manner, by acting selectively on each target human joint with the ultimate goal to retain and foster the physiological muscle synergies. In the last decade, they have been witnessed a capillary diffusion across several branches of modern society, as assistive tools in work environments, rehabilitation devices in clinical settings, or power augmentation systems in the military field. The Armeo Power (Nef et al. 2009 – Fig. 1c), and the Lokomat (Colombo et al. 2001 – Fig. 1d) are two powered exoskeletons commercialized by Hocoma (Volketswil, Switzerland), designed for the rehabilitation of the upper and lower limb respectively. They allow highly intensive and repetitive training of limb motor functions in a tridimensional environment, engaging patients with augmented performance feedback exercises. On the one hand, quantitative parameters are extracted to provide patients with visual feedback about their capability of fulfilling the current task, with the aim of increasing their motivation and improving their performance. On the other hand, such parameters are used to evaluate how patients perform during their therapy sessions and adaptively modify the support they receive from the robot. With respect to other robotic architectures, the improved performance in terms of power and precision and the possibility to address specifically each single joint retaining functional muscle synergies come at the expense of a higher mechanical complexity. Indeed, when motion is transmitted, the kinematic chain of the exoskeleton must be able to follow the physiological displacement of the human articulations, which exhibit a certain degree of laxity. Additional passive degrees of freedom (DOF) are usually embedded to realize so-called *self-aligning* kinematic chains, able to cover the residual movements of human joints. If proper alignment between the exoskeleton joint and the addressed articulation is guaranteed, force/torque can be transferred at the collocated human joint without generating residual tangential forces causing pain or discomfort to the user.

Finally, *powered “soft” exoskeletons*, including *exosuits* for the whole body or *exogloves* for the hand, are currently being developed to overcome the main limitations of their

rigid counterpart. Portability, wearability, and lightness are maximized by virtue of their mechanical structure, embedding actuators within clothing-like fabric frames intrinsically adaptable to the addressed body structures. Bioinspired cable transmission systems are used in a tendon-like fashion, with origin and insertion points across the human articulations, which are exploited directly to produce motion similarly to the human muscles and tendons. Soft architectures exhibit very low encumbrance and inertia and inherently solve the problem of joint misalignment. Nevertheless, with respect to rigid exoskeletons, they suffer from narrower bandwidth and lower transmission efficiency, resulting suitable for the rehabilitation of patients with a moderate level of impairment. The Gloreha Sinfonia (*Idrogenet Srl*, Lumezzane, Italy – Fig. 1e) is an example of commercial exoglove designed for hand rehabilitation, able to assist totally or partially the movement of the hand, according to the patient’s residual capabilities (Borboni et al. 2016).

Common Features of Rehabilitation Robots

Common to all rehabilitation robots is their capability to adapt the supplied therapy (i.e., movement assistance) to the end users, thus coping with their different levels of movement capabilities. Adaptation is realized by means of a mix of both hardware and software device capabilities.

From the hardware perspective, a desirable feature of rehabilitation robots is their capability to exhibit – when needed – a minimum-to-null output impedance, thus allowing the implementation of a variety of physical training strategies, with a different level of involvement of the patient in the execution of the movement. Within this framework, the use of compliant actuators, e.g., series elastic actuators (Pratt and Williamson 1995), is becoming a favorable choice to invent and develop innovative rehabilitation robots (Otten et al. 2015; Ercolini et al. 2018). Indeed, hardware compliance is introduced in the robot design to upper limit the inherent robot stiffness

so that the user will always experience a safe and comfortable interaction even in the case of sudden and impulsive movements, such as in the cases of spastic reactions or abnormal muscle activations.

From the viewpoint of the software, the more relevant common feature is the capability of the control system to adapt the assistive action to the different user's residual movement capabilities, along with the progression of the recovery process. All control strategies employed in robot-aided rehabilitation fall within two main categories, namely, "*device-in-charge support*" and "*patient-in-charge support*" strategies (Haarman et al. 2016). The device-in-charge scheme operates by dragging the user's limb to follow a preset trajectory in the working space. The robot delivers completely passive mobilization, by compensating for any deviation from the desired path regardless of the contribution of the user. Usually implemented in the form a position control, where each of the robot joint follows a pre-defined trajectory, this modality of operation is preferred in the early stages of rehabilitation, in which patients exhibit nearly null movement capabilities.

On the contrary, the patient-in-charge scheme includes all situations in which an active interaction between the user and the robot is required. It replicates physician's actions on the user's limb, who initiate the movement and provide assistive forces *as needed*, leaving the patient the control of the task. Patient-in-charge schemes are implemented via force or torque control, enabling the robot either to act as "transparent" to the user without hindering spontaneous movements (zero-torque control) or to provide specific joint torques in aid of the patient. This scheme is particularly suitable at later stages of the rehabilitation process, when patients have regained some motor functions.

Switching between these features can be attained by means of *impedance* or *admittance* control strategies. Mechanical impedance is defined as the relationship between the net force applied to the robot $F(s)$ and the resulting

displacement $X(s)$. For a linear, time-invariant system, the mechanical impedance in the frequency domain has the following formulation (Masia et al. 2018):

$$Z(s) = \frac{F(s)}{X(s)} = Ms^2 + Bs + K, \quad (1)$$

where M , B , and K represent the mass, damping, and stiffness parameters of the coupled system (i.e., robot and human). The admittance is defined as the reciprocal of the impedance. An impedance control strategy is based on the tuning of a desired impedance Z_d , by choosing the desired virtual parameters M_d , B_d , K_d of the coupled system.

A closed-loop impedance control comprises an inner force controller and an outer position controller. For a given desired trajectory $X_d(s)$, the desired force for the actuator is given by:

$$F_d(s) = Z_d(s) [X_d(s) - X(s)]. \quad (2)$$

Thus, the robot can be transparent not to hinder user's movements by tuning its mechanical output impedance to a quasi-zero value. Desired forces for the actuator will be relatively low, even in case of high deviations from the desired path. When impedance is set to high values, the robot becomes stiffer, and its contribution in mobilizing the patient's limb is predominant, i.e., deviations from the desired path are smaller.

For non-backdrivable actuators, or when an accurate force control of the actuator is not practicable (e.g., lacking of joint force sensors), admittance control strategies are preferred. A closed-loop admittance control comprises an inner position controller and an outer torque controller. The desired actuator position $X_d(s)$ is generated based on the net force $F(s)$ applied to the robot and measured by a force sensor located at the robot end effector:

$$X_d(s) = \frac{F(s)}{Z_d(s)}. \quad (3)$$

In many cases, an adaptive law is chosen in order to adjust the value of the impedance within each movement, session, or day of treatment, according to specific performance parameters (Marchal-Crespo and Reinkensmeyer 2009):

$$G_{i+1} = fG_i + ge_i, \quad (4)$$

where i can be the current movement, session, or day, G is the value of the control parameter (e.g., robot stiffness), e is the performance error, and f and g are forgetting and gain factors, respectively. As example, some studies involving the endpoint manipulator MIT-Manus consider four performance measures, calculated based on the ability of the patient to (i) initiate the movement; (ii) move from the starting position to the target; (iii) maintain a linear trajectory toward the target; and (iv) reach the target position (Krebs et al. 2003). Control parameters such as movement time are then adapted with a forgetting factor $f = 1$. For an impedance control, by choosing a forgetting factor between 0 and 1, the robot impedance decreases when the performance error is small, making the rehabilitation exercise more challenging.

Assistive strategies are by far the most common in rehabilitation robotics and has been validated extensively in clinical trials. Nevertheless, according to the method adapted to encourage brain plasticity, other control strategies have been developed, including *challenge-based* algorithms (Marchal-Crespo and Reinkensmeyer 2009). These include all the controllers that operate by resisting user's movements. In their simplest implementation, constant resistive forces can be applied to the affected limb, regardless of the intended movement of the user. In other cases, an error-amplification control strategy can be employed. Opposed to error-cancellation strategies, it creates a divergent force field around the desired movement, in order to enhance kinematic errors and stimulate motor adaptation in performing the task. Resistive strategies can also be implemented by selectively cancelling gravity support to the robot, forcing

the user to contribute to the movement with higher effort.

Summary and Future Directions

Robot-aided rehabilitation is a well-established paradigm in current clinical practice. Robotic architectures for rehabilitation and related control strategies have been developed with a variety of features and the possibility to treat several pathologies and rehabilitate different body segments. A novel trend toward optimization of robot-aided rehabilitation is focused on the creation of robotic gyms, in which several robots are at the disposal of users having different levels of impairments, involving either upper or lower limbs or both. A robotic gym can constitute potentially a cost-effective solution to classic one-to-one therapy, by allowing several patients to perform their rehabilitation sessions at the same time, under the supervision of a single therapist. On the same direction, another cost-effective solution is the deployment of rehabilitation robots in home environment. The patient's home can be transformed into a place where the same rehabilitation treatment can be performed by means of personalized, game-like challenging exercises, guaranteeing a continuum of the rehabilitation process in the medium and long term. For this purpose, virtual reality systems can be employed to broaden the use scenarios, by creating a variety of engaging exercises and real-life tasks, whose difficulty can be graded to the user's capabilities. At the same time, remote monitoring of the patient's performance would allow therapists to keep track of the user's progress and modify the treatment accordingly.

Cross-References

- [Adaptive Control: Overview](#)
- [Force Control in Robotics](#)
- [Physical Human-Robot Interaction](#)
- [Robot Motion Control](#)

Bibliography

- Béjot Y, Bailly H, Durier J, Giroud M (2016) Epidemiology of stroke in Europe and trends for the 21st century. *Presse Med* 45:e391–e398. <https://doi.org/10.1016/j.lpm.2016.10.003>
- Borboni A, Mor M, Faglia R (2016) Gloreha—hand robotic rehabilitation: design, mechanical model, and experiments. *J Dyn Syst Meas Control* 138:111003. <https://doi.org/10.1115/1.4033831>
- Colombo G, Wirz M, Dietz V (2001) Driven gait orthosis for improvement of locomotor training in paraplegic patients. *Spinal Cord* 39:252–255. <https://doi.org/10.1038/sj.sc.3101154>
- Ercolini G, Trigili E, Baldoni A, et al (2018) A novel generation of ergonomic upper-limb wearable robots: design challenges and solutions. *Robotica* 1–17. <https://doi.org/10.1017/S0263574718001340>
- Haarman JA, Reenalda J, Buurke JH, et al (2016) The effect of ‘device-in-charge’ versus ‘patient-in-charge’ support during robotic gait training on walking ability and balance in chronic stroke survivors: a systematic review. *J Rehabil Assist Technol Eng* 3:205566831667678. <https://doi.org/10.1177/2055668316676785>
- Krebs H, Ferraro M, Buerger SP, et al (2004) Rehabilitation robotics: pilot trial of a spatial extension for MIT-Manus. *J Neuroeng Rehabil* 1:5. <https://doi.org/10.1186/1743-0003-1-5>
- Krebs HI, Dipietro L, Volpe BT, Hogan N (2003) Rehabilitation robotics? Performance-based progressive robot-assisted therapy. *Auton Robot* 15:7–20
- Marchal-Crespo L, Reinkensmeyer DJ (2009) Review of control strategies for robotic movement training after neurologic injury. *J Neuroeng Rehabil* 6:1–15. <https://doi.org/10.1186/1743-0003-6-20>
- Masia L, Xiloyannis M, Khanh DB, et al (2018) Actuation for robot-aided rehabilitation: design and control strategies. Roberto C, Vittorio S, Rehabilitation Robotics, Academic Press, Elsevier Ltd. pp 47–61. <https://doi.org/10.1016/B978-0-12-811995-2.00003-5>
- Nef T, Guidali M, Riener R (2009) ARMin III – arm therapy exoskeleton with an ergonomic shoulder actuation. *Appl Bionics Biomech* 6:127–142. <https://doi.org/10.1080/11762320902840179>
- Otten A, Voort C, Stienen A, et al (2015) LIMPACT: a hydraulically powered self-aligning upper limb exoskeleton. *IEEE/ASME Trans Mechatron* 20:2285–2298. <https://doi.org/10.1109/TMECH.2014.2375272>
- Pratt GA, Williamson MM (1995) Series elastic actuators. In: *Proceedings 1995 IEEE/RSJ international conference on intelligent robots and systems*, pp 399–406
- Stienen AH, Hekman EEG, Van Der Helm FCT, et al (2007) Freebal: dedicated gravity compensation for the upper extremities. In: *2007 IEEE 10th international conference on rehabilitation robotics ICORR'07* 00: 804–808. <https://doi.org/10.1109/ICORR.2007.4428517>

Reinforcement Learning and Adaptive Control

Girish Chowdhary, Girish Joshi, and Aaron Havens
University of Illinois at Urbana Champaign,
Urbana, IL, USA

Abstract

Reinforcement learning (RL) is a machine learning paradigm in which an agent attempts to learn a control policy that can generate the desired sequence of actions for achieving a higher level objective. RL promises to provide a learning mechanism via which autonomous agents can learn to control themselves directly through experience, without requiring manual coding of control policies. Similar to other machine learning paradigms, RL research heavily focuses on end-to-end learning, which in this case is learning of policies directly through experience. Recent successes of RL have shown that agents can learn to decision making and control policies on complex simulations for which it would have been very difficult to manually create control policies. Some examples include chess, go, and more recently complex continuous time-simulated domains. Some pressing issues include sample complexity, robustness, and reliable simulations to the real-world transfer.

Keywords

Control · Adaptation · Reinforcement learning

Introduction

Reinforcement learning (RL) is a machine learning paradigm in which an agent attempts to learn a control policy that can generate the desired sequence of actions for achieving a higher-level objective. RL promises to provide a learning

mechanism via which autonomous agents can learn to control themselves directly through experience, without requiring manual coding of control policies. Similar to other machine learning paradigms, RL research heavily focuses on end-to-end learning, which in this case is learning of policies directly through experience. Recent successes of RL have shown that agents can learn decision-making and control policies on complex simulations for which it would have been very difficult to manually create control. Some challenges and future directions of work include improving the sample complexity of finding policies, and increasing the robustness of learned policies to noise and dynamic changes.

RL in the Markov Decision Process Framework

A stochastic process $\{s_k\}$ is termed Markovian if the conditional probability of a future event s_{k+1} depends only on the event immediately preceding, and not the entire history of events. In particular, $p(s_{k+1}|s_k) = p(s_{k+1}|s_k, s_{k-1}, s_{k-2}, \dots)$. The Markov decision process (MDP) framework leverages this very key property to lay out a sequential decision-making framework: a Markov decision process (MDP) is a tuple $\langle \mathcal{S}, \mathcal{A}, R, \mathcal{P}, \gamma \rangle$, with state space $\mathcal{S}$, action space $\mathcal{A}$, reward function $r(s, a) \in \mathbb{R}$, transition function $p(s_{k+1}|s_k, a_k) \in \mathcal{P}$, and discount factor $\gamma \in [0, 1)$. The goal of the decision-making agent is to find a sequence of actions a_k such that the reward is maximized. The reward can be sparse and delayed, that is, it is only available at some states, or only available after a sequence of actions are performed. This makes the MDP sequential decision-making problems quite different and rather interesting. As such, the formulation we discuss below assumes that $\gamma < 1$ and is necessary for infinite horizon sequential decision-making problems. In finite horizon problems, such as those studied in Bertsekas et al. (1995) and Bertsekas and Tsitsiklis (1996), γ can take other values.

Given a state s_k , a deterministic non-stationary policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that outputs an action $a_k = \pi(s_k)$. The goal of MDP problems is to seek out a policy that leads to maximum reward. To further formulate the problem, consider that starting from an initial state s_0 , the MDP following a policy π will result in a trajectory $\zeta = \{[s_0, a_0, r_0], [s_1, a_1, r_1], \dots\}$. Let us now define the so-called Q-function of each state-action pair (s, a) under policy π as the expected sum of discounted reward obtained by starting at any state s and following a policy π thereafter after taking the action $a = \pi(s)$:

$$Q^\pi(s, a) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r(s_k, a_k) | s_0 = s \right]. \quad (1)$$

Note here that the expectation is on π , that is, following a different policy will lead to a different expected Q-value even when one starts on the same state. The Q-function is defined slightly (but very importantly) different than the V-function or the value function V^π for each state s under policy π . The value function is defined only over s (and not over (s, a)) as $V^\pi(s) = E_\pi[\sum_{k=0}^{\infty} \gamma^k r(s_k, a_k) | s_0 = s]$. That is, it is defined simply as the expected sum of discounted rewards obtained by following the policy π starting from the state s . The goal is to find the optimal policy that maximizes the expected cumulative discounted reward in all states, that is, to find $\pi^*(s) = \arg \max_\pi V^\pi(s)$. Here note that the maximum is over the policy π . The value function of the optimal policy π^* is termed as V^* .

The Bellman equation states that the value of taking action a_k in state s_k under the policy π equals the sum of the immediate reward and the discounted value achieved by following the policy π :

$$Q^\pi(s_k, a_k) = r(s_k, a_k) + \gamma Q^\pi(s_k, \pi(s_k, a_k)). \quad (2)$$

It is rather straightforward to derive this equation, by noting that $Q^\pi(s_k, a_k) = \sum_{k=0}^{\infty} \gamma^k r(s_k, a_k) =$

$r(s_k, a_k) + \gamma \sum_{k=1}^{\infty} \gamma^{k-1} r(s_k, \pi(s_k))$. Further expanding the sum leads to the recursion in (2).

Similarly, it can be shown that the optimal Q function Q^* follows the Bellman optimality equation $Q^*(x, a) = r(x, a) + \gamma \max_{a'} Q^*(x, a')$. The V -function also satisfies the optimality equation $V^*(s_k) = \max_a (r(s_k, a_k) + \gamma V^*(s_{k+1}))$; however note that to find the optimal action, the knowledge of s_{k+1} or alternatively $\mathcal{P}$ is necessary. On the other hand, with the Q -function

$$\pi^*(s_k) = \arg \max_a Q^*(s_k, a). \quad (3)$$

What we have presented is one of the most commonly studied MDP formulations. One key variant is the formulation with stochastic policies, where the policy is a distribution over actions, and given a state s_k , the action is sampled from the policy distribution $a_k \sim \pi(s_k)$.

There are many algorithms available to solve the MDP problem in the dynamic programming setting when $\mathcal{R}$ and $\mathcal{P}$ are known. Most of these leverage the fact that the recursion in Equation 2 is a contraction. Therefore starting with any policy π , it is possible to follow first a policy evaluation step to populate the V - or the Q -function and then a policy improvement step where for each state the policy $\pi(s) \leftarrow \arg \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \mathcal{P}(r(s, a) + \gamma V^\pi(s'))$. Several of these algorithms are discussed in detail in Geramifard et al. (2013).

In this entry, we focus on the so-called reinforcement learning (RL) problem, in which $\mathcal{R}$ and $\mathcal{P}$ are assumed to be unknown, and the agent figures out an optimal policy through experience. There are many algorithms available to solve the RL problem, some of which are summarized in Fig. 1. We provide details of some key methods now.

Q-Learning and Other Value-Based Methods

Using the Markov assumption on the transition model and the Bellman formula, recursive update expression for learning the state-action value function $Q^\pi(s_t, a_t)$ can be defined as

$$Q^\pi(s_k, a_k) = r(s_k, a_k) + \gamma Q^\pi(s_{k+1}, \pi(s_{k+1}))$$

The above definition leads us to the famous Q-learning and SARSA algorithms, which use the following update form

$$Q^\pi(s_k, a_k) \leftarrow Q^\pi(s_k, a_k) + \alpha \delta$$

where α is the learning rate and δ is defined as temporal-difference (TD-error). Different choice of TD δ error leads to different algorithm

- Q-Learning: $\delta = r(s_k, a_k) + \gamma \max_a Q^\pi(s_{k+1}, a) - Q^\pi(s_k, a_k)$
- SARSA: $\delta = r(s_k, a_k) + \gamma Q^\pi(s_{k+1}, \pi(s_{k+1})) - Q^\pi(s_k, a_k)$

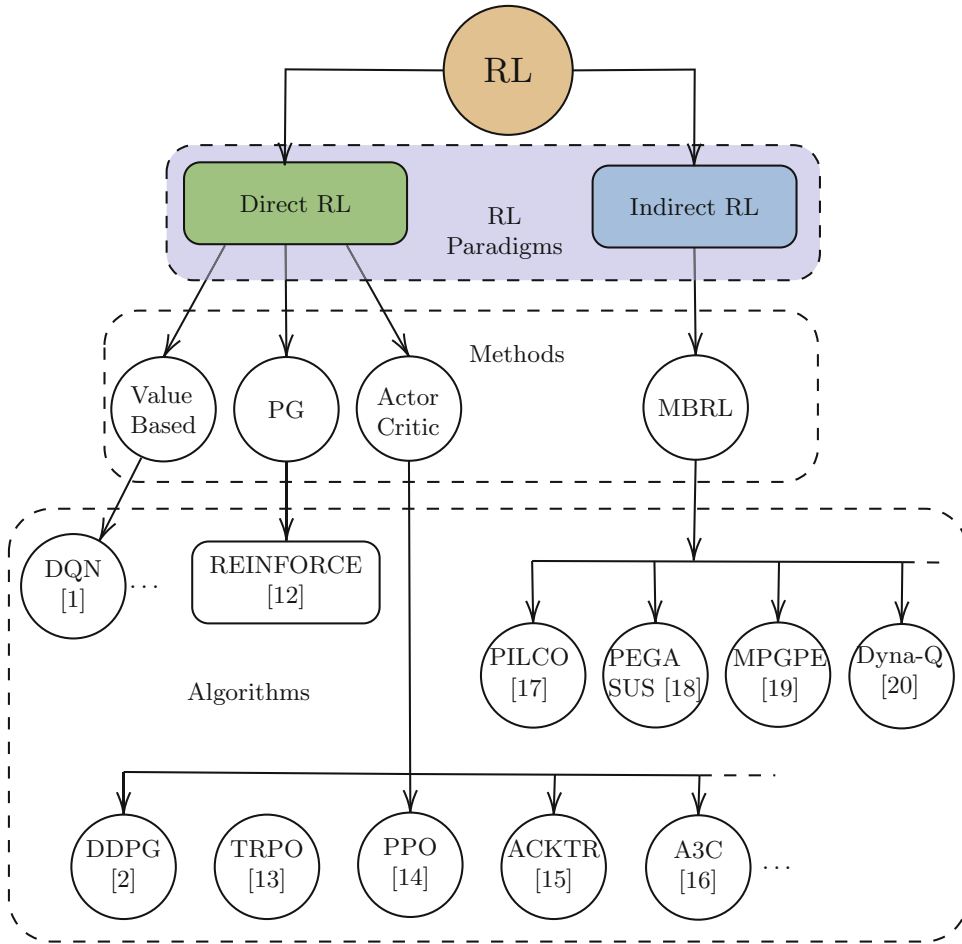
For large or continuous state-action spaces, a tabular representation quickly becomes intractable (Geramifard et al. 2013). This issue can be resolved by using a function approximator, such as linear function approximators (Geramifard et al. 2013), Gaussian processes (Chowdhary et al. 2014; Liu et al. 2018; Kuss 2006), or neural networks (Mnih et al. 2015, 2016; Bertsekas and Tsitsiklis 1996; Schulman et al. 2017, 2015b). Deep Q-networks (DQN) is one such algorithm that has solved instability issues of approximate Q-learning with deep neural network function approximators and techniques like experience replay and soft update of the network (Mnih et al. 2015). DQN uses a cost function of the form

$$J(\theta) = \mathbb{E}_{\{s, a, r, s'\} \sim \mathcal{B}} \left(r + \gamma \max_{a'} Q(s', a', \theta^-) - Q(s, a, \theta) \right)^2$$

where the samples $\{s, a, r, s'\}$ are drawn from the replay buffer $\mathcal{B}$ and θ^- is frozen network parameter from last update.

Policy Gradient (PG) and Actor-Critic Methods

Policy gradient method aims at optimizing a parameterized policy $\pi_\theta(a_t | s_t)$ resulting in the higher total discounted return. The total discounted return the policy gradient aims to



Reinforcement Learning and Adaptive Control,

Fig. 1 Taxonomy of Reinforcement Learning Algorithms: A3C Mnih et al. (2016), ACKTR Wu et al. (2017), DQN Mnih et al. (2015), DDPG Lillicrap et al. (2015), DYNA-Q Sutton (1991), MPGPE Tangkaratt

et al. (2014), PEGASUS Ng and Jordan (2000), PILCO Deisenroth and Rasmussen (2011), PPO Schulman et al. (2017), REINFORCE Sutton et al. (2000), TRPO Schulman et al. (2015b)

R

maximize is defined as

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}(\tau)} \left[\sum_{\tau} r(s_k, a_k) \right]$$

where the $\pi_{\theta}(\tau)$ is the probability of the trajectory and is defined as

$$\begin{aligned} \pi_{\theta}(\tau) &= \pi_{\theta}(s_0, a_0, \dots, s_T, a_T) \\ &= \rho(s_0) \prod_{k=1}^T \pi_{\theta}(a_k | s_k) p(s_{k+1} | s_k, a_k) \end{aligned}$$

PG uses gradient ascent for maximizing the total expected return $J(\theta)$. The policy gradient update is given as

$$\theta = \theta + \alpha \nabla_{\theta} J(\theta)$$

where $\nabla_{\theta} = \mathbb{E}_{\tau \sim \pi_{\theta}(\tau)} [\nabla_{\theta} \log \pi_{\theta}(\tau) R(\tau)]$. The gradient calculation using likelihood ratio is first introduced in REINFORCE paper by Williams et al. (Sutton et al. 2000). This method makes use of likelihood ratio trick given by identity $\nabla_{\theta} \pi_{\theta}(\tau) = \pi_{\theta}(\tau) \nabla_{\theta} \log \pi_{\theta}(\tau)$ in policy-gradient theorem.

Several variants of the policy gradient can be constructed by replacing the total return performance metric in the gradient expression as follows: (Schulman et al. 2015a)

$$\nabla_{\theta} = \mathbb{E}_{\tau \sim \pi_{\theta}(\tau)} [\nabla_{\theta} \log \pi_{\theta}(\tau) \zeta(\tau)].$$

Let $\zeta(\tau)$ be the placeholder for total return. Using different estimates of the total return, we can build different policy gradient algorithms, for example, $\zeta \tau = R(\tau) = \sum_{k=1}^T r(s_k, a_k)$ is the vanilla REINFORCE algorithm which uses bootstrapped total return on trajectory. The causality correction to the total return estimate for gradient variance reduction in REINFORCE uses $\zeta \tau = \sum_{k'=1}^T r(s_{k'}, a_{k'})$ (Sutton et al. 2000). A baseline corrected return for variance reduction of gradient estimate employs $\zeta \tau = \sum_{k=1}^T r(s_k, a_k) - b_k$, where b_k can be average reward (Peters and Schaal 2006). An actor-critic algorithm uses $\zeta \tau = Q^{\pi}(s_k, a_k)$ (Peters and Schaal 2008), while an advantage actor-critic algorithm would use $\zeta \tau = A^{\pi}(s_k, a_k)$, where A^{π} is the advantage function (Mnih et al. 2016).

Model-Based RL or “Indirect RL”

In model-based RL a goal is to attempt to learn $\mathcal{T}$ and $\mathcal{R}$ from state-action-reward tuples and use this in learning the policy. Once $\mathcal{T}$ and $\mathcal{R}$ are learned, they may be used in any capacity to solve the MDP. The open-ended nature of how to use the model in this setting has brought about much debate and discussion on what constitutes model-based RL, as one could argue that model-free uses the value function and MDP assumption as a “model.” Generally model-based RL aims take advantage of structure in the dynamics and cost model resulting in orders of magnitude improvements in data efficiency over model-free methods, which is essential for some real-world applications. Modern methods typically fall under two different approaches. The first category strongly resembles system identification where one iteratively estimates a global dynamics model through interactions and uses the model to plan an optimal policy as in model predictive control (MPC), usually via a shooting method. One example is probabilistic ensembles with trajectory sampling (PETS) (Chua et al. 2018): a probabilistic

dynamic model is learned via a model ensemble through environment interactions. The optimal control is given by a sampling-based shooting method using the learned probabilistic model. Another approach is data-efficient reinforcement learning with probabilistic model predictive control (Kamthe and Deisenroth 2018): a probabilistic model is learned using a Gaussian process in order to account for long-term prediction uncertainty. The optimal control is found using MPC and analytic gradients computed via maximum principle.

The second category makes use of global or local models to do policy optimization directly as in the model-free case either by using the model derivatives to compute analytic policy updates or sampling the update using the learned dynamics model. In probabilistic inference for learning control (PILCO) (Deisenroth and Rasmussen 2011), a probabilistic model is learned using a Gaussian process and exploits analytic gradients of resulting distribution to optimize the policy. In Guided Policy Search (GPS) (Levine and Koltun 2013), locally linearized models of the dynamics and cost function are used to iteratively update a global policy.

The model-based assumption can also become a detriment. The policy is biased by the chosen model parameterization and data distribution and may result in compounding errors in planning or poor local solutions in policy optimization. This is why the most successful model-based methods emphasize model uncertainty representation. Many ongoing works aim to investigate how to most effectively and generally exploit model assumptions and scale to arbitrarily problems with high-dimensional observations.

Connections Between RL and Adaptive Control

Both RL and adaptive control are frameworks designed to find control policies when the model transitions are not known. Adaptive control has classically considered control of continuous-time systems using state-space models and well-defined tracking objectives (Tao 2003; Narendra and Balakrishnan 1997; Chowdhary et al. 2015;

Calise et al. 2001; Åström and Wittenmark 1995), while RL has classically focused on decision-making policies in discrete-state MDPs. The key power of RL is that reward can be sparse or “delayed,” that is, the agent does not need to have a reward at every time step k , rather only in certain states. For example, an agent trying to find a path on a map need only obtain the reward when it reaches its goal, and not necessarily along the way. This is in contrast with traditional adaptive control problems such as model reference adaptive control in which the agent receives a tracking error reward at every time step. Traditionally, continuous-time RL problems are considered difficult to solve due to (1.) challenges in learning the right approximate representation for continuous domain state-action spaces (Geramifard et al. 2013; Kaelbling et al. 1996, 1998; Sutton and Barto 1998) and (2.) excessive number of experiments required to handle the explore-exploit dilemma in continuous state spaces (Sutton and Barto 1998; Busoniu et al. 2010; Axelrod and Chowdhary 2015). Recently advances in computing power and training of deep neural networks for supervised learning have rekindled and fueled a number of works focused on solving continuous-time control problems with RL in simulation (Duan et al. 2016; Lillicrap et al. 2015). The results are promising, given that the agent can learn high-dimensional control tasks merely from trials; however, transition from simulation to real world has been a challenge, especially in the face of results that show that the learned policies may not transfer well across similar systems or degrade rapidly when parameters are only slightly varied. In essence, these works have focused on learning control policies through experience, but are not yet equipped to adapt the policies in face of change. There has been interest in also using the RL frameworks for solving classical control problems (Lewis et al. 2012; Modares et al. 2014; Kiumarsi et al. 2014).

Summary and Future Directions

In summary, given enough samples and time, RL can learn policies for complex tasks. However,

much work remains to be done in figuring out principled mechanisms to quickly and efficiently transfer policies learned between domains to bring RL to the real world. A significant body of literature on transfer in RL is focused on initialized RL in the target domain using learned source policy, known as jump-start/warm-start methods (Taylor et al. 1999, 2005; Ammar et al. 2012, 2015; Banerjee and Stone 2007; Peters and Schaal 2006; Peng et al. 2017b; Ross et al. 2011; Yan et al. 2017). However, warm-start, which has been reasonably successful for supervised learning, can often lead to mixed and even negative results in RL (Joshi and Chowdhary 2018; Taylor and Stone 2009). Efforts are also being made in exploring accelerated learning directly on real robots, through Guided Policy Search (GPS) (Levine et al. 2015), and parallelizing the training across multiple agents using meta-learning (Levine et al. 2016; Nagabandi et al. 2018; Zhu et al. 2018). However, these and other reported architectures do not *necessarily* lead to improved sample efficiency, nor are guaranteed to handle relatively minor changes in the transition model, and are even known to cause negative transfer. In essence, many exciting research directions remain in ensuring RL fails gracefully when the underlying model changes or has robustness margins like linear controllers. Hierarchical RL and behavioral adaptation are accordingly also receiving increasing attention (Joshi and Chowdhary 2018).

Cross-References

- [Robot Learning](#)
- [Stochastic Dynamic Programming](#)

Acknowledgments This work was supported in part by Navy #N00014-19-1-2373 and NSF-CPS NIFA award #2018-67007-28379

Bibliography

- Åström KJ, Wittenmark B (1995) Adaptive control, 2nd edn. Addison-Wesley, Reading
- Ammar HB, Tuyls K, Taylor ME, Driessens K, Weiss G (2012) Reinforcement learning transfer via sparse coding. In: Proceedings of the 11th international

- conference on autonomous agents and multiagent systems, vol 1. International Foundation for Autonomous Agents and Multiagent Systems, pp 383–390
- Ammar HB, Eaton E, Ruvoilo P, Taylor ME (2015) Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold alignment. In: Proceedings of the AAAI
- Axelrod A, Chowdhary G (2015) The explore-exploit dilemma in nonstationary decision making under uncertainty. In: The explore-exploit dilemma in nonstationary decision making under uncertainty, ser 2198–4182, 1st edn. Springer International Publishing. <https://www.springerprofessional.de/en/the-explore-exploit-dilemma-in-nonstationary-decision-making-und/7454158>
- Banerjee B, Stone P (2007) General game learning using knowledge transfer. In: IJCAI, pp 672–677
- Bertsekas DP, Tsitsiklis JN (1996) Neuro-dynamic programming, vol 5. Athena Scientific Belmont
- Bertsekas DP, Bertsekas DP, Bertsekas DP, Bertsekas DP (1995) Dynamic programming and optimal control. Athena Scientific, Belmont
- Busoniu L, Babuska R, Schutter BD, Ernst D (2010) Reinforcement learning and dynamic programming using function approximators, 1st edn. CRC Press
- Calise A, Hovakimyan N, Idan M (2001) Adaptive output feedback control of nonlinear systems using neural networks. *Automatica* 37(8):1201–1211. Special issue on Neural Networks for Feedback Control
- Chowdhary G, Liu M, Grande R, Walsh T, How J, Carin L (2014) Off-policy reinforcement learning with gaussian processes. *IEEE/CAA J Automat Sin* 1(3):227–238
- Chowdhary G, Kingravi HA, How JP, Vela PA (2015) Bayesian nonparametric adaptive control using gaussian processes. *IEEE Trans Neural Netw Learn Syst* 26(3):537–550
- Chua K, Calandra R, McAllister R, Levine S (2018) Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In: Advances in Neural Information Processing Systems 31, Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R, Eds. Curran Associates, Inc., pp 4754–4765 [Online]. Available: <http://papers.nips.cc/paper/7725-deep-reinforcement-learning-in-a-handful-of-trials-using-probabilistic-dynamics-models.pdf>
- Deisenroth M, Rasmussen CE (2011) Pilco: a model-based and data-efficient approach to policy search. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp 465–472
- Duan Y, Chen X, Houthoofd R, Schulman J, Abbeel P (2016) Benchmarking deep reinforcement learning for continuous control. In: International conference on machine learning, pp 1329–1338
- Geramifard A, Walsh TJ, Tellex S, Chowdhary G, Roy N, How JP et al (2013) A tutorial on linear function approximators for dynamic programming and reinforcement learning. *Found Trends Mach Learn* 6(4):375–451
- Heess N, Sriram S, Lemmon J, Merel J, Wayne G, Tassa Y, Erez T, Wang Z, Eslami A, Riedmiller M, et al (2017) Emergence of locomotion behaviours in rich environments. arXiv preprint arXiv:1707.02286
- Joshi G, Chowdhary G (2018) Cross-domain transfer in reinforcement learning using target apprentice. In: 2018 IEEE international conference on robotics and automation (ICRA). IEEE, pp 7525–7532
- Kaelbling LP, Littman ML, Moore AW (1996) Reinforcement learning: a survey. *J Artif Intell Res* 4:237–285
- Kaelbling L, Littman M, Cassandra A (1998) Planning and acting in partially observable stochastic domains. *Artif Intell* 101:99–134
- Kamthe S, Deisenroth M (2018) Data-efficient reinforcement learning with probabilistic model predictive control. In: International conference on artificial intelligence and statistics, pp 1701–1710
- Kiumarsi B, Lewis FL, Modares H, Karimpour A, Naghibi-Sistani M-B (2014) Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. *Automatica* 50(4):1167–1175
- Kuss M (2006) Gaussian process models for robust regression, classification, and reinforcement learning. Ph.D. dissertation, Technische Universität Darmstadt
- Levine S, Koltun V (2013) Guided policy search. In: International conference on machine learning, pp 1–9
- Levine S, Wagener N, Abbeel P (2015) Learning contact-rich manipulation skills with guided policy search. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp 156–163
- Levine S, Pastor P, Krizhevsky A, Quillen D (2016) Learning hand-eye coordination for robotic grasping with large-scale data collection. In: International symposium on experimental robotics. Springer, pp 173–184
- Lewis FL, Vrabie D, Syrmos VL (2012) Optimal control. John Wiley & Sons, Hoboken
- Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2015) Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971
- Liu L, Hodgins J (2017) Learning to schedule control fragments for physics-based characters using deep Q-learning. *ACM Trans Graph (TOG)* 36(3):29
- Liu M, Chowdhary G, Da Silva BC, Liu S-Y, How JP (2018) Gaussian processes for learning and control: a tutorial with examples. *IEEE Control Syst Mag* 38(5):53–86
- Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533
- Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. In: International conference on machine learning, pp 1928–1937
- Modares H, Lewis FL, Naghibi-Sistani M-B (2014) Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown

- constrained-input continuous-time systems. *Automatica* 50(1):193–202
- Nagabandi A, Kahn G, Fearing RS, Levine S (2018) Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In: 2018 IEEE international conference on robotics and automation (ICRA). IEEE, pp 7559–7566
- Narendra KS, Balakrishnan J (1997) Adaptive control using multiple models. *IEEE Trans Autom Control* 42(2):171–187
- Ng AY, Jordan M (2000) Pegasus: a policy search method for large MDPs and POMDPs. In: Proceedings of the sixteenth conference on uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., Stanford CA, pp 406–415
- Peng XB, Berseth G, Van de Panne M (2016) Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Trans Graph (TOG)* 35(4):81
- Peng XB, Berseth G, Yin K, Van De Panne M (2017a) Deeploco: dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans Graph (TOG)* 36(4):41
- Peng XB, Andrychowicz M, Zaremba W, Abbeel P (2017b) Sim-to-real transfer of robotic control with dynamics randomization. *arXiv preprint arXiv:1710.06537*
- Peters J, Schaal S (2006) Policy gradient methods for robotics. In: 2006 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 2219–2225
- Peters J, Schaal S (2008) Natural actor-critic. *Neurocomputing* 71(7):1180–1190
- Ross S, Gordon G, Bagnell D (2011) A reduction of imitation learning and structured prediction to no-regret online learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp 627–635
- Schulman J, Moritz P, Levine S, Jordan MI, Abbeel P (2015a) High-dimensional continuous control using generalized advantage estimation. *CoRR*, abs/1506.02438
- Schulman J, Levine S, Abbeel P, Jordan MI, Moritz P (2015b) Trust region policy optimization. In: ICML, pp 1889–1897
- Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*
- Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D (2016) Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489
- Sutton RS (1991) Integrated modeling and control based on reinforcement learning and dynamic programming. In: Advances in neural information processing systems, pp 471–478
- Sutton RS, Barto AG (1998) Reinforcement learning: An introduction, vol 1, no 1. MIT Press, Cambridge
- Sutton RS, McAllester DA, Singh SP, Mansour Y (2000) Policy gradient methods for reinforcement learning with function approximation. In: Advances in neural information processing systems, pp 1057–1063
- Tangkaratt V, Mori S, Zhao T, Morimoto J, Sugiyama M (2014) Model-based policy gradients with parameter-based exploration by least-squares conditional density estimation. *Neural Netw* 57:128–140
- Tao G (2003) Adaptive control design and analysis. New York: Wiley
- Taylor ME, Stone P (2009) Transfer learning for reinforcement learning domains: a survey. *J Mach Learn Res* 10:1633–1685
- Taylor ME, Stone P, Liu Y (1999, 2005) Value functions for RL-based behavior transfer: a comparative study. In: Proceedings of the national conference on artificial intelligence, vol 20, no 2. AAAI Press/MIT Press, Menlo Park/London/Cambridge, MA, p 880
- Vinyals O, Ewalds T, Bartunov S, Georgiev P, Vezhnevets AS, Yeo M, Makhzani A, Küttler H, Agapiou J, Schrittwieser J et al (2017) Starcraft II: a new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*
- Wu Y, Mansimov E, Liao S, Grosse R, Ba J (2017) Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *Adv Neural Inf Proces Syst* pp 5279–5288
- Yan M, Frosio I, Tyree S, Kautz J (2017) Sim-to-real transfer of accurate grasping with eye-in-hand observations and continuous control. *arXiv preprint arXiv:1712.03303*
- Zhu H, Gupta A, Rajeswaran A, Levine S, Kumar V (2018) Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. *arXiv preprint arXiv:1810.06045*

Reinforcement Learning for Approximate Optimal Control

R

Warren E. Dixon

Department of Mechanical and AeroMechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA

Abstract

Approximate dynamic programming is an approach to balance computational demands with optimal decision-making. Dynamic programming is a feedback control method that generates an optimal policy, which is obtained by solving the Hamilton-Jacobi-Bellman

(HJB) equation. Since the solution to the HJB is intractable in general and requires exact knowledge of the system dynamics, actor-critic-based reinforcement learning (RL) methods can be used to approximate the value function and, hence, the optimal controller. Emerging advances in approximate dynamic programming are described that focus on the computational expense to minimize a defined cost functional, including RL approaches that seek to rapidly identify the optimal value function.

Keywords

Reinforcement learning · Nonlinear systems · Lyapunov analysis · Approximate dynamic programming · Neural network

Introduction

Optimal control problems focus on minimizing a trade-off between an end goal and the action required to reach the goal. Mathematically, for a controlled dynamical system such as

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \quad (1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ are locally Lipschitz functions, $x : \mathbb{R}_{\geq t_0} \rightarrow \mathbb{R}^n$ denotes the system state, $u : \mathbb{R}_{\geq t_0} \rightarrow U \subset \mathbb{R}^m$ denotes the control input, and U denotes the action space, the objective for optimal control problems can be stated as the desire to minimize a cost functional such as

$$J(t_0, x_0, u) = \int_{t_0}^{t_f} L(x(t), u(t), t) dt + \Phi(x(t_f)), \quad (2)$$

under a set of constraints (e.g., dynamics of the state, state boundaries, and limits on the input rate or magnitude), where $L : \mathbb{R}^n \times U \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ is the Lagrange cost, $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$ is the Mayer cost, t_0 is the initial time, and t_f and x_f denote the final time and state, respectively. Solutions to the optimal control problem typically either use Pontryagin's maximum principle or dynamic

programming. Controllers based on the maximum principle seek to minimize (2) starting from a specific state, whereas dynamic programming yields controllers that minimize (2) starting from any state (i.e., generate optimal policies). The key difference is that the controller derived from dynamic programming is a feedback controller, in the sense that it is guaranteed to be optimal for any initial condition of the dynamical system, whereas the maximum principle generates the optimal state, co-state, and control trajectories for a given initial condition, and the resulting controller must be implemented in an open-loop manner, because if the initial condition changes, the optimal solution is no longer valid and the optimal controller must be redetermined (Kamalapurkar et al. 2018). However, developing an optimal feedback controller carries a computational cost associated with solving the Hamilton-Jacobi-Bellman (HJB) equation, which is a nonlinear partial differential equation that involves exponential growth in numerical complexity as the dimension of the state space increases. Given the complexity of solving the HJB equation, this entry focuses on approximate dynamic programming (ADP) using reinforcement learning (RL) as a framework for developing a controller to minimize the unconstrained infinite-horizon cost given by

$$J(t_0, x_0, u) = \int_{t_0}^{\infty} Q(x(\tau)) + u^T(\tau)Ru(\tau)d\tau, \quad (3)$$

where $Q : \mathbb{R}^n \rightarrow \mathbb{R}$ is a positive definite function and $R \in \mathbb{R}^{m \times m}$ is a symmetric positive definite matrix. For nonlinear systems that are affine in the controller as in (1) with cost functions that are quadratic in the controller as in (3), under the assumptions that at least one admissible controller exists and the optimal control problem has a continuously differentiable value function, the controller $u(t) = u^*(x(t))$ where the policy $u^* : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is defined as

$$u^*(x) = -\frac{1}{2}R^{-1}g^T(x)(\nabla_x V^*(x))^T \quad (4)$$

is the optimal controller, where $V^*(x)$ is the time-independent optimal value function (i.e., cost-to-go) given by

$$V^*(x) \triangleq \inf_{u[t,\infty]} J(t, x, u). \quad (5)$$

Dynamic Programming Overview

For systems with finite state and action spaces, dynamic programming techniques such as policy iteration and value iteration (Bertsekas and Tsitsiklis 1996; Sutton and Barto 1998) are effective methods for optimal control synthesis, although drawbacks include the aforementioned curse of dimensionality and the need for perfect knowledge of the dynamics. Iterative/recursive (also called simulation-based) RL techniques such as Q-learning and temporal difference learning (Sutton and Barto 1998) avoid the need for exact model knowledge. Neuro-dynamic programming (cf. Werbos et al. 1992; Bertsekas and Tsitsiklis 1996 and the references therein) extends such methods to general finite state spaces through parametric approximation of the policy, replacing the arbitrarily large state space with a compact set of weights that encode the value function. Actor-critic RL is a popular method for generalized policy iteration algorithms: detailed reviews of actor-critic methods are provided in Kamalapurkar et al. (2018), Grondman et al. (2012), and Lewis and Liu (2013). In these methods, the critic estimates the value function (e.g., through temporal difference or heuristic dynamic programming), and the actor directly minimizes the estimated value function or the Bellman error (also known as the temporal difference error). The iterative nature of actor-critic methods is amenable to and has fueled a growing interest in discrete-time approximate optimal control problems over the past decade (Bhatnagar et al. 2009; Kiumarsi et al. 2017). Extensions to continuous time domains are also an active open area of research with recent advances providing convergence and stability guarantees (e.g., through a Lyapunov-based analysis) that dictate the design

of adaptive updates that adjust the actor and critic estimates.

Actor-Critic Approximation

In an approximate actor-critic-based solution, the optimal value function is replaced by a parametric estimate such as $\hat{V}(x, \hat{W}_c) \triangleq \hat{W}_c^T \sigma(x)$, and the optimal policy is replaced by a parametric estimate such as $\hat{u}(x, \hat{W}_a) \triangleq -\frac{1}{2} R^{-1} g(x)^T \nabla_x \sigma^T(x) \hat{W}_a$ where $\hat{W}_c \in \mathbb{R}^L$ and $\hat{W}_a \in \mathbb{R}^L$ denote vectors of estimates of the ideal parameters for the critic and actor, respectively, $\sigma: \mathbb{R}^n \rightarrow \mathbb{R}^L$ is a continuously differentiable nonlinear activation function, and L denotes the number of unknown parameters. After replacing the optimal controller and ideal value function in (4) and (5), respectively, with their estimates, the resulting Bellman error $\delta: \mathbb{R}^n \times \mathbb{R}^L \times \mathbb{R}^L \rightarrow \mathbb{R}$ is defined as

$$\begin{aligned} \delta(x, \hat{W}_c, \hat{W}_a) \triangleq & \nabla_x \hat{V}(x, \hat{W}_c) (f(x) \\ & + g(x) \hat{u}(x, \hat{W}_a)) \\ & + L(x, \hat{u}(x, \hat{W}_a)), \end{aligned} \quad (6)$$

where $\delta(x(t), \hat{W}_c(t), \hat{W}_a(t)) = 0$ when the approximations are exact (i.e., the Bellman error provides a measure of suboptimality). Computation of the Bellman error in (6) requires exact knowledge of (1). To provide robustness to uncertainty in (1), integral RL (e.g., Chapter 3 of Vrabie et al. 2013) is a method that uses an integral form of the HJB equation. An alternative approach is to estimate the state derivative directly as in Bhasin et al. (2013), where the critic computes the Bellman error at each time instance using an estimate of the state derivative. Another alternative is to simultaneously learn the dynamics while also learning the value function and hence optimal policy (e.g., an actor-critic-identifier strategy Bhasin et al. 2013). Such methods highlight the complexity of simultaneously producing an admissible policy while adapting

to uncertainty in the dynamics and learning the optimal policy.

In real-time implementations of RL, the approximated policy is used to control the system; therefore, both stability and optimality concerns dictate the need to obtain a good approximation of the value function (i.e., system identification in the sense that the approximated parameters converge to the ideal parameters). Typically, the objective in adaptive control problems is simply boundedness of the parameter estimates, and convergence to the actual parameters is achieved only if a sufficient exploration condition (i.e., persistence of excitation (PE)) is satisfied. In general, and especially for nonlinear systems, the traditional PE condition is impossible to guarantee a priori or verify during execution. Hence, stochastic systems pursue a randomized stationary policy, and an ad hoc exploration signal is often added for deterministic systems, but the type of signal and length of time a signal should be added are unsolved problems. Moreover, to approximate the total cost over an operating domain, the whole domain should be explored (i.e., $x(t)$ should pass through the entire operating domain) for the critic to learn the value function. The trade-off between the need to explore the operating domain to exactly learn the value function versus the need to perform actions to minimize the cost is the classic exploration versus exploitation problem.

For model-free approaches (e.g., integral RL), exploration conditions can be relaxed using experience replay Modares et al. (2014), where each evaluation of the Bellman error is stored in a history stack; however, a finite amount of exploration is still required since the values stored in the history stack are also constrained to the system trajectory. Similar strategies have been explored for stochastic systems by solving the optimal control problem off-line in the sense that repeated learning trials need to be performed before the algorithm learns the controller. Such model-free approaches are often limited to on-policy learning, in the sense that the integral Bellman error is only meaningful if it is evaluated along the system trajectories, and likewise, state derivative estimators can only generate numerical

estimates of the state derivative along the system trajectories.

Model-based approaches have the advantage that the model can be used to evaluate the Bellman error at any number of desired points in the state space. Specifically, the state derivative and, hence, the Bellman error can be evaluated at any point in the domain. As described in Kamalapurkar et al. (2018), the use of a model to evaluate the Bellman error at an unexplored point in the state space can be interpreted as a simulation of experience (i.e., the value function can be estimated at any arbitrary point x_i in the state space without actually visiting x_i). The ability to learn the value function from off-policy exploration provides a parallelization of learning that results in simultaneous exploration and exploitation. These methods can even be exploited when the model is uncertain through actor-critic-identifier methods (Bhasin et al. 2013). Concurrent learning (Chowdhary and Johnson 2011; Parikh et al. 2019) is a process where each input-output data point is stored in a history stack that can be used to formulate a PE-like condition to determine sufficient exploration. The significant advantage of concurrent learning is that the resulting sufficient excitation condition involves a minimum eigenvalue computed over a summation of data. While the time (or number of data points) required to obtain sufficient excitation is unknown a priori, the condition can be checked at each time instance and verified online, allowing exponential convergence of the value function approximates to the actual value function.

Summary and Future Directions

By its very nature, optimal control involves a trade-off between a goal and the effort to obtain the goal, encoded in a cost functional. Approximate optimal control methods are motivated by a further trade-off between the minimal cost and the required computational resources and hence time required to make a decision. RL-based ADP methods are a mainstream research focus that offsets the computational demands associated with

a continuous state space by a parameterization of the optimal value function by a finite set of weights and basic functions. The challenge in such problems is to minimize the cost by rapidly learning the exact value function (and potentially the system dynamics). As described previously, to resolve the resulting exploration versus exploitation conundrum, new research directions are exploiting parallelized/concurrent learning to simultaneously execute actions while learning the uncertainty, including the use of stored data and extrapolation of states for off-policy learning (Sledge et al. 2018; Chowdhary et al. 2013; Wawrzyński 2009). Additional emerging directions include learning methods that yield finite-time learning (cf. Gupta et al. 2019; Hong et al. 2006; Karafyllis and Krstic 2018). Such methods could facilitate learning for finite-time optimal control problems and can potentially enable the inclusion of more complex problems that involve a hybrid mixture of continuous and discrete dynamics where switching occurs between subsystems, which may involve the development of dwell-time conditions.

Cross-References

- ▶ [Nonlinear Adaptive Control](#)
- ▶ [Nonparametric Techniques in System Identification](#)
- ▶ [Optimal Control and the Dynamic Programming Principle](#)
- ▶ [Optimal Sampled-Data Control](#)
- ▶ [Reinforcement Learning and Adaptive Control](#)
- ▶ [Reinforcement Learning for Approximate Optimal Control](#)
- ▶ [Reinforcement Learning for Control Using Value Function Approximation](#)
- ▶ [Stability and Performance of Complex Systems Affected by Parametric Uncertainty](#)

Bibliography

Bertsekas D, Tsitsiklis J (1996) *Neuro-dynamic programming*. Athena Scientific

Bhasin S, Kamalapurkar R, Johnson M, Vamvoudakis KG, Lewis FL, Dixon WE (2013) A novel actor-critic-identifier architecture for approximate optimal con-

trol of uncertain nonlinear systems. *Automatica* 49(1): 89–92

Bhatnagar S, Sutton RS, Ghavamzadeh M, Lee M (2009) Natural actor critic algorithms. *Automatica* 45(11):2471–2482

Chowdhary GV, Johnson EN (2011) Theory and flight-test validation of a concurrent-learning adaptive controller. *J Guid Control Dyn* 34(2):592–607

Chowdhary G, Yucelen T, Mühlegg M, Johnson EN (2013) Concurrent learning adaptive control of linear systems with exponentially convergent bounds. *Int J Adapt Control Signal Process* 27(4):280–301

Grondman I, Buşoniu L, Lopes GA, Babuška R (2012) A survey of actor-critic reinforcement learning: standard and natural policy gradients. *IEEE Trans Syst Man Cybern Part C Appl Rev* 42(6):1291–1307

Gupta H, Srikant R, Ying L (2019) Finite-time performance bounds and adaptive learning rate selection for two time-scale reinforcement learning. In: *Advances in neural information processing systems*

Hong Y, Wang J, Cheng D (2006) Adaptive finite-time control of nonlinear systems with parametric uncertainty. *IEEE Trans Autom Control* 51(5):858–862

Kamalapurkar R, Walters PS, Rosenfeld JA, Dixon WE (2018) *Reinforcement learning for optimal feedback control: a Lyapunov-based approach*. Springer

Karafyllis I, Krstic M (2018) Adaptive certainty-equivalence control with regulation-triggered finite-time least-squares identification. *IEEE Trans Autom Control* 63(10):3261–3275

Kiumarsi B, Vamvoudakis KG, Modares H, Lewis FL (2017) Optimal and autonomous control using reinforcement learning: a survey. *IEEE Trans Neural Netw Learn Syst* 29(6):2042–2062

Lewis FL, Liu D (2013) *Reinforcement learning and approximate dynamic programming for feedback control*, vol 17. John Wiley & Sons

Modares H, Lewis FL, Naghibi-Sistani M-B (2014) Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems. *Automatica* 50(1):193–202

Parikh A, Kamalapurkar R, Dixon WE (2019) Integral concurrent learning: adaptive control with parameter convergence using finite excitation. *Int J Adapt Control Signal Process* 33(12):1775–1787

Sledge JJ, Emigh MS, Príncipe JC (2018) Guided policy exploration for Markov decision processes using an uncertainty-based value-of-information criterion. *IEEE Trans Neural Netw Learn Syst* 29(6):2080–2098

Sutton RS, Barto AG (1998) *Reinforcement learning: an introduction*. MIT Press, Cambridge, MA

Vrabie D, Vamvoudakis KG, Lewis FL (2013) *Optimal adaptive control and differential games by reinforcement learning principles*. The Institution of Engineering and Technology

Wawrzyński P (2009) Real-time reinforcement learning by sequential actor-critics and experience replay. *Neural Netw* 22(10):1484–1497

Werbos PJ (1992) Approximate dynamic programming for real-time control and neural modeling. In: White DA, Sorce DA (eds) Handbook of intelligent control: neural, fuzzy, and adaptive approaches, vol 15. Norstrand, New York, pp 493–525

Reinforcement Learning for Control Using Value Function Approximation

Konstantinos Gatsis¹ and George J. Pappas²

¹Department of Engineering Science, University of Oxford, Oxford, UK

²Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA

Abstract

This entry provides a short introduction to a class of reinforcement learning algorithms, in particular value function approximation, applied to stochastic optimal control problems. The entry demonstrates how core ideas from dynamic programming and Bellman equations are utilized in common data-driven reinforcement learning algorithms, as well as discuss fundamental challenges of the approach.

Keywords

Optimal control · Stochastic control · Machine learning · Reinforcement learning

Introduction

Sequential decision-making problems under uncertainty are a general class of problems encountered in engineering practice. Stochastic optimal control theory provides a classical way of formalizing such problems. The major outcome of this theory is the development of dynamic programming algorithms which can be used to find optimal solutions. In practice however dynamic programming algorithms have

high computational complexity and can be used primarily in problems with finite and relatively small state and action spaces, excluding higher dimensional and continuous control problems. Moreover, classical dynamic programming is developed based on a known model of the dynamics and the evolution of the system. This assumption is challenging in practice where a system is not completely known a priori or subject to changes and disturbances.

Reinforcement learning is a collection of data-driven algorithms that have been proposed to address the above challenges. Data in this context usually involve trajectories of the system to be controlled over time and associated costs. Given such data, and often without explicit knowledge of the system description, these algorithms aim to find good control policies. Such approaches have gained particular interest in recent years due to the advancements in computing resources as well as the development of efficient machine learning architectures, such as neural networks (LeCun et al. 2015). Examples of the success of the reinforcement learning approach are using neural networks to learn how to play video games from pixels (Mnih et al. 2015) and in the game of Go (Silver et al. 2016), as well as in robotics (Levine et al. 2016). This article focuses on value function approximation algorithms, while policy search methods constitute a separate class of algorithms in the literature (Bertsekas 2015; Sutton and Barto 2018).

Reinforcement Learning for Control Using Value Function Approximation

The approach follows a stochastic optimal control problem formulation. The interest is in controlling a dynamical system of the form

$$x_{k+1} = f(x_k, u_k, w_k), \quad (1)$$

where x_k is the system state taking values in general in (or a subset of) $\mathbb{R}^n$, u_k is the control input taking values in general in (or a subset of) $\mathbb{R}^m$, and w_k is a noise term, typically modeled as an independent and identically distributed random

variable. There is also a given scalar *cost function at each time step* $c(x_k, u_k)$. This is an important part of the problem formulation, as the scalar cost function is the main mechanism for enforcing some desired control behavior; hence it should capture all information relevant to achieving a task.

The goal is to choose the control inputs $u_k, k \geq 0$ as a function of current state, or in other words optimize over the control policy denoted by $u_k = \pi(x_k)$. The criterion for choosing the control policy is the cost accumulated over time. This entry studies an infinite horizon cost discounted by a factor $\gamma \in (0, 1)$ defined as

$$\mathbb{E} \sum_{k=0}^{\infty} \gamma^k c(x_k, u_k), \quad (2)$$

and the approach can be similarly followed for minimizing finite horizon, undiscounted, or average costs (Bertsekas 2015). The standard linear quadratic regulator is a particular example of this problem with a closed form solution. The approach to find the optimal policy described in this entry is based on learning value functions from data, that is, trajectories sampled from the system dynamics. As a first step, we consider the problem of computing the cost associated with following a fixed control policy. This step is called *policy evaluation*. Then a procedure to search for an optimal policy is described.

Policy Evaluation by Value Function Approximation

For any policy π , suboptimal in general, there is a function, called the value function V_π , which expresses the future expected cost starting from a point $x \in \mathbb{R}^n$ and following the policy π , i.e.,

$$V_\pi(x) = \mathbb{E}^\pi \left[\sum_{k=0}^{\infty} \gamma^k c(x_k, u_k) \middle| x_0 = x \right], \quad (3)$$

From standard dynamic programming arguments for such a function, it holds that

$$\begin{aligned} V_\pi(x_k) &= c(x_k, \pi(x_k)) \\ &\quad + \gamma \mathbb{E}_{w_k} [V_\pi(x_{k+1}) \mid x_k, u = \pi(x_k)] \end{aligned} \quad (4)$$

for all $x_k \in \mathbb{R}^n$. This is called the *Bellman equation* for the given control policy π . The Bellman equation is a functional equation, and it is hard to solve for the desired value function V_π even if the dynamics are available. This issue is commonly referred to as the curse of dimensionality.

To alleviate the curse of dimensionality, an alternative approach is to search for an approximate solution to this equation based on data. Specifically, a *family of functions* $V(x; \theta)$ described by a set of parameters θ taking values in some subset of $\mathbb{R}^p$ is commonly assumed, and then the problem is relaxed to just searching for appropriate values for θ . A special case is a linear parametrization $V(x; \theta) = \sum_{i=1}^p \phi_i(x) \theta_i = \phi(x)^T \theta$ where $\phi_i(x)$ are fixed functions, often referred to as features of the state. Examples include the following:

- One simple approach is to discretize the state space and search within the family of functions that are piecewise constant on the discretized space. For example, if $x_k \in [0, 1]$ and we discretize the space evenly in $p = 2^L$ segments, then define $V(x; \theta) = \sum_{i=1}^{2^L} \theta_i \mathbb{I}(x \in [(i-1)2^{-L} \leq x \leq i2^{-L}])$ using an indicator function notation. This is a special case of a linear approximation. Discretization is a general approach but does not scale well for higher-dimensional state spaces (curse of dimensionality).
- A linear combination of Fourier basis functions or radial basis functions may be used as the family of value functions (Sutton and Barto 2018, Ch. 9).
- A linear combination of polynomials of the state x up to some fixed degree may be used for $V(x; \theta)$. As an example, in the case of the linear quadratic control problem, it is well-known that the value functions of linear policies are quadratic functions; hence it suffices to approximate over the family of quadratic value functions.
- A neural network with multiple layers may be used for $V(x; \theta)$ where x is the input and θ is the collection of weights across all layers (Sutton and Barto 2018; Silver et al. 2016).

The data that are used to find the parameters of a value function approximation are typically trajectories of the dynamical system and the cost generated by following the policy that one is interested in evaluating. More specifically they are of the form

$$\{x_0, u_0, c_0, x_1, u_1, c_1, \dots, x_N\}. \quad (5)$$

That is, an initial starting state x_0 is selected, then the control input $u_0 = \pi(x_0)$ is applied to the system following the policy, and the current cost $c_0 = c(x_0, u_0)$ is obtained, as well as the next random state x_1 following the system dynamics (1), and the process repeats till some time N . For simplicity of exposition, this entry discusses the case where a single trajectory is given. In practice one creates a pool of multiple trajectories from potentially different initial points and processes parts of these collection of trajectories at random. The procedure that generates the data/trajectories can be, for example, a simulation of the system dynamics assuming that it is available, or a physical experiment. It is also worth noting that in this setup, the cost c_k per time step could also be measured directly (in simulation or experiment), rather than computed from a known cost function.

With the collected data points following the policy, one needs to find the parameters θ of the value function family that better fit the Bellman equation. A simple criterion for this is to compute the parameters that minimize the average square error in Bellman equation (4) as in

$$\text{minimize}_{\theta} \sum_{k=1}^{N-1} (V(x_k; \theta) - c_k - \gamma V(x_{k+1}; \theta))^2. \quad (6)$$

Standard gradient descent can be used to find a local optimum of this criterion, or variants such as stochastic gradient descent where only few samples are selected at random from the pool of data to approximate the gradient. Regularization terms on θ may also be added in (6) to avoid overfitting to data.

From a computation point of view, it is useful to note that in the special case of approximating

the value function as a linear combination of basis functions, the above process simplifies to a (convex) quadratic optimization problem

$$\text{minimize}_{\theta} \sum_{k=1}^{N-1} \left(\phi(x_k)^T \theta - c_k - \gamma \phi(x_{k+1})^T \theta \right)^2. \quad (7)$$

The optimal solution θ satisfies the p -dimensional linear system of equations

$$\begin{aligned} & \sum_{k=0}^{N-1} (\phi(x_k) - \gamma \phi(x_{k+1})) (\phi(x_k) - \gamma \phi(x_{k+1}))^T \theta \\ &= \sum_{k=0}^{N-1} c_k (\phi(x_k) - \gamma \phi(x_{k+1})), \end{aligned} \quad (8)$$

and the solution is unique under standard rank conditions and can be readily computed either by directly solving the above equation or by gradient descent.

It is intuitively clear that in order to obtain a better fit to the Bellman equation, one needs to use rich enough samples where x_k spans a large area of the state space. However samples are generated following a given policy; hence fewer data will be generated at parts of the state space that are visited less often by the policy. This is an exploration problem. A typical mechanism to alleviate this problem is to pick a policy π that includes some randomness to promote exploration of the state space. For example, the policy may include an additive i.i.d. noise. At a high level, this is related to the notions of persistently exciting inputs in adaptive control and system identification (Kumar and Varaiya 2015).

Beyond the simple square Bellman error in (6), other criteria may be used to learn the parameters θ . A more commonly used approach for linear value function approximations is to solve what is termed *projected Bellman equation* – see, for example, (Bertsekas 2015, Ch. 6) and (Sutton and Barto 2018, Ch. 11). Moreover the above discussion is for an offline setup where a batch of data/trajectories are available to process, while alternatively online algorithms have been proposed where iteratively at each step, only

the currently observed data sample is processed in order to find the appropriate value function approximation (Sutton et al. 2009).

Policy Optimization by Value Function Approximation

Building upon the previous section, one can devise a procedure to approximately search for optimal policies. There is one extra technical step that needs to be added. Beyond the value function that depends on each state, it is also possible to define a function that depends on both the state and the control input and is commonly referred to as the Q-function. Specifically for any control policy π , there exists a function $Q_\pi : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ such that

$$Q_\pi(x_k, u_k) = c(x_k, u_k) + \gamma \mathbb{E}_{w_k} [Q(x_{k+1}, \pi(x_{k+1})) | x_k, u_k] \quad (9)$$

holds for all $x_k \in \mathbb{R}^n$ and all $u_k \in \mathbb{R}^m$. This is again a form of Bellman equation associated with the policy π . The approach to approximate the Q-function from data is the same as in the previous section. The first step is to collect trajectories/samples of the form

$$\{x_0, u_0, c_0, x_1, u_1, c_1, \dots, x_N\}, \quad (10)$$

following policy π and then to fit a value function on this data selected from some family of functions $Q(x, u; \theta)$ parametrized by θ . The same type of families can be employed as explained in the previous section. The main challenge is that the function that is being parametrized in this case is in general more complex as it is defined on the higher-dimensional domain of both states and inputs. To fit the parameters θ , a common approach is to similarly minimize a square Bellman error with respect to Q-functions, e.g.,

$$\begin{aligned} & \text{minimize}_{\theta} \sum_{k=1}^{N-1} (Q(x_k, u_k; \theta) - c_k \\ & - \gamma Q(x_{k+1}, u_{k+1}; \theta))^2. \end{aligned} \quad (11)$$

The main advantage of working with the Q-function of a policy π is that it becomes straightforward to obtain a new improved policy that has a lower cost. This is an iterative procedure called *policy iteration* described in Algorithm 1. In principle, assuming no approximation errors, or no randomness due to the data samples, the procedure is guaranteed to provide an improved policy at each iteration, and at least in finite state and action spaces converge to the optimal policy (Bertsekas 2015, Ch. 2). In the face of bias from the function approximation procedure and randomness from the sampled data, there are in general no guarantees of improvement or convergence, but one expects that better policies will be obtained.

Algorithm 1 Policy iteration algorithm with Q-functions

- 1: **for** Iteration $t = 1, \dots$ **do**
- 2: Collect sample trajectories $\{x_0, u_0, c_0, x_1, u_1, c_1, \dots, x_N\}$, following policy π_t
- 3: Learn an approximate Q-function Q_{π_t} of current policy from data samples, e.g., using (11)
- 4: Update policy to

$$\pi_{t+1}(x) = \operatorname{argmin}_u Q_{\pi_t}(x, u) \quad (12)$$

5: **end for**

We point out that the exploration problem discussed in the previous section is also present when trying to learn Q-functions and is even further complicated by the fact that data need to be collected from many input points $u \in \mathbb{R}^m$. Moreover, as the above algorithm shows, the standard policy iteration algorithm states that one should update the policy to a new one selected as the minimum of the approximated Q-function pointwise at each state x , and hence the new policy is deterministic. This in turn limits exploration of the state and input spaces at the next iteration. A common approach to alleviate this situation is to update to a randomized policy instead. For example a common option called ϵ -greedy is a new policy which with high probability $1 - \epsilon$ selects the minimum of the Q-function, and with

some small probability ϵ , it randomizes over possible inputs in order to explore (Sutton and Barto 2018, Ch. 2).

Finally, it is worth pointing out that performing the pointwise minimization step to obtain a new policy may be computationally hard. This is often alleviated by approximating also the policy through a different set of parameters, separate from the value function approximation parameters. This gives rise to a class of reinforcement learning algorithms called actor-critic algorithms (Sutton and Barto 2018, Ch. 13).

Summary and Future Directions

This entry describes main algorithmic approaches for solving optimal control problems using reinforcement learning and in particular value function approximation approaches. Key challenges and limitations of the approach are also discussed. Extended discussions can be found in recent books (Bertsekas 2015; Sutton and Barto 2018) and reviews (Lewis and Vrabie 2009; Buşoniu et al. 2018).

A general drawback of reinforcement learning methods, including the ones relying on value function approximation in this entry, is that they require a large number of samples, which can limit their practical use. Current efforts are focused on analyzing this sample complexity (Tu and Recht 2018; Dean et al. 2018). Moreover, safety is an important challenge in reinforcement learning approaches. Typically as the approach depends on randomly obtained data and on fitting value functions from a potentially large class of functions, the overall process lacks safety guarantees. Safety and stability properties exploiting prior system knowledge are a current research topic (Fisac et al. 2019; Berkenkamp et al. 2017; Rosolia and Borrelli 2017).

Cross-References

- [Model-Free Reinforcement Learning-Based Control for Continuous-Time Systems](#)
- [Multiagent Reinforcement Learning](#)

- [Optimal Control and the Dynamic Programming Principle](#)
- [Stochastic Dynamic Programming](#)

Recommended Reading

Bertsekas (2015) and Sutton and Barto (2018)

Bibliography

- Berkenkamp F, Turchetta M, Schoellig A, Krause A (2017) Safe model-based reinforcement learning with stability guarantees. In: *Advances in neural information processing systems*, pp 908–918
- Bertsekas DP (2015) *Dynamic programming and optimal control*, vol II, 4th edn. Athena Scientific, Nashua
- Buşoniu L, de Bruin T, Tolić D, Kober J, Palunko I (2018) Reinforcement learning for control: performance, stability, and deep approximators. *Ann Rev Control* 46:8–28
- Dean S, Mania H, Matni N, Recht B, Tu S (2018) Regret bounds for robust adaptive control of the linear quadratic regulator. In: *Advances in neural information processing systems*, pp 4188–4197
- Fisac JF, Akametalu AK, Zeilinger MN, Kaynama S, Gillula J, Tomlin CJ (2019) A general safety framework for learning-based control in uncertain robotic systems. *IEEE Trans Autom Control* 64(7):2737–2752
- Kumar PR, Varaiya P (2015) *Stochastic systems: estimation, identification, and adaptive control*, vol 75. Society for Industrial and Applied Mathematics, Philadelphia
- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436
- Levine S, Finn C, Darrell T, Abbeel P (2016) End-to-end training of deep visuomotor policies. *J Mach Learn Res* 17(1):1334–1373
- Lewis FL, Vrabie D (2009) Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits Syst Mag* 9(3):32–50
- Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540):529
- Rosolia U, Borrelli F (2017) Learning model predictive control for iterative tasks. a data-driven control framework. *IEEE Trans Autom Control* 63(7):1883–1896
- Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M et al (2016) Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484
- Sutton RS, Barto AG (2018) *Reinforcement learning: an introduction*. MIT Press, Cambridge

- Sutton RS, Maei HR, Precup D, Bhatnagar S, Silver D, Szepesvári C, Wiewiora E (2009) Fast gradient-descent methods for temporal-difference learning with linear function approximation. In: Proceedings of the 26th annual international conference on machine learning. ACM, pp 993–1000
- Tu S, Recht B (2018) Least-squares temporal difference learning for the linear quadratic regulator. In: International conference on machine learning, pp 5012–5021

Reverse Engineering and Feedback Control of Gene Networks

Mario di Bernardo^{1,2} and Diego di Bernardo^{3,4}

¹University of Naples Federico II, Naples, Italy

²Department of Electrical Engineering and ICT, University of Naples Federico II, Napoli, Italy

³Telethon Institute of Genetics and Medicine, Pozzuoli, Italy

⁴Department of Chemical, Materials and Industrial Engineering, University of Naples Federico II, Napoli, Italy

from other areas of engineering and applied science such as network reconstruction in packet data networks and network control systems. Obviously, biological applications pose unique challenges that are still open problems such as the level of noise, the often global effects of local perturbations to the network, and the curse of dimensionality which poses several problems in terms of size of the required data sets and computational complexity of the reverse engineering algorithms. In synthetic biology, feedback control strategies can be used to render the network behavior more reliable and less sensitive to unwanted noise and perturbations. Here, after reviewing some of the key approaches for network reconstruction, we highlight the different types of control approaches that can be used to control a gene regulatory network.

Keywords

Gene networks · Reverse engineering · Feedback control · Synthetic biology

Abstract

Over the past decades, increasingly larger and more complex gene regulatory networks (GRNs) have been synthesized de novo in synthetic biology or analyzed and studied in a number of different biological contexts. A crucial problem often encountered in applications is to reconstruct the structure of the regulatory interactions between genes and proteins in the network. This entails using experimental data to infer or reverse engineer the network structure by carrying out a combination of data processing and statistical analysis, model construction, and validation by comparing in silico predictions with experiments. The goal is to establish an iterative procedure so that models can be increasingly refined as experimental data becomes available. The problem of reverse engineering and modelling gene networks is an area of active research in systems and control overlapping with similar problems

Introduction

Synthetic biology is a new field of research where methods and tools from engineering are used to synthesize novel biological devices in vivo. At the core of any engineering design lies the ability of interconnecting elementary components to assemble circuits able to perform complex functions. Think of electronics where the interconnection of elementary components such as the resistors, capacitors, transistors, and inductances can be used to create logic gates, amplifiers, and other devices. In biology, gene regulatory networks that combine activation and inhibition among genes can be exploited to endow new functions to a living cell. Among the first networks to be proposed, the toggle switch and the repressilator represent cornerstones in the ability of synthetic biologists to do so as they endow a cell with the ability to store information or to oscillate.

Nature exploits gene regulations in many ways. Whether they are engineered or occur

in nature, gene regulatory networks can have complex structures, and it is essential to investigate methods to reverse engineer their structures from data. In particular, understanding the set of regulatory interactions (and their directions) in a network is essential to carry out fundamental research in systems biology and to design reliable GRNs in synthetic biology. Also, to enhance robustness of GRNs performing a desired function and compensate uncertainties and unmodelled dynamics, it is useful to construct feedback control mechanisms stabilizing their behavior in the presence of perturbations and noise.

Reverse engineering deals with these problems and in particular with the process of network inference from experimental data. As highlighted in Delgado and Gomez-Vela (2018), such an inference process relies on a Knowledge Database Discovery (KDD) workflow which involves going from data analysis to the validation of the reconstructed network models by contrasting *in silico* results with *in vivo* experimental data sets.

In this entry, we give a brief overview of the problems of reverse engineering and controlling gene regulatory networks. We discuss strategies for reconstructing the structure of a regulatory network of interest from data and give a summary of the key results concerning their control highlighting the different approaches that can be used to tackle this pressing open problem.

Reverse Engineering Methods

Reverse engineering GRNs from data involve a number of different goals and methods. Typically the key objectives of reverse engineering are (i) to reconstruct the (directed) network of interactions between genes and molecules for a GRN of interest, (ii) to derive mathematical models capturing the network behavior *in silico*, and (iii) to estimate the biochemical parameters characterizing the nature and strength of the interaction. Rather than giving an exhaustive review, we focus here on some of the key approaches used to infer the structure of a GRN from data

and derive a mathematical model able to approximate and capture the dynamics of the real network. We refer the reader for more details to the many exhaustive review papers, e.g., (Le Novère 2015; Klinger and Bluethgen 2018; Delgado and Gomez-Vela 2018; De Smet and Marchal 2010; He et al. 2009; Bansal et al. 2006, 2007).

The goal of network inference methods in computational biology is that of using data to unfold the structure of the gene regulatory network of interest, or to check in synthetic biological applications that the engineered gene interactions *in vivo* correspond to those designed and tested *in silico*. Typically, information on the network structure is obtained from gene expression data, although information can also be obtained from other omics data sets (e.g., proteomics, metabolomics, etc.).

There are currently four different approaches that can be used to reverse engineer the network from data. They can either be used alone or, more effectively, be combined to provide a more reliable reconstruction of the gene interactions in the network. The four main approaches are those based on correlation, information theoretic tools, Bayesian inference, or the use of dynamical systems and differential equations. Each is encoded in a variety of different software packages that are well described in Le Novère (2015).

Correlation and more generally statistical methods such as regression can shed light on whether two genes in the network share some regulatory features, for example, they are both up- or downregulated. Typically, these methods return undirected networks that, as noted in Le Novère (2015) are not entirely suitable to derive mathematical models. (Some methods to assign directions to links can be found in the literature, e.g., Gitter et al. 2011).

An alternative approach is to use information theoretic tools such as transfer entropy, mutual information, and their more recent variations such as causation entropy (Sun et al. 2015) that can be effectively used to reduce the number of spurious links in the reconstruction. Such methods often require steady-state expression data enabling the discovery of large GRN and are computationally less expensive than

other approaches. An example of such methods is ARACNE (Algorithm for the Reverse engineering of Accurate Cellular Network) (Margolin et al. 2006) to cite one of the earliest examples. Typically, the reconstructed networks are undirected, although some methods exist to obtain directed interactions (Feizi et al. 2013).

To overcome this problem, Bayesian inference can be used to give directed gene regulation edges in the reconstruction and decide the most likely network structure fitting the data.

Finally, methods based on dynamical systems, e.g., Gardner et al. (2003) and Tegner et al. (2003), can be used when dynamic measurements are available and the goal is to obtain information on the dynamic interactions between genes and proteins. One limitation of many of these latter approaches is the use of linear models that, while allowing inferring interactions from data, are far from capturing the complexity of the nonlinear dynamics describing the network of interest; see, for example, our own work on capturing the structure and dynamics of a synthetic network of five genes in yeast (Cantone et al. 2009). Modular response analysis (MRA) is often used in this context, whereas steady-state data is collected after node-wise perturbations of the network dynamics. The problem as highlighted in Kang et al. (2015) is often that point-wise perturbations of a biological network produce global effects that render it difficult to infer the correct interaction links among genes. Several approaches have been presented to overcome this problem or at least to limit its potential pitfalls; see, for example, the procedure proposed in Kang et al. (2015) or, more recently, in Klingler and Bluethgen (2018).

As is usual in identification and reconstruction strategies in Engineering, each of the methodologies described above has its pros and cons (Delgado and Gomez-Vela 2018). The performances of the various algorithms can be better uncovered by using common benchmarks, or golden standards, to compare their performance, for example, via the results of the DREAM challenge for network reconstruction as described in Marbach et al. (2012). Often a combination of

different reverse engineering methods is the best way forward to obtain the network structure with a higher degree of confidence.

Once the network structure has been obtained, it is possible to construct qualitative and quantitative models of the network dynamics by using mathematical modelling techniques. One of the most widely used approaches, which is also the best suited for control system design, is the use of linear or nonlinear differential equations capturing the nature of the regulatory links between genes and proteins in the network (Chen et al. 2010). Once model equations have been obtained, the next step is to estimate the model parameters from data. This can be done by using identification techniques from control theory, optimization tools, and a combination of data mining from the existing literature. Usually, it is necessary to carry out ad hoc experiments and use nonlinear optimization tools such as genetic algorithms in order to solve the problem of identifying a typically large number of parameters given the available data sets. A combination of steady state and time course experiments is often required to identify the unknown parameters in the model. The process is often cumbersome requiring a cycle of multiple iterations involving numerical simulations, data analysis, optimization, further in vivo experiments, and so on. An example highlighting the problems involved is described in Marucci et al. (2011), which details the mathematical model derivation and parameterization for the five gene regulatory network in yeast mentioned earlier in this entry. An excellent overview comparing different approaches and highlighting their strengths and weaknesses can be found in Delgado and Gomez-Vela (2018).

It is worth mentioning here that recent advances in machine learning have brought forward a new set of approaches based on the use of AI and neural models. These methods though require a large amount of data and are therefore of limited application when experimental data are cumbersome or expensive to acquire. Also, as noted in Delgado and Gomez-Vela (2018), such experiments are hard to carry out because every situation requires a different learning rate definition. Nevertheless, they allow both linear

and dynamic behavior and are suitable for both steady-state and time course expression profiles.

Controlling Gene Networks

Reverse engineering methods can yield models affected by uncertainties and unmodelled dynamics. For instance, linear models can be derived approximating the nonlinear dynamics of the network. Feedback control strategies can be used to compensate this effect and endow the network with some desired function. Control strategies for cell populations endowed with GRNs can be classified as external, internal, or multicellular (Del Vecchio et al. 2016; Del Vecchio et al. 2018).

External control is typically achieved via microfluidics and optogenetics platforms where the host cells are controlled by closing the feedback loop through some “external” devices such as a fluorescence microscope to measure the output of interest via segmentation algorithms, a PC implementing the control action, and actuated syringes or lasers providing the control input to the cells. Successful attempts to control gene expression, or even signaling pathways, via external strategies have been described in the literature (Miliadis-Argeitis et al. 2011; Toettcher et al. 2011), (Uhlendorf et al. 2012; Menolascina et al. 2014; Fiore et al. 2015, 2016; Fracassi et al. 2015). They mainly differ in the control input (osmotic pressure, light, small molecules) and the control strategy adopted. Optogenetics-based light inducible systems have been exploited to control gene expression in yeasts to regulate intracellular signaling dynamics in mammalian cells and to drive protein levels by using light-switchable two-component systems in bacteria. Microfluidic-based devices, allowing a tight control of cellular growing medium and the administration of inducer small molecules, have been successfully employed to investigate synchronization properties of synthetic biological clocks in bacterial cells (Danino et al. 2010), to control the transcription from the HOG1 promoter in yeast *S. cerevisiae* by varying the osmotic pressure (Uhlendorf et al. 2012), and, in our own work, to control

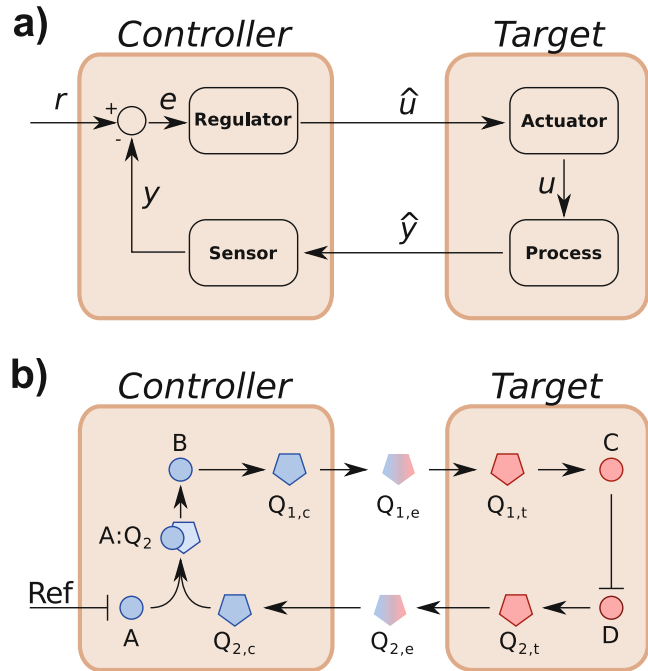
transcription from the GAL1 promoter using galactose and glucose as input (Menolascina et al. 2014). In this context control strategies that have already been described in the literature include the Proportional-Integral (PI) control and the Model Predictive Control (MPC). Also a different control strategy named Zero Average Dynamics (ZAD), an approach inspired by sliding control techniques, was also applied. A comparison between the performance of these strategies to control expression of the GAL1 promoter in yeast can be found in Fiore et al. (2015).

Internal (or embedded) control aims instead to embed all functions required to achieve the control goal within the cell membrane. The idea is to provide cells with some reference signal, encoded by an inducer molecule, for example, or an external optogenetic input, and let GRNs within the cell implement the sensing, actuation, and computation functions required to close the feedback loop. While many theoretical studies are available showing that such an approach is feasible in silico, the validation in vivo of such a technique has been somewhat limited. This is probably due to the complexity of the regulatory networks required to implement the various control functions and the subsequent metabolic burden that can be lethal for the cell. (Here by metabolic burden we mean as in Wu et al. (2016) the proportion of the resources of a host cell that are used to construct and operate engineered pathways.) A recent example of internal control that has been validated in vivo is the “antithetic feedback control” strategy recently presented in Aoki et al. (2019).

To reduce metabolic burden to the cells and enhance modularity of the control design, an alternative approach is to achieve control by assigning the different control functions to different cell populations within a microbial consortium. The idea presented in Fiore et al. (2016) is to create a consortium where one population, the “controllers,” senses and controls the output of another population, the “targets,” so that actuation, sensing, and computation are split among them; see Fig. 1. In this way, the GRNs to be embedded in each of the two types of cells become simpler, and metabolic burden

Reverse Engineering and Feedback Control of Gene Networks,

Fig. 1 Main idea (a) and schematic biological implementation (b) of multicellular control: the sensing, actuation, and computation functions are distributed across two different cell populations, the controllers and the targets communicating with each other via orthogonal quorum sensing channels. (Reprinted with permission from (Fiore et al. 2016). Copyright 2016 American Chemical Society.)



to each single cell is reduced. Also, flexibility and modularity of the design are enhanced as it is possible to re-use controller cells to tame the dynamics of a different cell population simply by replacing the targets with another population of interest. A crucial challenge in order to establish the control loop is that the two populations need to communicate with each other so that the control input, $\hat{u}$, can be sent from the controllers to the targets and the process output, $\hat{y}$, can be fed back from the targets to the controllers. Also the controllers must be able to sense the target outputs, compare it with some reference signal of interest, and vary their output accordingly.

Multicellular control strategies can be employed to enable cellular functions over space, for example, to induce spatial patterns for coordinated self-organization of cells, thus mimicking developmental processes (Arcak 2012; Del Vecchio et al. 2016). Control strategies across multiple populations have been utilized to control cell growth and maintain populations' size at a desired equilibrium (Scott et al. 2017). For example, in You et al. (2004), a population control circuit is proposed in silico to autonomously maintain the density of *E. coli* at a

desired level, a goal also achieved more recently in Ren et al. (2017) where feedback control across two populations was used for controlling the population size of a microbial culture through the production of toxins and anti-toxins between the populations. While having been tested in a number of scenarios in silico, multicellular control has not been fully implemented in vivo yet. Also, in silico validation often relies on the use of aggregate deterministic models or agent-based simulations. Hence, the full experimental implementation and validation of a multicellular control strategy remains a crucial open problem as well as its modelling via approaches that can fully take into account the unavoidable stochastic and spatial distribution effects it entails.

Summary and Future Directions

The increasing successes of computational biology will make computational tools such as reverse engineering methods and control strategies more and more likely to be used in practical applications. As highlighted above, a key open problem is the curse of dimensionality

that affects all the proposed methods. As the complexity of the networks of interest increases, the computational complexity of the reconstruction methods and the requirement for larger and larger data sets become a pressing open problem. Also, better methods to obtain directed regulatory links are needed and to avoid the presence of spurious links due to global effects of modular response analysis or other “local” perturbation techniques. From the control viewpoint, some crucial open challenges remain, for instance, the in vivo validation of all the proposed implementations (external, internal, or multicellular) of a feedback control strategy to regulate the behavior of a GRN of interest via appropriate inducible promoters.

Cross-References

- [Deterministic Description of Biochemical Networks](#)
- [Identification and Control of Cell Populations](#)
- [Synthetic Biology](#)
- [System Identification: An Overview](#)

Bibliography

- Aoki SK, Lillacci G, Gupta A, Baumschlager A, Schwein-gruber D, Khammash M (2019) A universal biomolecular integral feedback controller for robust perfect adaptation. *Nature* 570:533–537
- Arcak M (2012) Pattern formation by lateral inhibition in large-scale networks of cells. *IEEE Trans Autom Control* 58:1250–1262
- Bansal M, Gatta GD, Di Bernardo D (2006) Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics* 22:815–822
- Bansal M, Belcastro V, Ambesi-Impiombato A, Di Bernardo D (2007) How to infer gene networks from expression profiles. *Mol Syst Biol* 3:1
- Cantone I et al (2009) A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. *Cell* 137:172–181
- Chen L, Wang R, Li C, Aihara K (2010) Modeling biomolecular networks in cells: structures and dynamics. Springer, London
- Danino T, Mondragon-Palomino O, Tsimring L, Hasty J (2010) A synchronized quorum of genetic clocks. *Nature* 463:326
- Delgado FM, Gomez-Vela F (2018) Computational methods for gene regulatory networks reconstruction and analysis: a review. *Artif Intell Med*. <https://doi.org/10.1016/j.artmed.2018.10.006>
- Del Vecchio D et al (2016) Control theory meets synthetic biology. *J R Soc Interface* 13:20160380
- Del Vecchio D et al (2018) Future systems and control research in synthetic biology. *Ann Rev Control* 45: 5–17
- De Smet R, Marchal K (2010) Advantages and limitations of current network inference methods. *Nat Rev Microbiol* 8:717
- Feizi S, Marbach D, Médard M, Kellis M (2013) Network deconvolution as a general method to distinguish direct dependencies in networks. *Nat Biotechnol* 31:726–733
- Fiore G, Perrino G, Di Bernardo M, Di Bernardo D (2015) In vivo real-time control of gene expression: a comparative analysis of feedback control strategies in yeast. *ACS Synth Biol* 5:154–162
- Fiore G et al (2016) In-silico analysis and implementation of a multicellular feedback control strategy in a synthetic bacterial consortium. *ACS Syn Biol* 6: 507–517
- Fracassi C, Postiglione L, Fiore G, Di Bernardo D (2015) Automatic control of gene expression in mammalian cells. *ACS Synth Biol* 5:296–302
- Gardner TS, di Bernardo D, Lorenz D, Collins JJ (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* 301(5629):102–105
- Gitter A, Klein-Seetharaman J, Gupta A, Bar-Joseph Z (2011) Discovering pathways by orienting edges in protein interaction networks. *Nucleic Acids Res* 39:e22
- He F, Balling R, Zeng AP (2009) Reverse engineering and verification of gene networks: principles, assumptions, and limitations of present methods and future perspectives. *J Biotechnol* 144:190–203
- Hurley DG et al (2015) NAIL, a software toolset for inferring, analyzing and visualizing regulatory networks. *Bioinformatics* 31:277–278
- Kang T et al (2015) Discriminating direct and indirect connectivities in biological networks. *Proc Natl Acad Sci U S A* 112:12893–12898
- Klinger B, Bluethgen N (2018) Reverse engineering gene regulatory networks by modular response analysis – a benchmark. *Essays Biochem* 62:535–547
- Le Novère N (2015) Quantitative and logic modelling of molecular and gene networks. *Nat Rev Gen* 16:146
- Marbach D et al (2012) Wisdom of crowds for robust gene network inference. *Nat Methods* 9:796–804
- Margolin AA et al (2006) ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinform* 7:S7
- Marucci L, Santini S, Di Bernardo M, Di Bernardo D (2011) Derivation, identification and validation of a computational model of a novel synthetic regulatory network in yeast. *J Math Biol* 62(5):685–706
- Menolascina F et al (2014) In-vivo real-time control of protein expression from endogenous and synthetic gene networks. *PLoS Comput Biol* 10:e1003625

- Miliars-Argeitis A et al (2011) In silico feedback for in vivo regulation of a gene expression circuit. *Nat Biotechnol* 29:1114
- Ren X, Baetica AA, Swaminathan A, Murray RM (2017). Population regulation in microbial consortia using dual feedback control. In: 2017 IEEE 56th annual conference on decision and control (CDC), pp 5341–5347
- Scott SR et al (2017) A stabilized microbial ecosystem of self-limiting bacteria using synthetic quorum-regulated lysis. *Nat Microbiol* 2:17083
- Sun J, Taylor D, Boltt EM (2015) Causal network inference by optimal causation entropy. *SIAM J Appl Dyn Syst* 14(1):73–106
- Tegner J, Yeung MKS, Hasty J, Collins JJ (2003) Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling. *Proc Natl Acad Sci U S A* 100:5944–5949
- Toettcher JE, Gong D, Lim WA, Weiner OD (2011) Light-based feedback for controlling intracellular signaling dynamics. *Nat Methods* 8:837
- Uhlendorf J et al (2012) Long-term model predictive control of gene expression at the population and single-cell levels. *Proc Natl Acad Sci* 109:14271–14276
- Wu G, Yan Q, Jones JA, Tang YJ, Fong SS, Koffas MA (2016) Metabolic burden: cornerstones in synthetic biology and metabolic engineering applications. *Trends Biotechnol* 34:652–664
- You L, Cox III RS, Weiss R, Arnold FH (2004) Programmed population control by cell–cell communication and regulated killing. *Nature* 428:868

Risk-Sensitive Stochastic Control

Hideo Nagai

Department of Mathematics, Kansai University,
Osaka, Japan

Abstract

Motivated by understanding “robustness” from the viewpoints of stochastic control, the studies of risk-sensitive control have been developed. The idea was applied to portfolio optimization problems in mathematical finance, from which new kinds of problem on stochastic control, named “large deviation control,” have been brought and currently the studies are in progress.

Keywords

Robustness · Mathematical finance · Large deviation control

Risk-Sensitive Criterion

Risk-sensitive stochastic control has the criterion

$$J(x, 0; T; \gamma) = \frac{1}{\gamma} \log E \left[e^{\gamma \left\{ \int_0^T f(X_s, u_s) ds + \varphi(X_T) \right\}} \right] \quad (1)$$

with $\gamma \neq 0$, where X_t is the state variable process defined by the controlled stochastic differential equation

$$dX_t = \sigma(X_t)dB_t + b(X_t, u_t)dt, \quad X_0 = x \quad (2)$$

with the control parameter process u_t . Here $\sigma(x) : R^N \mapsto R^N \otimes R^d$ and $b(x, u) : R^N \times R^m \mapsto R^N$. When $\gamma \rightarrow 0$, the criterion behaves as

$$J(x, 0; T; \gamma) \sim E \left[\int_0^T f(X_s, u_s) ds + \varphi(X_T) \right] + \frac{\gamma}{2} \text{Var} \left[\int_0^T f(X_s, u_s) ds + \varphi(X_T) \right] + O(\gamma^2).$$

Then, minimizing the criterion with $\gamma > 0$ is considered to be risk averse, while with $\gamma < 0$, it is to be risk seeking. The problem minimizing the classical criterion $E[\int_0^T f(X_s, u_s) ds + \varphi(X_T)]$ corresponds to the case of $\gamma = 0$, which is risk neutral.

When $f(x, u) = \frac{1}{2}x^*Qx + \frac{1}{2}u^*Su$, $\varphi(x) = \frac{1}{2}x^*Ux$, and $b(x, u) = Ax + Cu$, $\sigma(x) \equiv \Sigma$ with constant matrices Q, S, U, A, C, Σ , minimizing the criterion subject to the state variable processes X_t is called a linear-exponential-quadratic Gaussian (LEQG) control problem, where one may assume Q and U to be nonnegative definite and S positive definite.

H-J-B Equations

The Hamilton-Jacobi-Bellman (H-J-B) equation for the problem minimizing criterion J defined by (1) among the controlled processes governed by (2) is seen to be

$$\begin{cases} \frac{\partial v}{\partial t} + \frac{1}{2} \text{tr}[a(x)D^2v] + H(x, \nabla v) = 0 \\ v(T, x) = \varphi(x), \end{cases} \quad (3)$$

where $a(x) := (a^{ij}(x)) = ((\sigma\sigma^*)^{ij}(x))$ and

$$H(x, p) = \frac{\gamma}{2} p^* a(x) p + \inf_u \{b(x, u)^* p + f(x, u)\}.$$

In an LEQG case, where we assume that Q , U are nonnegative definite and S positive definite, the H-J-B equation has the solution expressed as:

$$v(t, x) = \frac{1}{2} x^* P(t)x + G(t),$$

by using the solutions $G(t)$ of ordinary differential equation

$$\dot{G}(t) + \frac{1}{2} \text{tr}[P \Sigma \Sigma^*] = 0, \quad G(T) = 0$$

and $P(t)$ of the Riccati equation

$$\begin{aligned} \dot{P}(t) + PA + A^*P - P(CS^{-1}C^* - \gamma \Sigma \Sigma^*)P \\ + Q = 0 \end{aligned}$$

with the terminal condition $P(T) = U$, provided that it has a nonnegative definite solution $P(t)$. However, it may occur that the Riccati equation does not have any solution if γ is large. In that case, we say that the risk-sensitive control problem “breaks down.” Namely, there is no control which makes the criterion have a finite value. On the other hand, if it has a solution, then the optimal feedback control is seen to be $-S^{-1}C^*P(t)x$, and the optimal diffusion process turns out to be the solution to

$$dX_t = \Sigma dB_t + (AX_t - CS^{-1}C^*X_t)dt, \quad X_0 = x.$$

The situation can extend to certain general cases. Under sufficiently general conditions, one can

say that if H-J-B equation (3) has a solution, then no “breakdown” occurs in the corresponding risk-sensitive stochastic control problem, and thus relevant studies on existence and uniqueness of the solution to (3) have been done (cf. Nagai 1996, Bensoussan et al. 1998, and Bensoussan and Nagai 2000).

The LEQG problems were first investigated in Jacobson (1973), and then a theory of the LEQG control with complete or partial state information is developed in Whittle (1981) and Bensoussan and Schuppen (1985). Development of the studies of nonlinear risk-sensitive control can be seen in Bensoussan et al. (1998), Nagai (1996), Fleming and McEneaney (1995) etc.

Singular Limits and H^∞ Control

The large deviation theory of Freidlin-Wentzell applies to the risk-sensitive control problem with the criterion

$$\begin{aligned} J_\epsilon(x, 0; T) \\ = \frac{\epsilon}{\theta} \log E[e^{\frac{\theta}{\epsilon} \int_0^T \{\frac{1}{2} u_s^* S(X_s) u_s + V(X_s)\} ds}], \end{aligned} \quad (4)$$

$\epsilon > 0$, and the controlled dynamics

$$dX_t = \sqrt{\epsilon} \sigma(X_t) dB_t + \{b(X_t) + C(X_t)u_t\}dt. \quad (5)$$

The corresponding H-J-B equation is

$$\begin{cases} \frac{\partial v_\epsilon}{\partial t} + \frac{\epsilon}{2} \text{tr}[a(x)D^2v_\epsilon] + H_0(x, \nabla v_\epsilon) + V = 0 \\ v_\epsilon(T, x) = 0, \end{cases} \quad (6)$$

$$\begin{aligned} H_0(x, p) &= \frac{\theta}{2} p^* a(x) p + b(x)^* p \\ &\quad + \inf_{u \in R^m} \{u^* C(x) p + \frac{1}{2} u^* S(x) u\} \\ &= b(x)^* p - \frac{1}{2} p^* \left\{ C S^{-1} C(x)^* \right. \\ &\quad \left. - \theta a(x) \right\} p. \end{aligned}$$

By employing viscosity solution theory, we can see the solution v_ϵ of (6) converges to the viscosity solution v_0 of the equation

$$\begin{cases} \frac{\partial v_0}{\partial t} + H_0(x, \nabla v_0) + V = 0 \\ v_0(T, x) = 0. \end{cases} \quad (7)$$

when sending $\epsilon \rightarrow 0$. Noting that $H_0(x, p)$ can be regarded as

$$H_0(x, p) = b(x)^* p + \sup_z \{z^* p - \frac{1}{2\theta} z^* a(x)^{-1} z\} \\ + \inf_u \{u^* C(x) p + \frac{1}{2} u^* S(x) u\},$$

Eq. (7) is seen to be an H-J-Isaacs equation. Further, this equation has a unique viscosity solution under suitable conditions (Bensoussan and Nagai 1997), and it characterizes the lower value of the differential game with the criterion

$$I(0, T; z, u(z)) = \int_0^T \Psi(x_s, z_s, u(z)_s) ds, \\ \Psi(x, z, u) := -\frac{1}{2\theta} z^* a(x)^{-1} z \\ + \frac{1}{2} u^* S(x) u + V(x)$$

and the controlled dynamics

$$dx_s = \{b(x_s) + z_s + C(x_s)u(z)_s\}ds, \quad x_0 = x.$$

Here the sets $\mathcal{Z}_T$ and Γ_U of admissible strategies for z_s and $u(z)_s$ are, respectively, defined as follows. Let $\mathcal{Z}_T$ and $\mathcal{U}_T$ be, respectively, the totality of a measurable, R^N -valued (resp. R^m -valued) function on $[0, T]$ such that $\int_0^T |z_s|^2 ds < \infty$ (resp. $\int_0^T |u_s|^2 ds < \infty$). Then, for each $z \in \mathcal{Z}_T$ let $u(z)$ be a map defined on $\mathcal{Z}_T$ with its value on $\mathcal{U}_T$ such that whenever for each $0 \leq \tau \leq T$ and $z^{(1)}, z^{(2)} \in \mathcal{Z}$, $z^{(1)}(s) = z^{(2)}(s)$, almost everywhere on $0 \leq s \leq \tau$, then $u(z^{(1)})_s = u(z^{(2)})_s$, a.e. on $0 \leq s \leq \tau$. The set of such $u(z)$ is denoted by Γ_U . Thus, the lower value $\hat{v}_0^*$ of the game is defined as

$$\hat{v}_0^* = \inf_{u(z) \in \Gamma_U} \sup_{z \in \mathcal{Z}_T} I(0, T; z, u(z))$$

(cf. Elliott and Kalton (1972), Bensoussan and Nagai (1997) and references therein), and it is characterized by $v_0(0, x)$ with the unique

solution v_0 to (7). The differential game is known to be related to H^∞ or robust control. Function $V(x)$ is often assumed to be bounded from below but not from above, and $S(x)$ is to be positive definite. In that case, if θ is very large, then H-J-B equation (6) may fail to have a solution (cf. Bensoussan and Nagai 1997 and Bensoussan and Nagai 2000). The size of θ ensuring the existence of a solution to (6) is related to the level of robustness which the above differential game concerns (Basar and Bernhard 1991; Whittle 1990; Bensoussan et al. 1998; Bensoussan and Nagai 1997, 2000).

Risk-Sensitive Asset Management

The idea of risk-sensitive control applies to mathematical finance (Fleming 1995; Bielecki and Pliska 1999). Consider a market model with $m+1$ securities, where the security prices are governed by the ordinary and stochastic differential equations

$$dS^0(t) = r(X_t)S^0(t)dt, \quad S^0(0) = s^0,$$

$$dS^i(t) = S^i(t)\{\alpha^i(X_t)dt + \sum_{k=1}^N \sigma_k^i(X_t)dB_t^k\},$$

$$S^i(0) = s^i,$$

$i = 1, \dots, m$, with an N -dimensional Brownian motion process $B_t = (B_t^1, B_t^2, \dots, B_t^N)$ defined on a filtered probability space $(\Omega, \mathcal{F}, P; \mathcal{F}_t)$. The volatilities σ , the instantaneous mean returns α of the risky assets, and the interest rate r of the risk-less asset are affected by the economic factors $(X_t^1, \dots, X_t^n)$ defined as the solution of the stochastic differential equation

$$dX_t = \beta(X_t)dt + \lambda(X_t)dB_t, \quad X(0) = x \in R^n.$$

Let us set the total wealth W_T an investor possesses at time T to be $W_T = \sum_i N_T^i S_T^i$ with number of the share N_T^i invested to i -th security S_T^i . Expected power utility maximization maximizing $\frac{1}{\gamma} E[W_T^\gamma] = \frac{1}{\gamma} E[e^{\gamma \log W_T}]$, $\gamma < 1, \neq 0$ (Karatzas and Shreve 2010; Merton 1998) is equivalent to

$$\sup \frac{1}{\gamma} \log E[e^{\gamma \log W_T}], \quad (8)$$

and it has been studied in terms of “risk-sensitive asset management.” When introducing the portfolio proportion h_t^i invested to i -th security defined by $h^i(t) = \frac{N^i(t)S^i(t)}{W(t)}$ for each $i = 0, \dots, m$ and setting $h(t)^* = (h^1(t), h^2(t), \dots, h^m(t))$, $h^0(t) = 1 - \sum_{i=1}^m h^i(t)$, the total wealth W_t turns out to satisfy

$$\begin{aligned} \frac{dW(t)}{W(t)} &= \{r(X_t) + h(t)^* \hat{\alpha}(X_t)\} dt \\ &+ h(t)^* \sigma(X_t) dB_t, \end{aligned}$$

under the self-financing condition, where $\hat{\alpha}(x) = \alpha(x) - r(x)\mathbf{1}$, $\mathbf{1} = (1, 1, \dots, 1)^*$. Thus, W_t is considered to be a solution of the stochastic differential equation with a given initial wealth W_0 and a portfolio proportion $h(t)$ invested to the risky assets. In considering the maximization problem, the portfolio proportion h_t is considered an investment strategy to be controlled and assumed to be $\mathcal{G}_t^{S,X} := \sigma(S(u), X(u), u \leq t)$ progressively measurable in the case of full information. The problem is often considered under partial information where h_t is assumed to be $\mathcal{G}_t^S := \sigma(S(u), u \leq t)$ measurable. Here we first discuss the case of full information, and the set of admissible strategies $\mathcal{A}(T)$ (or $\mathcal{A}$) is determined as the totality of $\mathcal{G}_t^{S,X}$ progressively measurable investment strategies satisfying some suitably defined integrability conditions.

Considering (8) for $\gamma < 0$ amounts to studying the minimization problem

$$\hat{v}(0, x) = \inf_{h \in \mathcal{A}(T)} \log E[e^{\gamma \log W_T(h)}]. \quad (9)$$

Then introducing a probability measure p^h defined by

$$p^h(A) = E \left[e^{\gamma \int_0^T h_s^* \sigma(X_s) dW_s - \frac{\gamma^2}{2} \int_0^T h_s^* \sigma \sigma^*(X_s) h_s ds} : A \right],$$

$A \in \mathcal{F}_T$, the value function is expressed as

$$\begin{aligned} \hat{v}(0, x) &= \gamma \log W_0 \\ &+ \inf_{h \in \mathcal{A}(T)} \log E^h [e^{-\gamma \int_0^T \eta(X_s, h_s) ds}] \end{aligned}$$

with the initial wealth W_0 , where

$$\eta(x, h) = -h^* \hat{\alpha}(x) + \frac{1-\gamma}{2} h^* \sigma \sigma^*(x) h - r(x)$$

and $\hat{\alpha}(x) = \alpha(x) - r(x)\mathbf{1}$. By using the Brownian motion $B_t^h := B_t - \gamma \int_0^t \sigma^*(X_s) h_s ds$ under the new probability measure P^h , the dynamics of the economic factor X_t is written as

$$dX_t = \{\beta(X_t) + \gamma \lambda \sigma^*(X_t) h_t\} dt + \lambda(X_t) dB_t^h. \quad (10)$$

Thus, we arrive at the risk-sensitive control problem with the value function $\hat{v}(0, x)$ and the controlled dynamics X_t governed by (10). Note that $\frac{1-\gamma}{2} > 0$ for $\gamma < 1$ and that the case where $\gamma < 0$ is called risk averse, which we mainly discuss here. Then the corresponding H-J-B equation is deduced as

$$\begin{cases} \frac{\partial v}{\partial t} + \frac{1}{2} \text{tr}[\lambda \lambda^* D^2 v] + \frac{1}{2} (Dv)^* \lambda \lambda^* Dv \\ + \inf_h \{[\beta + \gamma \lambda \sigma^* h]^* Dv - \gamma \eta(x, h)\} = 0, \\ v(T, x) = \gamma \log W_0, \end{cases} \quad (11)$$

which can be rewritten as

$$\begin{cases} \frac{\partial v}{\partial t} + \frac{1}{2} \text{tr}[\lambda \lambda^* D^2 v] + \beta_\gamma^* Dv \\ + \frac{1}{2} (Dv)^* \lambda N_\gamma^{-1} \lambda^* Dv - U_\gamma = 0, \\ v(t, x) = \gamma \log W_0. \end{cases} \quad (12)$$

Here $U_\gamma = -\frac{\gamma}{2(1-\gamma)} \hat{\alpha}^* (\sigma \sigma^*)^{-1} \hat{\alpha} + r(x)$, $\beta_\gamma = \beta + \frac{\gamma}{1-\gamma} \lambda \sigma^* (\sigma \sigma^*)^{-1} \hat{\alpha}$ and $N_\gamma^{-1} = I + \frac{\gamma}{1-\gamma} \sigma^* (\sigma \sigma^*)^{-1} \sigma$. Under suitable conditions H-J-B equation (12) has a solution with sufficient regularities (Bensoussan et al. 1998; Nagai 2003). Moreover, identification

$$v(0, x; T) \equiv v(0, x) = \hat{v}(0, x) \quad (13)$$

can be verified. Further, an optimal investment strategy for problem (9) is given by $\hat{h}(t, X_t) = \frac{1}{1-\gamma} (\sigma \sigma^*)^{-1} \{\hat{\alpha}(X_t) + \sigma \lambda^* Dv(t, X_t)\}$ using solution $v(t, x)$ to (11) (Nagai 2003).

A typical example is the case of linear Gaussian model such that $r(x) = r$, $\alpha(x) = Ax + a$, $\sigma(x) = \Sigma$, $\beta(x) = Bx + b$, $\lambda(x) = \Lambda$, where A , B , Σ , Λ are constant matrices, a , b are constant vectors, and r is a constant. Then, the

solution to (12) has an explicit representation as $v(t, x) = \frac{1}{2}x^*P(t)x + q(t)^*x + k(t)$, where $P(t)$ is the negative semi-definite solution to Riccati equation

$$\begin{aligned} \dot{P}(t) + P(t)\Lambda N^{-1}\Lambda^*P(t) + K_1^*P(t) + P(t)K_1 \\ + \frac{\gamma}{1-\gamma}A^*(\Sigma\Sigma^*)^{-1}A = 0, \quad P(T) = 0 \end{aligned} \quad (14)$$

and $q(t)$, $k(t)$ are, respectively, the solutions to

$$\begin{aligned} \dot{q}(t) + (K_1 + \Lambda N^{-1}\Lambda P(t))^*q(t) + P(t)b \\ + \frac{\gamma}{1-\gamma}(A^* + P(t)\Lambda\Sigma^*)(\Sigma\Sigma^*)^{-1}\hat{a} = 0, \\ q(T) = 0 \end{aligned} \quad (15)$$

and

$$\begin{aligned} \dot{k}(t) + \frac{1}{2}\text{tr}[\Lambda\Lambda^*P(t)] + \frac{1}{2}q(t)^*\Lambda\Lambda^*q(t) \\ + \frac{\gamma}{2(1-\gamma)}(\hat{a} + \Sigma\Lambda^*q(t))^*(\Sigma\Sigma^*)^{-1} \\ (\hat{a} + \Sigma\Lambda^*q(t)) = 0, \quad k(T) = \gamma \log W_0, \end{aligned} \quad (16)$$

where $K_1 := B + \frac{\gamma}{1-\gamma}\Lambda\Sigma^*(\Sigma\Sigma^*)^{-1}A$, $\hat{a} = a - r\mathbf{1}$ and $N^{-1} := I + \frac{\gamma}{1-\gamma}\Sigma^*(\Sigma\Sigma^*)^{-1}\Sigma$. In this case the optimal strategy has a more explicit form: $\hat{h}_t = \frac{1}{1-\gamma}(\Sigma\Sigma^*)^{-1}[\hat{a} + AX_t] + \frac{1}{1-\gamma}(\Sigma\Sigma^*)^{-1}[\Sigma\Lambda^*q(t) + \Sigma\Lambda^*P(t)X_t]$ (cf. Kuroda and Nagai 2002 and Davis and Lleo 2008).

The economic factor X_t may be more suitably considered to be unobservable, and then the problem should be formulated as the risk-sensitive stochastic control problem under partial information. Indeed, one can formulate the problem by regarding the log prices $Y_t^i := \log S_t^i$, $i = 0, 1, 2, \dots, m$ as the observable quantities and the economic factor X_t as the unobservable system process. As for linear Gaussian models and hidden Markov models, the problems are reduced to be the ones of full information by obtaining the relevant controlled dynamics in a finite dimension through deducing the filtering

equation by the methods of change of measure (Nagai 1999; Nagai and Runggaldier 2008). Further, one can obtain the explicit form of the optimal strategy, which is $\mathcal{G}_t^S$ measurable, in the case of linear Gaussian model (Nagai 1999) as the parallel result to the above.

The case of $0 < \gamma < 1$ is called risk seeking, and problem (8) is equivalent to

$$\sup_h \log E[e^{\gamma \log W_T(h)}] \quad (17)$$

and its infinite time horizon counterpart is

$$\bar{\chi}(\gamma) = \sup_h \overline{\lim}_{T \rightarrow \infty} \frac{1}{T} \log E[e^{\gamma \log W_T(h)}]. \quad (18)$$

These problems for linear Gaussian models are extensively studied in Fleming and Sheu (1999, 2002). If $0 < \gamma$ is small, then there is a stationary solution of (14), and the verification theorem holds for the problem on infinite horizon (so does for the problem on a finite time horizon). Further, under some conditions there is a threshold $\bar{\gamma}$ such that $\bar{\chi}(\gamma) = \infty$ for $\bar{\gamma} < \gamma$. To know explicitly the size of $\bar{\gamma}$ is important, while it is limited to the case of 1 – dimension to be realized.

Problems on Infinite Horizon

The value for the problem on infinite time horizon counterpart of (9) is defined as

$$\hat{\chi}(\gamma) = \inf_{h \in \mathcal{A}} \lim_{T \rightarrow \infty} \frac{1}{T} \log E[e^{\gamma \log W_T(h)}], \quad (19)$$

when suitably setting the set $\mathcal{A}$ of admissible strategies. The corresponding H-J-B equation of ergodic type for the problem is seen to be

$$\begin{aligned} \chi(\gamma) = \frac{1}{2}\text{tr}[\lambda\lambda^*D^2w] + \beta_\gamma^*Dw \\ + \frac{1}{2}(Dw)^*\lambda N_\gamma^{-1}\lambda^*Dw - U_\gamma. \end{aligned} \quad (20)$$

However, when setting as $\mathcal{A} = \{h_{\cdot|[0,T]} \in \mathcal{A}(T), \quad \forall T\}$, identification of $\hat{\chi}(\gamma)$ with the solution $\chi(\gamma)$ to the H-J-B equation (20) cannot

be seen in general. Indeed, even in the case of linear Gaussian model, such identification does not always hold (Fleming and Sheu 1999; Kuroda and Nagai 2002; Nagai 2003) if $\gamma < 0$. Instead, introduce the asymptotic value

$$\tilde{\chi}(\gamma) = \lim_{T \rightarrow \infty} \frac{1}{T} \inf_{h \in \mathcal{A}(T)} \log E[e^{\gamma \log W_T(h)}].$$

Then we can see that $\chi(\gamma) = \tilde{\chi}(\gamma)$ under sufficiently general conditions (cf. Hata et al. 2010 and Nagai 2012). Since verification (13) in the case of a finite time horizon is seen under fairly general conditions, we can obtain asymptotically optimal strategy for $\tilde{\chi}(\gamma)$. Further, if the solution $w(x)$ to (20) satisfies

$$(\nabla w)^* \lambda N_\gamma^{-1} \lambda^* \nabla w(x) - \hat{\alpha}^*(\sigma \sigma^*)^{-1} \hat{\alpha}(x) \rightarrow -\infty \quad (21)$$

as $|x| \rightarrow \infty$, then we can see $\hat{\chi}(\gamma) = \tilde{\chi}(\gamma)$, and an optimal strategy for (19) is given by $\tilde{h}(X_t) := \frac{1}{1-\gamma}(\sigma \sigma^*)^{-1} \{\hat{\alpha}(X_t) + \sigma \lambda^* \nabla w(X_t)\}$ (Nagai 2012).

In the case of linear Gaussian model, the solution to the H-J-B equation of ergodic type is given by $w(x) = \frac{1}{2} x^* \bar{P} x + \bar{q}^* x$ with the stationary solutions $\bar{P}$ of (14) and $\bar{q}$ of (15), and if

$$\bar{P} \lambda \Sigma^* (\Sigma \Sigma^*)^{-1} \Sigma \Lambda^* \bar{P} < A^* (\Sigma \Sigma^*)^{-1} A$$

holds, then one can see that $\chi(\gamma) = \hat{\chi}(\gamma)$ (Kuroda and Nagai 2002). Further, the optimal strategy is given by $\hat{h}_t = \hat{h}(X_t)$, with $\hat{h}(x) = \frac{1}{1-\gamma}(\Sigma \Sigma^*)^{-1} [\hat{\alpha} + \Sigma \Lambda^* \bar{q} + (A + \Sigma \Lambda^* \bar{P})x]$ (Kuroda and Nagai 2002). Decomposition as $\hat{h}_t = \frac{1}{1-\gamma} \hat{h}_t^1 + \frac{\gamma}{1-\gamma} \hat{h}_t^2 := \frac{1}{1-\gamma}(\Sigma \Sigma^*)^{-1} [\hat{\alpha} + A X_t] + \frac{\gamma}{1-\gamma}(\Sigma \Sigma^*)^{-1} [\Sigma \Lambda^* \bar{q} + \Sigma \Lambda^* \bar{P} X_t]$, $\bar{P} = \frac{1}{\gamma} \bar{P}$, $\bar{q} = \frac{1}{\gamma} \bar{q}$ is regarded as a generalization of Merton's mutual fund theorem (Davis and Lleo 2008; Merton 1998). Here $\hat{h}_t^1$ is a log utility portfolio (Kelly portfolio) (Kelly 1956). While, the partial information counterpart of Kuroda and Nagai (2002) was discussed in Nagai and Peng (2002) and the corresponding results are obtained.

Risk-sensitive investment management in the models admitting jumps in asset prices and also factor processes is studied by Davis and Lleo (cf. Davis and Lleo (2015) and references there in). They also consider the case when a benchmark is involved in the asset management.

In relation to the above problems on mathematical finance, a new kind of problem studying

$$I(\kappa) = \lim_{T \rightarrow \infty} \frac{1}{T} \inf_{h \in \mathcal{A}(T)} \log P(\log W_T(h) \leq \kappa T) \quad (22)$$

for a given constant κ arises, and it is called “downside risk minimization.” The problem is considered “large deviation control” and can be discussed as the dual to risk-sensitive asset management (19) in the risk-averse case $\gamma < 0$ (Hata et al. 2010; Nagai 2012, 2011). Indeed, we obtain

$$I(\kappa) = - \inf_{k \in (-\infty, \kappa]} \sup_{\gamma < 0} \{\gamma k - \hat{\chi}(\gamma)\}.$$

Further, an asymptotically optimal strategy is given as follows. For given κ , take $\gamma(\kappa)$ which attains the supremum in $\sup_{\gamma < 0} \{\gamma \kappa - \hat{\chi}(\gamma)\}$, then the optimal strategy $\hat{h}(t, X_t)$, $0 \leq t \leq T$ for problem (9) with $\gamma = \gamma(\kappa)$ forms the asymptotically optimal strategy for (22). If the solution $w(x)$ to (20) with $\gamma = \gamma(\kappa)$ satisfies (21), then

$$J(\kappa) := \inf_{h \in \mathcal{A}} \lim_{T \rightarrow \infty} \frac{1}{T} \log P(\log W_T(h) \leq \kappa T)$$

holds and $\tilde{h}(X_t) := \frac{1}{1-\gamma}(\sigma \sigma^*)^{-1} \{\hat{\alpha}(X_t) + \sigma \lambda^* \nabla w(X_t)\}$ forms an optimal strategy for $J(\kappa)$ (Nagai 2012).

Historically, the studies of “upside maximization” concerning

$$\bar{I}(\kappa) = \sup_{h \in \mathcal{A}} \lim_{T \rightarrow \infty} \frac{1}{T} \log P(\log W_T(h) \geq \kappa T)$$

have been preceding (cf. Pham 2003), and the duality relationship between this and (18) was discussed under restrictive conditions. To develop further studies for the problem, there are difficulties to know the size of threshold $\bar{\gamma}$, which is

mentioned at the end of section “[Risk-Sensitive Asset Management](#)”.

Cross-References

- [Credit Risk Modeling](#)
- [Financial Markets Modeling](#)
- [Investment-Consumption Modeling](#)

Bibliography

- Basar T, Bernhard P (1991) H^∞ – optimal control and related minimax design problems. Birkhäuser, Basel
- Bensoussan A (1992) Stochastic control of partially observable systems. Cambridge University Press, Cambridge/New York
- Bensoussan A, Nagai H (1997) Min – max characterization of a small noise limit on risk-sensitive control. SIAM J Control Optim 35:1093–1115
- Bensoussan A, Nagai H (2000) Conditions for no breakdown and Bellman equations of risk-sensitive control. Appl Math Optim 42:91–101
- Bensoussan A, Van Schuppen JH (1985) Optimal control of partially observable stochastic systems with an exponential-of-integral performance index. SIAM J Control Optim 23:599–613
- Bensoussan A, Frehse J, Nagai H (1998) Some results on risk-sensitive control with full information. Appl Math Optim 37:1–41
- Bielecki TR, Pliska SR (1999) Risk sensitive dynamic asset management. Appl Math Optim 39:337–360
- Davis M, Lleo S (2008) Risk-sensitive benchmarked asset management. Quant Finan 8:415–426
- Davis M, Lleo S (2015) Risk-sensitive investment management. World Scientific, Singapore
- Elliott RJ, Kalton NJ (1972) The existence of value in differential games. Memoirs Am Math Soc (126)
- Fleming WH (1995) Optimal investment models and risk-sensitive stochastic control. IMA Math Appl 65:75–88
- Fleming WH, McEneaney WM (1995) Risk-sensitive control on an infinite horizon. SIAM J Control Optim 33:1881–1915
- Fleming WH, Sheu SJ (1999) Optimal long term growth rate of expected utility of wealth. Ann Appl Probab 9(3):871–903
- Fleming WH, Sheu SJ (2002) Risk-sensitive control and an optimal investment model. II Ann Appl Probab 12(2):730–767
- Hata H, Nagai H, Sheu SJ (2010) Asymptotics of the probability minimizing a “down-side” risk. Ann Appl Probab 20:52–89
- Jacobson DH (1973) Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games. IEEE Trans Auto Control 18:124–131
- Karatzas I, Shreve SE (2010) Methods of mathematical finance. Springer, New York
- Kelly J (1956) A new interpretation of information rate. Bell Syst Tech J 35:917–926
- Kuroda K, Nagai H (2002) Risk sensitive portfolio optimization on infinite time horizon. Stochastics Stochastics Rep 73:309–331
- Merton RC (1998) Continuous time finance. Blackwell, Malden
- Nagai H (1996) Bellman equations of risk-sensitive control. SIAM J Control Optim 34:74–101
- Nagai H (1999) Risk-sensitive dynamic asset management with partial information. Stochastics in finite and infinite dimensions, a volume in honor of G. Kallianpur. Birkhäuser, Boston, pp 321–340
- Nagai H (2003) Optimal strategies for risk-sensitive portfolio optimization problems for general factor models. SIAM J Control Optim 41:1779–1800
- Nagai H (2011) Asymptotics of the probability minimizing a “down-side” risk under partial information. Quant Finan 11:789–803
- Nagai H (2012) Downside risk minimization via a large deviation approach. Ann Appl Probab 22:608–669
- Nagai H, Peng S (2002) Risk-sensitive dynamic portfolio optimization with partial information on infinite time horizon. Ann Appl Probab 12(1):173–195
- Nagai H, Runggaldier WJ (2008) PDE approach to utility maximization for market models with hidden Markov factors. In: Dalang RC et al (eds) Seminar on stochastic analysis, random fields and applications V, Progress in probability. Birkhäuser, Basel, pp 493–506
- Pham H (2003) A large deviations approach to optimal long term investment. Finance Stochast 7:169–195
- Whittle P (1981) Risk-sensitive linear/quadratic/Gaussian control. Adv Appl Probab 13:764–767
- Whittle P (1990) A risk-sensitive maximum principle. Syst Control Lett 15:183–192

Robot Grasp Control

R

Domenico Prattichizzo and
Maria Pozzi

Department of Information Engineering and
Mathematics, University of Siena, Siena, Italy
Advanced Robotics Department, Istituto
Italiano di Tecnologia, Genova, Italy

Abstract

Given a grasping system composed by a robotic hand and a grasped object, the aim of *grasp control* is to make the object follow a certain desired trajectory while preserving

the contact. In this essay, after a concise explanation of the mathematical model of a grasping system, with a particular focus on the contact constraints, a computed-torque controller that guarantees trajectory tracking and grasp maintenance is described. Last sections contain suggestions for further reading and information about open research challenges in the field of grasp control.

Keywords

Multifingered robotic hands · Grasping · Dexterous manipulation

Introduction

Grasp control refers to the art of controlling the motion of an object by constraining its dynamics through contacts with a hand. The process of controlling the grasp is not limited to robotic hands only, but also applies to human hands (Johansson and Edin 1991) and to all other mechanisms using contact constraints to control the motion of the manipulated object (Brost and Goldberg 1996).

A crucial role in control of grasping is played by contact constraints. All the interactions between the robotic hand and the grasped object occur at the contacts whose understanding is paramount (Salisbury and Roth 1983).

The unilateral nature of contact interaction in grasping makes the control problems much more challenging than cooperative manipulation where multiple arms hold the object rigidly allowing bilateral force transmission at each contact point (Chiacchio et al. 1991).

The importance of unilateral contact constraints in grasping led large part of the literature to focus on the closure properties of the grasp (Bicchi 1995). Those properties refer to the ability of a grasp to prevent motions of the grasped object relying only on unilateral frictionless constraints in case of form closure (Reuleaux 1876) and on contact constraints with friction in case of force closure (Nguyen 1988). While form closure is a purely geometric property of the grasp and depends on where the unilateral

contact points are on the object, force closure depends on the ability that the robotic hand has to resist and apply forces to the object through the contacts while satisfying the friction constraints. In other terms, force closure directly involves the control of the robotic hand kinematics and not only the geometry of the contacts (Bicchi 1995). This essay focuses on force-closed grasps.

The optimal choice of the contact points on the object surface is a critical issue known as grasp planning. Among the many optimal criteria that have been proposed in the literature to choose the contact points, we recall the one proposed in Ferrari and Canny (1992) where the grasping configuration is evaluated according to the magnitude of the largest worst-case disturbance wrench that can be resisted by the grasp.

Many approaches have been studied in the literature on grasp planning in presence of uncertainties. The uncertainty can be either due to the shape of the object which is partially known or partially sensed as in Goldfeder et al. (2009) or due to the errors in positioning the fingers on the object during the grasping (Roa and Suarez 2009). In what follows all the parameters of the grasp, including those related to the hand, the object and the contact points are assumed to be known with no uncertainties.

The main objective of grasp control is that of tracking a desired trajectory with the grasped object by applying a set of contact forces satisfying the friction constraints (Bicchi and Kumar 2000). Complex in-hand object motions can be obtained by rolling and sliding the contact points on the object surface as proposed in Montana (1988) or by using finger gaiting to get large-scale motions (Han and Trinkle 1998). This essay deals with non-rolling and non-sliding contact points and summarizes the fundamental theory of computed-torque control for object trajectory and internal force control proposed in Li et al. (1989).

For a comprehensive review of the theory of grasping and its control, the reader is referred to Murray et al. (1994), Shimoga et al. (1996), Okamura et al. (2000), Bicchi and Kumar (2000), and Prattichizzo and Trinkle (2016).

Grasp Modeling and Control

Contact and Grasp Model

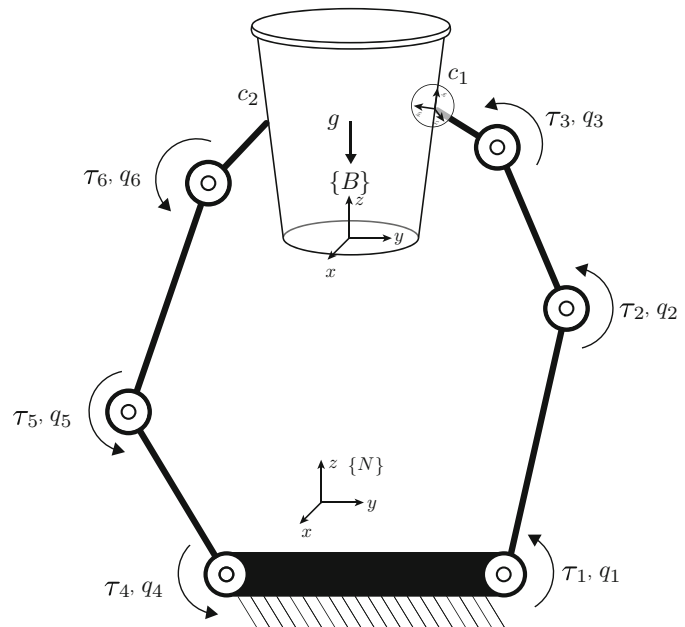
Notations and definitions on grasping are taken from Prattichizzo and Trinkle (2016). Refer to Fig. 1 and let $\{N\}$ represent the inertial frame fixed to the palm of the robotic hand. Let $u = [p^T, \phi^T]^T \in \mathbb{R}^6$ denote the vector describing the position and orientation of frame $\{B\}$, fixed to the object, relative to $\{N\}$. Vector ϕ expresses the Euler angles, the pitch-roll-yaw variables, or the exponential coordinates parameterizing $SO(3)$. Denote by $v = [v^T \omega^T]^T \in \mathbb{R}^6$, the twist of the object. It is worth to note that v is not equal to $\dot{u}$, but satisfies $v = U(u)\dot{u}$ where matrix $U \in \mathbb{R}^{6 \times 6}$ is such that UU^T is the identity matrix, and the dot over the variable implies differentiation with respect to time (Murray et al. 1994). The joint variables of the robotic hand are defined by $q = [q_1 \cdots q_{n_q}]^T \in \mathbb{R}^{n_q}$. Let n_c be the number of contact points. The position of contact point i in $\{N\}$ is defined by the vector $c_i \in \mathbb{R}^3$, in the contact point frame $\{C\}_i$ whose axes are $\{\hat{n}_i, \hat{t}_i, \hat{f}_i\}$ where the unit vector $\hat{n}_i$ is normal to the tangent plane at the contact, and directed toward the object while the other two unit vectors are orthogonal and lie in the tangent plane.

Two matrices are of utmost importance in grasp analysis: the *grasp matrix* G and the *hand Jacobian* J . These two matrices are computed using the complete grasp matrix, the complete Jacobian and the contact selection matrix that are defined as follows: the transpose of the *complete grasp matrix* $\tilde{G}^T \in \mathbb{R}^{6n_c \times 6}$ maps the object twist to the n_c twist vectors of the contact frames $\{C\}_i$ as thought on the object: $v_{c,obj} = \tilde{G}^T v$ while the *complete hand Jacobian Matrix* $\tilde{J} \in \mathbb{R}^{6n_c \times n_q}$ maps the joint velocities to the twists of the contact frames as thought on the hand: $v_{c,hnd} = \tilde{J}\dot{q}$.

When a contact occurs between the hand and the object, assuming no sliding, some components of the relative contact twist between the object and the hand are set to zero according to the used contact model. In this essay the *hard-finger* (HF) and the *soft-finger* (SF) contact models are considered (Mason and Salisbury 1985). Those components are selected by the *contact selection matrix* which selects m components of the relative contact twists for all the contacts and sets them to zero: $H(v_{c,hnd} - v_{c,obj}) = 0$. For more details on how to compute the contact selection matrix, the reader is referred to Prattichizzo and Trinkle

Robot Grasp Control,

Fig. 1 A two-fingered hand grasping an object



(2016). Then the following contact constraint equation is obtained:

$$\begin{bmatrix} J & -G^T \end{bmatrix} \begin{bmatrix} \dot{q} \\ v \end{bmatrix} = 0 \quad (1)$$

where the transpose of the grasp matrix and the hand Jacobian are defined by multiplying the contact selection matrix and the transpose of the complete grasp matrix and the complete hand Jacobian as

$$G^T = H\tilde{G}^T \in \mathbb{R}^{m \times 6}$$

$$J = H\tilde{J} \in \mathbb{R}^{m \times n_q}$$

In the force domain, the wrenches that the hand applies to the object at the contact points are collected in the vector λ . Correspondingly, on the hand, a force vector $-\lambda$, opposite to the preceding one, is applied by the object through the contact points. At each contact point, the contact wrenches have components only along the directions constrained by the contact model. Furthermore, contact force components must satisfy the friction constraints (see section “[Force Closure and Grasp Control](#)”). More specifically, the m -dimensional vector $\lambda = [\lambda_1^T \cdots \lambda_{n_c}^T]^T$ contains the contact wrenches components applied to the object through the n_c contacts, where the wrench at contact i is defined, for the different contact models here considered, as $\lambda_i = [f_{in} \ f_{it} \ f_{io}]^T$ for the HF contact model and $\lambda_i = [f_{in} \ f_{it} \ f_{io} \ m_{in}]^T$ for the SF contact model. The subscripts indicate one normal (n) and two tangential (t,o) components of contact force f_i and moment m_i at contact i .

In terms of forces, the grasp matrix maps the transmitted contact wrenches λ to the set of wrenches that the hand can apply to the object $G\lambda$, and the transpose of hand Jacobian maps the contact forces $-\lambda$ to the corresponding vector of joint loads $-J^T\lambda$.

Grouping all the non-contact wrenches applied to the object in $g \in \mathbb{R}^6$ and all the non-contact contributions to the joint loads of the robotic hand in $\tau \in \mathbb{R}^{n_q}$, rigid body dynamic equations of the whole system, consisting of hand and of the grasped object, are

$$M_{\text{obj}}(u)\dot{v} + N_{\text{obj}}(u, v) = G\lambda + g$$

$$M_{\text{hnd}}(q)\ddot{q} + N_{\text{hnd}}(q, \dot{q}) = -J^T\lambda + \tau$$

where $M_{\text{obj}}(\cdot)$ and $M_{\text{hnd}}(\cdot)$ are symmetric, positive definite inertia matrices and $N_{\text{obj}}(\cdot, \cdot)$ and $N_{\text{hnd}}(\cdot, \cdot)$ are the velocity-product terms for the object and the hand, respectively. For the sake of simplicity, the gravity terms are disregarded.

The dynamics of the hand and object are not independent but depend on the kinematic constraints imposed by the contact model (1):

$$\begin{bmatrix} -G \\ J^T \end{bmatrix} \lambda = \begin{bmatrix} \bar{g} \\ \bar{\tau} \end{bmatrix} \quad (2)$$

subject to $J\dot{q} = G^Tv$

where

$$\bar{g} = g - M_{\text{obj}}(u)\dot{v} - N_{\text{obj}}(u, v).$$

$$\bar{\tau} = \tau - M_{\text{hnd}}(q)\ddot{q} - N_{\text{hnd}}(q, \dot{q})$$

It is worth underlying that dynamics can be disregarded for slow motions of the hand and of the object while it becomes very relevant in applications with high-speed grasping and manipulation as discussed in Namiki et al. (2003).

Controllable Wrenches and Twists

From the first equation in (2) to impose any motion to the object by contact forces, the grasp matrix G must be full row rank, i.e., $\text{rank}(G) = 6$ which is equivalent to have a trivial null space of G^T , i.e., $\mathcal{N}(G^T) = 0$. This is an important property of the grasp which has been referred to as *non-indeterminate* in Prattichizzo and Trinkle (2016) to reflect the idea that the contacts on the object are placed in a way that there are no twists of the object that are not controllable by contact wrenches.

However, this condition depends only on the contacts on the object and does not consider the role of the hand kinematics which comes from the second equation in (2) and from the contact constraint. Under the simplifying assumption that $\mathcal{N}(J^T) = 0$, referred to as *non-defective* grasp in Prattichizzo and Trinkle (2016), it is simple

to verify that, for any given contact wrench λ , a control torque τ exists which is able to apply the given contact wrench. The mechanical interpretation of this assumption is that when $\mathcal{N}(J^T) = 0$, there are no contact forces resisted by the robotic hand constraints, i.e., with zero joint load. The simplifying assumption of non-defective grasps ensures that $\mathcal{N}(J^T) \cap \mathcal{N}(G) = 0$ which is a necessary condition to determine the contact force λ from the rigid body equation (2) as shown in Prattichizzo and Trinkle (2016).

If a grasp is non-defective, it means that each finger of the robotic hand involved in the contact with the object must have a number of joints sufficient to control all the components of the contact wrench. For example, in case of two HF contact points occurring at the fingertips of a two-fingered robotic hand, each finger must have at least three joints and must be in a non-singular configuration.

This essay does not consider whole hand or power grasps which, differently from the fingertip grasps, exploit the whole surface of the fingers, including the palm, to constraint the object. The analysis of controllable wrenches and twists for whole-arm grasps, taking into account the hand and object dynamics, can be found in Prattichizzo and Bicchi (1997).

Force Closure and Grasp Control

The dynamic formulation of the grasp with the contact kinematic constraints given in (2) holds only if the contact forces satisfy the friction law imposing constraints on the components of the contact force and moment. Limiting the analysis to HF contact models, Coulomb friction law requires that the components of contact force λ_i at the i -th contact lie inside the friction cone $\mathcal{F}_i$:

$$\mathcal{F}_i = \{(f_{in}, f_{it}, f_{io}) \mid \sqrt{f_{it}^2 + f_{io}^2} \leq \mu_i f_{in}\} \quad (3)$$

where μ_i represents the friction coefficient at the i -th contact. Extending to all contact points, λ is constrained to lie in $\mathcal{F}$ where $\mathcal{F}$ is the generalized friction cone defined as $\mathcal{F} = \mathcal{F}_1 \times \dots \times \mathcal{F}_{n_c} = \{\lambda \in \mathbb{R}^m \mid \lambda_i \in \mathcal{F}_i; i = 1, \dots, n_c\}$.

While grasping an object, the applied contact forces must be consistent with the friction constraints. This is not straightforward for the grasp control and requires to exploit the beneficial characteristics of the internal forces. From the object dynamics in (2), for a given $\bar{g}$ one gets

$$\lambda = -G^+ \bar{g} + N(G) \gamma \quad (4)$$

where G^+ denotes the generalized inverse of the grasp matrix and $N(G)$ denotes a matrix whose columns form a basis for $\mathcal{N}(G)$ and γ is a vector parameterizing the solution set. The contact force λ consists of a particular solution balancing the $\bar{g}$ term and of a homogeneous solution belonging to the null space of the grasp matrix.

In general, the particular solution $-G^+ \bar{g}$ does not satisfy the friction constraint (3) at all the contact points and needs the homogeneous solution $\lambda_h = N(G) \gamma$ to keep the contact forces within the friction cones. Contact forces λ_h in $\mathcal{N}(G)$ are referred to as *internal forces* since they do not contribute to the object dynamics, i.e., $G \lambda_h = 0$. Instead, these forces affect the tightness of the grasp and play a crucial role in maintaining grasps that rely on friction. The existence of a nontrivial null space of the grasp matrix is a desirable property and has been referred to as *graspability* (Prattichizzo and Trinkle 2016).

Another relevant and desirable property of the grasp is the *frictional force closure* which means that for any non-contact wrench $\bar{g}$, an internal force λ_h exists such that the contact force λ in (4) belongs to the generalized friction cone $\mathcal{F}$. In Murray et al. (1994), authors state that a grasp has frictional form closure if and only if the grasp matrix is full row rank (non-indeterminate grasp) and there exists λ_h such that $G \lambda_h = 0$ and λ_h belongs to the interior of the generalized friction cone $\mathcal{F}$.

Grasp control is about using contact forces, which must satisfy the friction constraints, to let the object track a given trajectory. This is also referred to as dexterous manipulation (Bicchi and Kumar 2000). In Li et al. (1989), a computed-torque controller is proposed to track both the

desired trajectory of the grasped object u_{des} and the desired internal force $\lambda_{h,\text{des}}$. Under the additional simplifying assumption that the robotic hand Jacobian is invertible, i.e., there are no redundant motions of the fingers, the computed-torque control law

$$\begin{aligned}\tau = & N_{\text{hnd}}(q, \dot{q}) + J^T G^+ N_{\text{obj}}(u, v) \\ & - M_{\text{hnd}} J(q) J^{-1} \dot{J} \dot{q} + M_{\text{ho}} \dot{U} \dot{u} + \\ & + M_{\text{ho}} U (\ddot{u}_{\text{des}} - K_v \dot{e}_u - K_u e_u) \\ & + J^T (\lambda_{h,\text{des}} - K_s \int e_{\lambda_h}),\end{aligned}$$

with $M_{\text{ho}} = M_{\text{hnd}} J(q) J^{-1} G^T + J^T G^+ M_{\text{obj}} J$, guarantees that both the trajectory and the internal force errors

$$\begin{aligned}e_u &= u - u_{\text{des}} \\ e_{\lambda_h} &= \lambda_h - \lambda_{h,\text{des}}\end{aligned}$$

with respect to the desired object trajectory u_{des} and internal force $\lambda_{h,\text{des}}$, converge to zero according to a second- and first-order dynamics, respectively.

$$\begin{aligned}\ddot{e}_u + K_v \dot{e}_u + K_u e_u &= 0 \\ e_{\lambda_h} + K_s \int e_{\lambda_h} &= 0\end{aligned}$$

where K_v , K_u , and K_s are positive definite matrices.

The computed-torque controller proposed in Li et al. (1989) guarantees only that the desired object trajectory and the desired internal forces are asymptotically tracked, but it does not ensure the non-violation of friction constraints by the contact forces. To guarantee that the contact force vectors remain in the friction cone during the manipulation, a force distribution problem must be solved at each time instant. The force closure assumption ensures that a solution exists that satisfies the friction constraints during the manipulation. This solution, which becomes the reference for the internal force control, can be found with an efficient algorithm, based on the minimization of a convex function, that

checks the force closure property at each time instant (Bicchi 1995).

Summary and Future Directions

The basic foundation of grasp control has been reviewed with a particular attention to modeling of contact constraints, force closure, and control of object motion and internal forces. This essay did not explicitly address grasp stability that is often equated to grasp closure, because all external forces can be balanced by the hand. A more formal analysis of grasp stability in terms of deflection from an equilibrium point has been proposed for robotic hands with general kinematics in Jen et al. (1996).

The computed-torque control is a classical approach to the grasp control. For a deeper study of other approaches to grasp control based on passivity theory, the reader is referred to Wimboeck et al. (2011).

Developments in underactuated robotic hands (Birglen et al. 2008) have led to a renewed interest in grasp control. Designing hands with a lower number of actuators has a lot of advantages in terms of robustness and reliability, but dramatically reduces the dexterous manipulation abilities which can be recovered only by devising new control algorithms or innovative designs (Odhner and Dollar 2011; Prattichizzo et al. 2013; Santina et al. 2015).

Together with underactuation, *softness* has become an important design feature of robotic hands (Catalano et al. 2014; Deimel and Brock 2016). Endowing multifingered hands with passive compliance enhances adaptability, robustness, and safety and allows new grasp planning strategies based on *environmental constraints exploitation* (Brock et al. 2016). Grasping with soft hands was not explicitly addressed by this essay because such devices are mainly used to perform power grasps (Pozzi et al. 2017). Thus, instead of building a precise model of the hand structure and finely control the grasp, usually the focus is put on the control of the robot arm and on the closing capabilities of the used robotic hand, so to well align it to the grasped object (Pozzi et al. 2018).

Cross-References

Grasp control deals with control of contact forces and motion of the grasped object. The reader can refer to the essays on ► [“Robot Motion Control”](#) and ► [“Force Control in Robotics”](#) of this Encyclopedia to study the subjects in more depth. It is worth highlighting that the control of grasping shares some aspects of problem formalization and methods with parallel robots. An essay on ► [“Parallel Robots”](#) is included in this Encyclopedia.

The essay on ► [“Robot Visual Control”](#) is important for readers interested to study the problem of grasp control with vision sensing. The use of cameras is very common in robotic grasping. If you think of how humans grasp objects, it is natural to think about grasping supported by vision especially during the reaching phase before grasping the object.

The last cross-reference is with the essay on ► [“Walking Robots”](#). During a manipulation task, it can happen that the robot needs to change the contact points to hold the object in another configuration. This is obtained moving the fingers to establish new contact points with the object. In the literature this is referred to as finger gaiting and shares common problems with the problem of walking robots.

Recommended Reading

Grasp synthesis and dexterous manipulation are important research topics. Grasp synthesis is the problem of choosing the posture of the hand and contact point locations to optimize a grasp quality metric. One of the first studies of grasp synthesis for multi-fingered hands was undertaken in Jameson (1985) where the author proposed a Levenberg-Marquardt algorithm to search the surface of an object for the locations of three points that would achieve force closure. Since this work, many other metrics and approaches to searching for high-quality grasps have been implemented as discussed in Nguyen (1988), Park and Starr (1992), Chen and Burdick (1993), Pollard (1997), Pozzi et al. (2017), and therein references. A new trend is to tackle grasp

synthesis with approaches based on machine learning techniques. There are methods based on deep learning, for example, that take as input a RGBD image or point cloud of the object and output an estimate of the best grasp pose that can be performed, without assuming a known 3D model of the object (Mahler et al. 2017; ten Pas et al. 2017). These algorithms achieve impressive success rates, but for the moment are mainly applied to simple two-fingered grippers.

Dexterous manipulation is the capability of manipulating an object so as to arbitrarily steer its configuration in space. Research on dexterous manipulation first appeared in Hanafusa and Asada (1979) where the authors developed a plan to turn a nut onto a bolt. Since then a progression of increasingly complex manipulation tasks has been studied to varying degrees of detail. For the planar case, the reader is referred to Mason (1982), Brost (1991), Peshkin and Sanderson (1988), Lynch (1996) and therein references, whereas for the three dimensional case, several interesting approaches have been proposed (Cherif and Gupta 1999; Han et al. 2000; Higashimori et al. 2007). More recently, for example, kinematic trajectory optimization and bio-inspired methods have been applied for planning and executing dexterous manipulation tasks with fully actuated, torque-controlled hands (Sundaralingam and Hermans 2017; Ruiz Garate et al. 2018).

Dexterous manipulation can be evaluated with manipulability ellipsoids of velocity and force as proposed in Chiacchio et al. (1991) for multiple-fingered systems and more recently in Prattichizzo et al. (2012) for underactuated robotic hands.

Authors recently created video lectures on grasp control that are freely available on YouTube (<http://sirslab.dii.unisi.it/GraspingCourse/index.html>) (Pozzi et al. 2019).

Bibliography

- Bicchi A (1995) On the closure properties of robotic grasping. *Int J Robot Res* 14(4):319–334
- Bicchi A, Kumar V (2000) Robotic grasping and contact: a review. In: *Proceedings of the IEEE international conference on robotics and automation*, pp 348–353

- Bicchi A, Prattichizzo D (1998) Manipulability of cooperating robots with passive joints. In: Proceedings of the IEEE international conference on robotics and automation, Leuven, pp 1038–1044
- Birglen L, Gosselin CM, Laliberté T (2008) Underactuated robotic hands, vol 40. Springer, Berlin
- Brock O, Park J, Toussaint M (2016) Mobility and manipulation. Springer International Publishing, Cham, pp 1007–1036
- Brost RC (1991) Analysis and planning of planar manipulation tasks. Ph.D. thesis
- Brost RC, Goldberg KY (1996) A complete algorithm for designing planar fixtures using modular components. *IEEE Trans Robot Autom* 12(1):31–46
- Catalano MG, Grioli G, Farnioli E, Serio A, Piazza C, Bicchi A (2014) Adaptive synergies for the design and control of the Pisa/IIT SoftHand. *Int J Robot Res* 33(5):768–782
- Chen IM, Burdick JW (1993) Finding antipodal point grasps on irregularly shaped objects. *IEEE Trans Robot Autom* 9(4):507–512
- Cherif M, Gupta KK (1999) Planning quasi-static fingertip manipulation for reconfiguring objects. *IEEE Trans Robot Autom* 15(5):837–848
- Chiacchio P, Chiaverini S, Sciacivico L, Siciliano B (1991) Global task space manipulability ellipsoids for multiple-arm systems. *IEEE Trans Robot Autom* 7(5):678–685
- Deimel R, Brock O (2016) A novel type of compliant and underactuated robotic hand for dexterous grasping. *Int J Robot Res* 35(1–3):161–185
- Ferrari C, Canny J (1992) Planning optimal grasps. In: Proceedings of the IEEE international conference on robotics and automation. IEEE, pp 2290–2295
- Goldfeder C, Ciocarlie M, Peretzman J, Dang H, Allen PK (2009) Data-driven grasping with partial sensor data. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, IROS 2009, pp 1278–1283
- Han L, Trinkle JC (1998) Dexterous manipulation by rolling and finger gaiting. In: Proceedings of the IEEE international conference on robotics and automation, vol 1. IEEE, pp 730–735
- Han L, Li Z, Trinkle JC, Qin Z, Jiang S (2000) The planning and control of robot dexterous manipulation. In: Proceedings of the IEEE international conference on robotics and automation, pp 263–269
- Hanafusa H, Asada H (1979) Handling of constrained objects by active elastic fingers and its applications to assembly. *Trans Soc Instrum Control Eng* 15(1): 61–66
- Higashimori M, Kimura M, Ishii I, Kaneko M (2007) Friction independent dynamic capturing strategy for a 2D stick-shaped object. In: Proceedings of the IEEE international conference on robotics and automation, pp 217–224
- Jameson J (1985) Analytic techniques for automated grasp. Ph.D. thesis
- Jen F, Shoham M, Longman RW (1996) Liapunov stability of force-controlled grasps with a multi-fingered hand. *Int J Robot Res* 15(2):137–154
- Johansson RS, Edin BB (1991) Mechanisms for grasp control. In: Pedotti A, Ferrarin M (eds) Restoration of walking for paraplegics. Recent advancements and trends, 3rd edn. Edizioni Pro Juventute, IOS Press, pp 57–65
- Li Z, Hsu P, Sastry S (1989) Grasping and coordinated manipulation by a multifingered robot hand. *Int J Robot Res* 8(4):33–50
- Lynch K (1996) Nonprehensile manipulation: mechanics and planning. Ph.D. thesis
- Mahler J, Liang J, Niyaz S, Laskey M, Doan R, Liu X, Ojea JA, Goldberg K (2017) Dex-net 2.0: deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*
- Mason MT (1982) Manipulator grasping and pushing operations. PhD thesis, Massachusetts Institute of Technology. Reprinted in *Robot Hands and the Mechanics of Manipulation*, MIT Press, Cambridge, MA (1985)
- Mason MT, Salisbury JK (1985) Robot hands and the mechanics of manipulation. The MIT Press, Cambridge, MA
- Montana DJ (1988) The kinematics of contact and grasp. *Int J Robot Res* 7(3):17–32
- Murray RM, Li Z, Sastry SS (1994) A mathematical introduction to robotic manipulation. CRC Press, Boca Raton
- Namiki A, Imai Y, Ishikawa M, Kaneko M (2003) Development of a high-speed multifingered hand system and its application to catching. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, vol 3. IEEE, pp 2666–2671
- Nguyen V (1988) Constructing force-closure grasps. *Int J Robot Res* 7(3):3–16
- Odhner LU, Dollar AM (2011) Dexterous manipulation with underactuated elastic hands. In: Proceedings of the IEEE international conference on robotics and automation, pp 5254–5260
- Okamura AM, Smaby N, Cutkosky MR (2000) An overview of dexterous manipulation. In: Proceedings of the IEEE international conference on robotics and automation, pp 255–262
- Park YC, Starr GP (1992) Grasp synthesis of polygonal objects using a three-fingered robot hand. *Int J Robot Res* 11(3):163–184
- Peshkin MA, Sanderson AC (1988) Planning robotic manipulation strategies for workpieces that slide. *IEEE J Robot Autom* 4(5):524–531
- Pollard NS (1997) Parallel algorithms for synthesis of whole-hand grasps. In: Proceedings of the IEEE international conference on robotics and automation
- Pozzi M, Malvezzi M, Prattichizzo D (2017) On grasp quality measures: grasp robustness and contact force

- distribution in underactuated and compliant robotic hands. *IEEE Robot Autom Lett* 2(1):329–336
- Pozzi M, Salvietti G, Bimbo J, Malvezzi M, Prattichizzo D (2018) The closure signature: a functional approach to model underactuated compliant robotic hands. *IEEE Robot Autom Lett* 3(3):2206–2213
- Pozzi M, Malvezzi M, Prattichizzo D (2019) MOOC on the art of grasping and manipulation in robotics: design choices and lessons learned. In: Lepuschitz W, Merdan M, Koppensteiner G, Balogh R, Obdrzalek D (eds) *Robotics in education*. Springer International Publishing, Cham, pp 71–78
- Prattichizzo D, Bicchi A (1997) Consistent task specification for manipulation systems with general kinematics. *J Dyn Syst Meas Control* 119(4):760–767
- Prattichizzo D, Trinkle JC (2016) Grasping. In: Siciliano B, Khatib O (ed) *Springer handbook of robotics*. Springer Science & Business Media, Berlin, pp 955–988
- Prattichizzo D, Malvezzi M, Gabiccini M, Bicchi A (2012) On the manipulability ellipsoids of underactuated robotic hands with compliance. *Robot Auton Syst* 60(3):337–346. Elsevier
- Prattichizzo D, Malvezzi M, Gabiccini M, Bicchi A (2013) On motion and force controllability of precision grasps with hands actuated by soft synergies. *IEEE Trans Robot* 29(6):1440–1456
- Reuleaux F (1876) *The kinematics of machinery*. Macmillan. Republished by Dover, New York (1963)
- Roa MA, Suarez R (2009) Computation of independent contact regions for grasping 3-D objects. *IEEE Trans Robot* 25(4):839–850
- Ruiz Garate V, Pozzi M, Prattichizzo D, Tsagarakis N, Ajoudani A (2018) Grasp stiffness control in robotic hands through coordinated optimization of pose and joint stiffness. *IEEE Robot Autom Lett* 3(4):3952–3959
- Salisbury JK, Roth B (1983) Kinematic and force analysis of articulated mechanical hands. *J Mech Transm Autom Des* 105(1):35–41
- Santina CD, Grioli G, Catalano M, Brando A, Bicchi A (2015) Dexterity augmentation on a synergistic hand: the Pisa/IIT SoftHand+, pp 497–503
- Saxena A, Driemeyer J, Ng AY (2008) Robotic grasping of novel objects using vision. *Int J Robot Res* 27(2):157–173
- Shimoga KB (1996) Robot grasp synthesis algorithms: a survey. *Int J Robot Res* 15(3):230–266
- Sundaralingam B, Hermans T (2017) Relaxed-rigidity constraints: in-grasp manipulation using purely kinematic trajectory optimization. In: *Proceedings of robotics: science and systems*, Cambridge, MA
- ten Pas A, Gualtieri M, Saenko K, Platt R (2017) Grasp pose detection in point clouds. *Int J Robot Res* 36(13–14):1455–1473
- Wimboeck T, Ott C, Albu-Schaffer A, Hirzinger G (2011) Comparison of object-level grasp controllers for dynamic dexterous manipulation. *Int J Robot Res* 31(1):3–23

Robot Learning

Jens Kober

Cognitive Robotics Department, Delft University of Technology, Delft, The Netherlands

Abstract

With increasingly complex robot hardware and the push to enable robots to work in unstructured environments, there has been an increasing interest in applying machine learning and artificial intelligence techniques to systems and control problems in robotics. At the same time, many machine learning approaches have been developed with robotic problems as motivating use cases. Robot learning spans the problems of learning models, sensing, acting, as well as integrated approaches and has been applied to many types of robot embodiments. The algorithms cover most fields of machine learning. However, due to the specific challenges in robotics, these either need to be adapted or new approaches have to be developed.

Keywords

Supervised learning · Unsupervised learning · Reinforcement learning · End-to-end learning · Model learning

Introduction

Robot learning, as the name indicates, is the application of machine learning and artificial intelligence methods to robotic problems. Hence the field of robot learning is almost as broad as robotics itself, and applications range from the optimization of mechanical designs, over perception, high-level decision-making, and planning, to low-level control on all different kinds of robots like autonomous vehicles, drones, legged robots, or robot arms, on their own or in multi-robot settings, potentially also including interactions with humans.

Recently, there has been tremendous progress in the field of machine learning, in particular due to the availability of big data and the drastically increased computational power. The progress in deep learning has enabled many new applications, often outperforming classical, hand-designed approaches to an extent that those older methods have been largely superseded by learning-based methods. For example, image recognition, speech recognition, automated translations, text to speech, recommender systems, or text mining frequently are AI-based and widely deployed online and in smartphone apps. All these components are invaluable to build a complete robot that can naturally interact with human users and its environment. Here we will not focus on such components but rather on approaches that are closer to systems and control: namely, low-level and high-level control and planning, perception and sensor fusion, as well as models of the robot and its environment. Robot learning in this setting has close ties both to computer science and systems and control, in particular to optimal control, adaptive control, and system identification. In this article we will first discuss different types of machine learning, then outline different application areas of robot learning, and finally discuss the particular challenges of robot learning.

Types of Learning

In machine learning, different types of learning need to be distinguished: supervised learning, unsupervised learning, and reinforcement learning. Those approaches differ in what kind of data is available to the learning algorithm and in what it is supposed to learn from that data. Application areas will be discussed in the next section.

Supervised Learning

In supervised learning we assume that the learner has access to samples of input-output combinations. The task of the learner is to reproduce and to generalize this mapping, i.e., it also needs to interpolate between and to extrapolate from the samples. The tricky part here is to find a good,

compact representation of the mapping that only models the relevant parts and not unnecessary details like noise. The optimization criterion is hence a combination of fitting the observed data, predicting unobserved data well (which is in practice achieved by holding out a part of the dataset), and encouraging simple models (regularization). Supervised learning has close ties to system identification. Within supervised learning, there is an additional subdivision in classification and regression. In classification we have a fixed, finite set of labels (or their probability) that need to be predicted. In regression the learned system acts like a function with continuous outputs. Machine learning typically assumes independent and identically distributed samples, while data selection in robotics can be a major challenge.

Unsupervised Learning

In unsupervised learning the learner only has access to input data. The task is to discover structures in the data which enable lower-dimensional representations. Typical applications are clustering, where data needs to be grouped into several clusters with similar properties, and generative models, where the aim is to learn models that generate “fake” data that is distributed as similar as possible to the real data.

Reinforcement Learning

Reinforcement learning does not operate on a fixed dataset but rather learns the optimal mapping from inputs to outputs through interactions with the system. This mapping is a controller, called policy, mapping from states to optimal actions. The agent receives a reward, i.e., a scalar value that tells it how well it did, in every step, and its goal is to optimize for the cumulative (potentially discounted) reward, called the return. In contrast to optimal control, typically rewards are considered instead of costs, which do not necessarily need to conform to a particular form. Furthermore, reinforcement learning usually considers the underlying system dynamics and reward functions to be unknown, and the agent hence needs to implicitly learn them as well through interactions with the

system. Reinforcement learning approaches can be largely classified into model-based methods that first learn a model and then apply optimal control methods to derive the policy and model-free methods that directly learn a policy or a value function, i.e., the desirability of each state.

Combinations

As already alluded, the distinction between these types of learning is not that clear-cut, and combining them can be very beneficial. Semi-supervised learning describes the case in between supervised and unsupervised learning, i.e., where only some of the data points are labeled and the labels of the other points need to be inferred from the structure of the data. Reinforcement learning can employ models that have been learned via supervised learning, or can be initialized with policies learned by supervised learning, that are then further refined. Active learning denotes approaches where the agent does not operate on a fixed dataset but has the option to actively query for more data, e.g., for labels of inputs that it is unsure about.

Scope of Learning

In robot learning the big question is always “What parts of the system do we learn?”. Traditionally, learning approaches have been applied to one or several components in an overall system, usually the ones that are hard to design in a traditional manner. In a control system, these components could, for instance, replace an observer, a system model, or a controller. With the advent of deep learning, end-to-end learning has become possible, i.e., learning the complete mapping from raw, high-dimensional sensor inputs to control actions. The former approach integrates well with traditional system design approaches and makes optimal use of existing knowledge; the latter enables applications that were previously infeasible and potentially results in better overall performance, removing bottlenecks of specific components.

Components

In robotics, we traditionally distinguish components for sensing and acting. A third category are models of the system and environment that can be used both in sensing and acting.

Models

Here models of the environment and/or the system are learned. This ranges from learning models of the dynamics of the robot and subsequently using those in an inverse-dynamics control scheme to learning prediction models of human behavior that then can be used to plan collision-free movements. For a detailed overview, see Nguyen Tuong and Peters (2011). A large part of the work in this domain falls into the domain of supervised learning, where both the inputs and outputs of the robot system or parts of the environment are observed and the learning agent tries to fit a model that best explains and reproduces the observed behaviors. Similar to system identification, gray box models or black box models can be learned. In gray box models, the structure of the model is known from insights into how the system behaves and only parameter values need to be fitted. In black box models, we either assume a very generic parametrization or apply nonparametric models. Depending on the intended application, different types of models are learned: forward models or inverse models. A forward model predicts the next state, given the current state and the taken action. A forward model can be used for model-based controller design or simply as a component to predict the outcome of an action. Examples include using learned models of the system in optimal control, e.g., via iLQG, or in planning frameworks where they are used for predictions that are hard to model analytically, in a model-based filter for perception, or to replace components that are too time-consuming to solve analytically in every time-step. Also see an example in Fig. 1. Inverse models predict which action needs to be taken given a current state and a desired next state. Inverse models can hence be employed more directly in model-based control. In many cases an inverse model is not unique or an ill-posed problem. Learning

Robot Learning, Fig. 1

An example for learning models: the learned model is subsequently employed to learn inverted helicopter flight. (Source (Ng et al. 2006) with permission of Andrew Ng)



a forward and an inverse model simultaneously, called a mixed-model, can often be helpful to learn consistent models. Popular methods include linear regression, support vector regression, locally weighted projection regression, Gaussian mixture models, Gaussian processes, and (deep) neural networks.

Rather than learning models in a supervised fashion, there are also many approaches that learn generative models, which is a form of unsupervised learning. See Bengio et al. (2013) for a more extensive survey. Internally such generative models learn structures that form compact representations of the system. While the training of such generative models is often a form of supervised learning, i.e., minimizing the distance between real and fake data, the final result corresponds to unsupervised learning. Popular approaches here include hidden Markov models, Gaussian mixture models, variational autoencoders, and generative adversarial networks. A related problem is learning latent representations or state representations, i.e., a mapping from observation, which are high-dimensional inputs, to a state, which is a compact representation that is sufficient to describe the behavior of the system. Many learning approaches based on neural networks implicitly learn such state representations through imposed restrictions, e.g., deeper layers with limited size. When learning state representations explicitly, one of the most prominent criteria is autoencoding, i.e., being able to reconstruct the original input from the compact

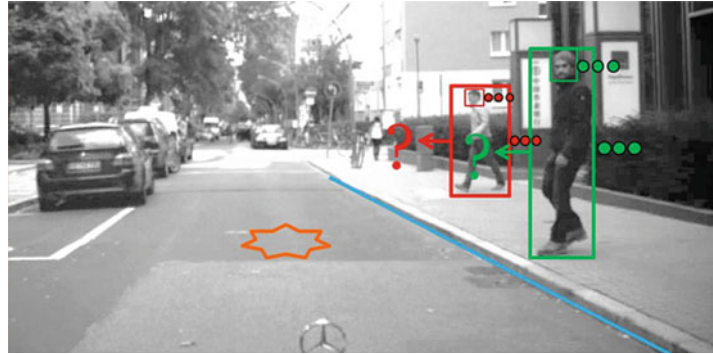
state, but can also include additional criteria like predicting the next state or rewards. For an extensive survey see (Lesort et al. 2018).

Sensing

A robot relies on sensors to perceive and understand its own state and the state of its environment. For extensive surveys, see Sünderhauf et al. (2018), Schwarting et al. (2018), Premebida et al. (2018) and Ruiz-del Solar et al. (2018). While some sensors, like joint encoders, deliver usable information almost directly, other sensors (e.g., like conventional cameras, time-of-flight cameras, LiDAR, radar, tactile arrays, or microphones) deliver high-dimensional inputs that first need to be processed before control decisions can be taken. Applications range from classification, i.e., detecting whether an object or feature is present or not, localizing objects, fusing multiple sensors, to high-level scene understanding. Examples include recognizing if there are any obstacles in front of the robot, detecting road boundaries, localizing sounds or objects to be picked up, recognizing the pose of humans, self-localization, estimating depth from 2D images, or combining information from GPS, cameras, and LiDAR. Also see an example in Fig. 2. Many approaches carry over from the communities that focus on exclusively on the perception part, like computer vision. Some of the specific challenges and advantages are real-time requirements, observation noise, dealing with false positives/negatives, representing uncertainty in

Robot Learning, Fig. 2

An example for sensing combined with a model: the position of pedestrians and their heads is detected and their paths predicted based on a learned model. (Source (Kooij et al. 2019) with permission of Julian Kooij)



perception, the possibility to include information from previous time-steps, and the possibility to do *active perception*, i.e., actively influence the environment by performing actions that improve perception (Bohg et al. 2017).

Acting

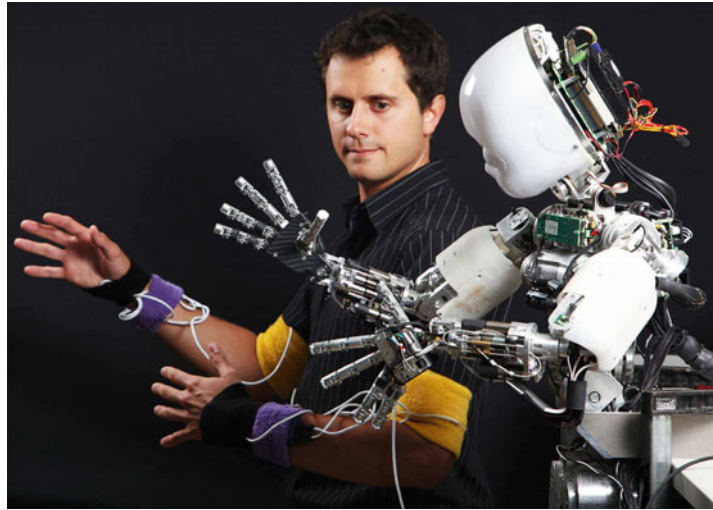
Controllers can learn on very different levels. Approaches range from low-level control of motors (currents or torques) to high-level decision-making and planning. Typically, the respective lower-level controllers are assumed to be fixed, and the higher-level learning approach has to learn to cope with their imperfections. For example, we could learn a task on the trajectory level, where the trajectory provides a position and velocity reference, which is translated to low-level motor commands via traditionally designed controllers. Some approaches also learn hierarchical controllers. Elementary tasks that take place in the order of a few seconds, like grasping or individual moves in a sport, are often called motor primitives. The goal of learning can then be to learn only one such primitive, to learn how to sequence several of these to achieve more complex tasks, or to do both at the same time.

One of the popular approaches for learning controllers is called *imitation learning* or learning from demonstration, which is a form of supervised learning; see Argall et al. (2009), Billard et al. (2016) and Osa et al. (2018) for extensive surveys. Here the learning agent's task is to imitate demonstrations from a human teacher or another agent. Repeating exactly the observed behavior is not the only objective; frequently

generalization to related but previously unseen situations is more important. One of the most important questions is 'what to imitate?', i.e., determining the relevant parts of the demonstration. Imitation learning can further be subdivided in the two fields of *behavioral cloning*, where the goal is to directly imitate the control policy, and *inverse reinforcement learning* (also known as inverse optimal control), where the goal is to imitate the optimization criterion of the demonstrations. The former is a more direct form of imitation, while the latter allows for easier transfer to different embodiments and situations. Some considerations in imitation learning include the type and quality of the demonstrations and the type of controllers we want to learn (Calinon and Lee 2019). Approaches range from learning open-loop controllers, e.g., desired trajectories, to closed-loop controllers. If demonstrations are provided by kinesthetic teach-in, i.e., taking the robot by the end-effector and moving it, or via tele-operation, the input-output relations are directly contained in the demonstrations; also see an example in Fig. 3. On the other extreme, if we want to learn from instruction videos, quite a lot of additional steps are required to extract useful information and to map it to the robot morphology. A related question is how to deal with sub-optimal and contradictory demonstrations. Many inverse reinforcement learning approaches rely on the availability of models of the system dynamics to reconstruct the reward function. The general idea is that under the reconstructed reward functions, the demonstrations are optimal; all other policies are then sub-optimal. Related approaches do

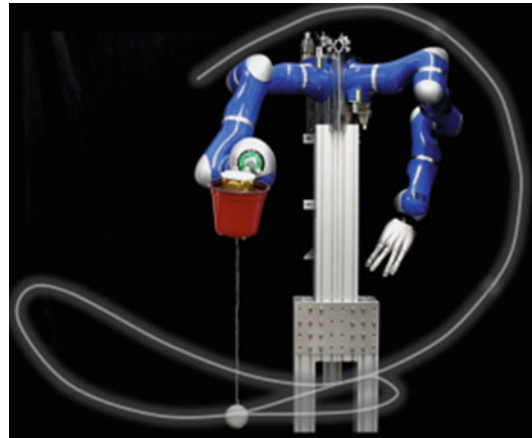
Robot Learning,

Fig. 3 An example for behavioral cloning: the robot imitates the movement observed via motion capture. (Source (Calinon et al. 2010) with permission of Sylvain Calinon)



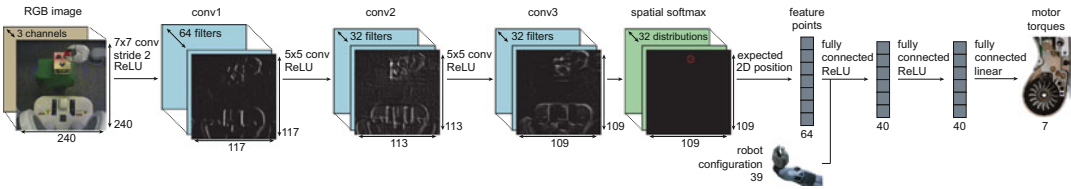
not attempt to learn the reward function from actions but rather directly from rewards or preferences the teacher assigns to various behaviors. Rather than learning behaviors or rewards from a fixed dataset, there are also interactive learning approaches that allow the teacher to provide additional demonstrations, feedback, or corrections while observing the learning process or the robot.

Another popular approach is *reinforcement learning*. Popular methods include optimal control, value function methods, actor-critic methods, gradient-based and gradient-free policy search, evolutionary strategies, and their respective deep versions. For extensive surveys, see Kober et al. (2013), Deisenroth et al. (2013), Tai et al. (2016), Arulkumaran et al. (2017), Chatzilygeroudis et al. (2019) and Sigaud and Stulp (2019). In the context of robotics, reinforcement learning has to cope with a number of specific challenges: typically its states and actions are continuous and high-dimensional. Real-world samples are expensive in terms of money and time; in most cases we want the robot to learn in a few trials, making data-hungry approaches like deep reinforcement learning difficult to apply directly. Especially, we do not want a robot to damage itself, its environment, or humans. One way to overcome this challenge is to learn the policy in simulation, i.e., based on models. However, reinforcement learning



Robot Learning, Fig. 4 An example for policy search: learning to catch the ball in the cup can be sped up by including intermittent human corrections (Source (Celemin et al. 2019) own work)

approaches tend to exploit model inaccuracies, requiring particular attention to render the transfer to real robots feasible. One approach is iteratively learning policies and improving the models. Another crucial component in reinforcement learning is the reward function, which needs to be specified by the engineer. Even if the robot perfectly learns to optimize a reward function, there might be unforeseen side effects, e.g., if you only tell the robot to remove all weeds in the garden, the flowers might be gone as well. Another important consideration



Robot Learning, Fig. 5 An example of an end-to-end architecture: the inputs are raw camera images and joint encoders, the outputs motor torques. (Source (Levine et al. 2016) with permission of Sergey Levine)

is how much the reward function should guide the learning process: on the one extreme, we have very sparse and uninformative rewards that only provide a reward once the robot has successfully completed the whole task, which typically requires very large numbers of trials. On the other extreme, we have reward functions that indicate the search direction, which allows the robot to learn faster but will not allow the robot to come up with truly novel strategies. To sum up, the specification of the reward function is often more an art than science. *Policy search* approaches and evolutionary methods have some advantages over value function-based methods in this setting as they allow more straightforward inclusion of prior knowledge, allow limiting the learning to safe updates, and often learn faster; also see an example in Fig. 4. However, this comes at the price of only optimizing locally.

End-to-End

With the recent rapid progress in deep learning, learning the whole control system simultaneously has become feasible. Rather than learning sensing and acting separately, we can now learn the mapping directly from raw sensor inputs to control actions, which might be very low level or more high level. The latter can then be seen as end-to-end learning for planning. End-to-end approaches make the learning problem significantly harder and typically require huge amounts of data. We will discuss methods to overcome this challenge in the next section. End-to-end learning has been applied to many different imitation learning and reinforcement learning problems, ranging from imitating behaviors from human demonstration videos, reinforcement learning of manipulation skills, to controlling an autonomous car; also see the example in Fig. 5

Summary and Future Directions

Robot learning spans the whole spectrum of supervised learning, unsupervised learning, and reinforcement learning and has been applied to virtually all systems and control problems in particular for sensing, acting, and models. Similar to most fields in machine learning, deep learning is currently one of the main topics and has enabled learning approaches that can deal with raw, high-dimensional sensor data. Many of the general machine learning and artificial intelligence techniques are not directly applicable to robotics due to a number of specific challenges. These, as well as steps toward overcoming them, will be outlined next.

Challenges in Robot Learning

While robots generate huge amounts of raw data, one of the biggest challenges is the high cost of *labeled* data. Tremendous effort is put into manually labeling data for supervised learning. This is similar for reinforcement learning: real robot trials are expensive, and only very few skills have been learned end-end on real robots without including task-specific prior knowledge.

Prior knowledge can be included in many different forms. The most common form is pre-structuring the learning problem and hence limiting the search space. This approach includes picking appropriate inputs and outputs to reflect what the actual interesting learning problem is, e.g., the level of the control actions or how much the inputs are pre-processed. A related approach is pre-structuring the representations, e.g., picking a suitable compact representation of the controller like a trajectory or an LQR,

defining the structure of the neural network, or even specifying a curriculum for the learning, i.e., learning a sequence of increasingly harder problems. Another form of prior knowledge is instrumented training where additional information is available during the training but no longer required once the behavior has been learned. As already mentioned above, a pre-training step can be included, e.g., a neural network might be trained first on a generic dataset, or reinforcement learning can be initialized with an imitation learning step.

Finally, models and simulators serve as a major source of prior knowledge. These can be employed to provide rough directions to steer the learning process or to completely replace data from the real system. The latter poses the problem of transferring the behaviors learned in simulation to the real system. In most cases there will be some differences between the simulation and the real world. When directly transferring the behaviors, they typically lose performance or fail completely. There are a few methods to deal with this transfer problem: we can abstract the differences away, i.e., working on a level that is sufficiently abstract that lower-level components take care of these differences. We can also make the models more similar to the real world, e.g., by incrementally learning better models or inversely learning components that make the real world more similar to the simulation. There is also the option to learn behaviors that are sufficiently robust to handle the differences, which might however result in a loss in performance. For example, the behaviors can be learned in a simulation that randomizes the visual appearance, randomizes physics parameters, adds noise, or a combination thereof that are larger than the differences between the simulation and the real system.

A final challenge is safe learning, i.e., avoiding damages to the robot and its environment. Here approaches range from traditional, low-level safety components, over enforcing gradual changes in the behavior, to approaches that explicitly consider safety in the updates. See Garcia and Fernández (2015) for a comprehensive survey in the field of reinforcement learning.

Future Directions

Many of the challenges above have been successfully addressed in various scenarios, but none of them have been solved in general. One of the major open questions still is sample efficient learning and how to deal with poor or incomplete data. Another open challenge is safe learning: How can we introduce guarantees, certification, and verification and make the learning process and the final result interpretable, especially if the system needs to generalize to previously unseen environments and situations? How do we prevent robots from learning undesirable actions, either accidentally or by malicious attacks? More generally the question of what should be learned is still open: Where and how do we best integrate traditional methods, e.g., from systems and control? How do we best integrate human expertise, experience, and intuition? Finally we have scalability questions: How do we go from learning confined tasks to learning for global systems that are interacting with an ever-changing environment and tasks? How can we make robot learning tools sufficiently robust and general to be able to create toolboxes for roboticists of other subfields and ultimately for end users in home environments and at their workplace?

Cross-References

- [Deep Learning in a System Identification Perspective](#)
- [Iterative Learning Control](#)
- [Learning Control: the Probably Approximately Correct Framework](#)
- [Reinforcement Learning for Approximate Optimal Control](#)

Recommended Reading

The field of robot learning is very active and broad. A good starting point are the surveys referenced above and textbooks on the underlying machine learning methods like (Sutton and Barto 2018; Bishop 2006; Hastie et al. 2009; Russell and Norvig 2009; Murphy 2012; Goodfellow et al. 2016).

Bibliography

- Argall BD, Chernova S, Veloso M, Browning B (2009) A survey of robot learning from demonstration. *Robot Auton Syst* 57(5):469–483
- Arulkumaran K, Deisenroth MP, Brundage M, Bharath AA (2017) Deep reinforcement learning: a brief survey. *IEEE Signal Process Mag* 34(6):26–38
- Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. *IEEE Trans Pattern Anal Mach Intell* 35(8):1798–1828
- Billard A, Calinon S, Dillmann R (2016) *Chap learning from humans*. Handbook of robotics, 2nd edn. Springer, Secaucus, pp 1995–2014
- Bishop CM (2006) *Pattern recognition and machine learning*. Springer, London
- Bohg J, Hausman K, Sankaran B, Brock O, Kragic D, Schaal S, Sukhatme GS (2017) Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics* 33(6):1273–1291
- Calinon S, Lee D (2019) *Chap learning control*. Humanoid robotics: a reference. Springer, Dordrecht
- Calinon S, D'halluin F, Sauser EL, Caldwell DG, Billard AG (2010) Learning and reproduction of gestures by imitation. *IEEE Robot Autom Mag* 17(2):44–54
- Celemin CE, Maeda G, Ruiz-del-Solar J, Peters J, Kober J (2019) Reinforcement learning of motor skills using policy search and human corrective advice. *Int J Robot Res* 38(14):1560–1580
- Chatzilygeroudis K, Vassiliades V, Stulp F, Calinon S, Mouret JB (2019) A survey on policy search algorithms for learning robot controllers in a handful of trials. *IEEE Trans. on Robotics* – accepted
- Deisenroth MP, Neumann G, Peters J (2013) A survey on policy search for robotics. *Found Trends® Robot* 2(1–2):1–142
- García J, Fernández F (2015) A comprehensive survey on safe reinforcement learning. *J Mach Learn Res* 16(1):1437–1480
- Goodfellow I, Bengio Y, Courville A (2016) *Deep learning*. MIT Press, Cambridge
- Hastie T, Tibshirani R, Friedman JH (2009) *The elements of statistical learning: data mining, inference, and prediction*, 2nd edn. Springer series in statistics, Springer, New York
- Kober J, Bagnell JA, Peters J (2013) Reinforcement learning in robotics: a survey. *Int J Robot Res* 32(11):1238–1274
- Kooij JFP, Flohr F, Pool EAI, Gavrila DM (2019) Context-based path prediction for targets with switching dynamics. *Int J Comput Vis* 127(3):239–262
- Lesort T, Díaz-Rodríguez N, Goudou JF, Filliat D (2018) State representation learning for control: An overview. *Neural Networks*. Elsevier 108:379–392
- Levine S, Finn C, Darrell T, Abbeel P (2016) End-to-end training of deep visuomotor policies. *J Mach Learn Res* 17(1):1334–1373
- Murphy KP (2012) *Machine learning: a probabilistic perspective*. The MIT Press, Cambridge
- Ng AY, Coates A, Diel M, Ganapathi V, Schulte J, Tse B, Berger E, Liang E (2006) Autonomous inverted helicopter flight via reinforcement learning. In: *Experimental robotics IX*. Springer, Berlin/Heidelberg, pp 363–372
- Nguyen Tuong D, Peters J (2011) Model learning in robotics: a survey. *Cogn Process* 12(4):319–340
- Osa T, Pajarinen J, Neumann G, Bagnell J, Abbeel P, Peters J (2018) An algorithmic perspective on imitation learning. *Found Trends® Robot*
- Premebida C, Ambrus R, Marton ZC (2018) *Chap intelligent robotic perception systems*. Applications of mobile robots. IntechOpen, London, pp 111–127
- Russell S, Norvig P (2009) *Artificial intelligence: a modern approach*, 3rd edn. Prentice Hall Press, Upper Saddle River
- Schwarting W, Alonso-Mora J, Rus D (2018) Planning and decision-making for autonomous vehicles. *Ann Rev Control Robot Auton Syst* 1(1):187–210
- Sigaud O, Stulp F (2019) Policy search in continuous action domains: an overview. *Neural Netw* 113:28–40
- Sünderhauf N, Brock O, Scheirer W, Hadsell R, Fox D, Leitner J, Upcroft B, Abbeel P, Burgard W, Milford M, Corke P (2018) The limits and potentials of deep learning for robotics. *Int J Robot Res* 37(4–5):405–420
- Ruiz-del Solar J, Loncomilla P, Soto N (2018) A survey on deep learning methods for robot vision. *arXiv preprint:180310862*
- Sutton RS, Barto AG (2018) *Reinforcement learning: an introduction*, 2nd edn. The MIT Press
- Tai L, Zhang J, Liu M, Boedecker J, Burgard W (2016) A survey of deep network solutions for learning control in robotics: from reinforcement to imitation. *arXiv preprint:161207139*

Robot Motion Control

Mark W. Spong

Department of Systems Engineering, Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, TX, USA

Abstract

The motion control problem for robot manipulators is to determine a time sequence of control inputs to achieve a desired motion. The control inputs are usually motor voltages or currents but can be translated into

velocities or torques for the purpose of control design. The desired motion is typically given by a reference trajectory, consisting of positions, velocities, and accelerations that are generated from motion planning and trajectory generation algorithms designed to calculate collision-free paths, taking into account various kinematic and dynamic constraints on the robot. In this chapter we give an overview of some basic and advanced control methods for motion control of robot manipulators.

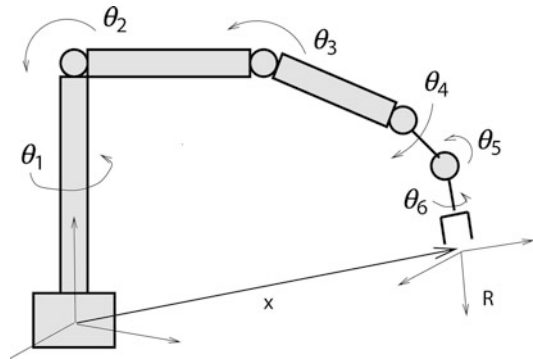
Keywords

Inverse dynamics · Feedback linearization · Trajectory planning · Passivity-based control · PID control · Robust control · Adaptive control · Feedforward control

Introduction

We consider the motion control problem for an n -degree-of-freedom robot manipulator, such as shown schematically in Fig. 1. The variables $\theta_1, \dots, \theta_n$ are the *joint variables*, which define the *configuration* of the robot at each instant of time. The manipulator *kinematics* defines the position and orientation, also called the *pose*, of the robot end-effector as a function of the joint variables.

A robot manipulator is fundamentally a positioning device *designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks* (Robot Institute of America, 1980). Thus, manipulator tasks, such as materials transfer, welding, painting, and even tasks involving the control of interaction forces, such as assembly or grinding, are performed through the coordination of the motion of the joints of the robot. A typical robot control architecture is shown in Fig. 2, which is designed to translate *sensing* into *action* through motion planning, trajectory generation, and feedback control. In this entry we concentrate on the function of the controller.

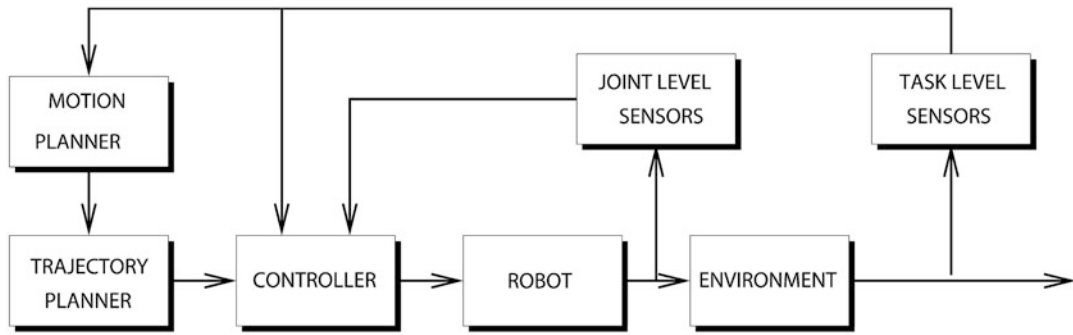


Robot Motion Control, Fig. 1 Six-link robot manipulator. The values of the joint variables determine the position and orientation of the end-effector

Motion Planning

The desired joint motions are specified as reference trajectories (positions, velocities, and accelerations) generated from motion planning algorithms that must determine collision-free paths for the manipulator taking into account various kinematic and dynamic constraints on the robot (Lavelle 2006). A detailed discussion of motion planning is outside the scope of this entry. The motion planning problem begins by decomposing a given task into a discrete set of end-effector motions. A continuous path for the end-effector in the *task space* is then computed, taking into account issues of joint limits and collisions with objects in the workspace, including self-collisions. Finding optimal paths in configuration space is computationally complex, and methods have been developed to determine feasible, suboptimal paths using various methods such as artificial potential functions, grid search, and roadmaps (Lavelle 2006).

Once a feasible path in task space is determined, a *trajectory*, which is a time-parameterized function in task space or configuration space, is computed. To compute configuration space or joint space trajectories from task space trajectories, the *inverse kinematics* of the manipulator is used.



Robot Motion Control, Fig. 2 Control architecture for robot control. The controller is designed so that the joint variables track the reference trajectory based on sensory

feedback. Not shown are noise and disturbance inputs, which limit the achievable performance of the controller

Trajectory Generation

To simplify computation, joint-level trajectories are typically generated by calculating the inverse kinematics only at discrete points along the task space trajectory and then interpolating between these points. Two of the most common interpolation schemes utilize either *polynomials* in time or *trapezoidal velocity* profiles. Figures 3 and 4 show examples of a cubic polynomial trajectory and a trapezoidal velocity trajectory, respectively.

The control design objective is to choose the control in such a way that the plant output, θ , tracks or follows a desired output, given by the reference signal, θ^r . The control signal, however, is not the only input acting on the system. *Disturbances*, which are really inputs that we do not control, also influence the behavior of the output. Therefore, the controller must be designed, in addition, so that the effects of the disturbance on the plant output are reduced. If this is accomplished, the plant is said to *reject* the disturbances. The twin objectives of *tracking* and *disturbance rejection* are central to any control methodology.

Independent Joint Control

The simplest approach to control design for a multi-degree-of-freedom manipulator is to treat

each joint of the robot as a single-input/single-output (SISO) system and design independent controllers for each joint. The dynamic coupling among the joints is treated as a disturbance. *Proportional, integral, derivative (PID)* control is the most common method employed in this case. This approach works well for geared manipulators moving at relatively low speeds, since a large gear reduction and low speed tend to reduce the coupling effects among the various links.

Let the dynamics of a single degree-of-freedom (joint) of a robot be represented as

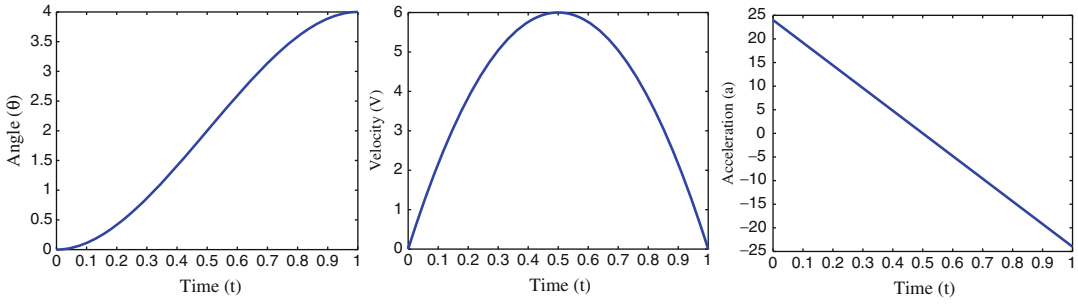
$$J\ddot{\theta} + B\dot{\theta} = u + d$$

where θ is the joint angle, u and d are the control input and disturbance input, respectively, with units of *torque*, and J and B represent the joint inertia and damping, respectively. A PID compensator for the above system is of the form

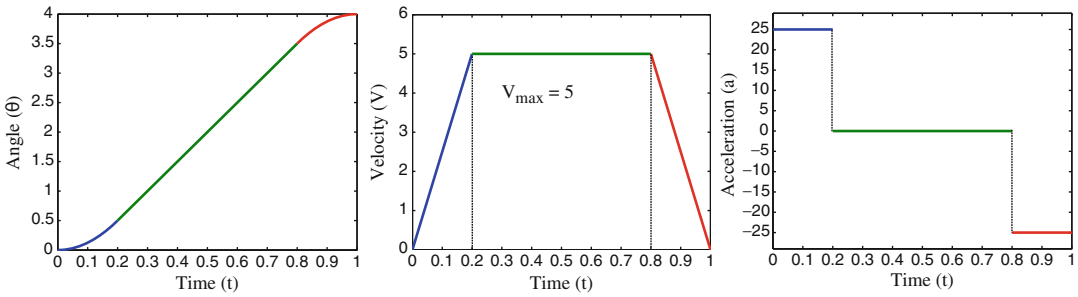
$$u(t) = K_P e(t) + K_I \int_0^t e(\sigma) d\sigma + K_D \frac{de}{dt} \quad (1)$$

where $e(t) = \theta(t) - \theta^r(t)$ is the *tracking error*. The controller parameters are the *proportional gain* K_P , *integral gain* K_I , and *derivative gain* K_D . A block diagram of the system is shown below in Fig. 5. The *plant* $P(s)$ and *compensator* $C(s)$ transfer functions are

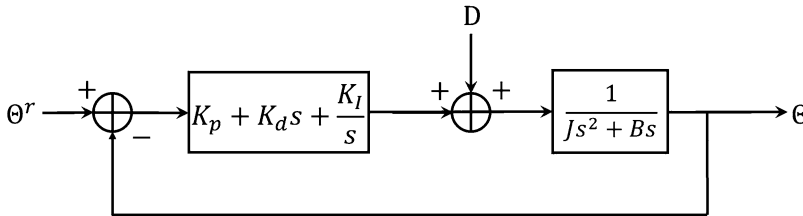
$$P(s) = \frac{1}{Js^2 + Bs} \quad (2)$$



Robot Motion Control, Fig. 3 A cubic polynomial reference trajectory. Position (left); velocity (center); acceleration (right) versus time



Robot Motion Control, Fig. 4 Trapezoidal velocity profile. Position (left); velocity (center); acceleration (right) versus time



Robot Motion Control, Fig. 5 The system with a PID compensator. K_P , K_I , and K_D are the proportional, integral, and derivative gains. θ^r is the joint angle reference signal to be tracked and D is the disturbance input

and

$$C(s) = K_P + K_D s + K_I/s$$

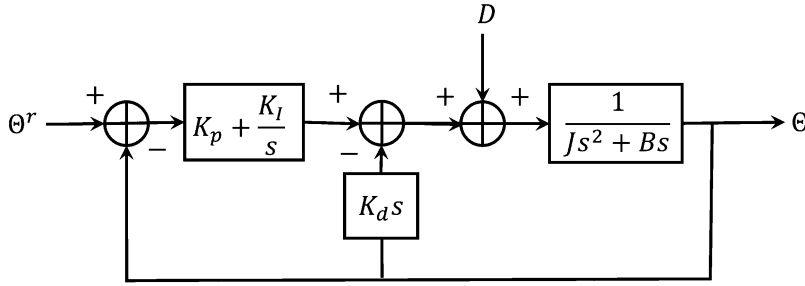
$$= \frac{K_D s^2 + K_P s + K_I}{s} \quad (3)$$

A special case of the PID compensator is a PD compensator in which the integral gain K_I is set to zero. Referring to Equation (3), the PID Compensator adds a pole (at $s = 0$) and two zeros (roots of $K_D s^2 + K_P s + K_I$) to the forward transfer function (The PD compensator adds a single zero and no pole). The design problem, known as

tuning, is then to choose the PID gains K_P , K_D , and K_I to achieve the desired performance.

Two-Degree-of-Freedom Controller

In Fig. 5, the compensator acts on the error signal. If the reference θ^r is a step reference, then the derivative term will introduce an impulse at $t = 0$. To avoid this we may use the alternative architecture shown in Fig. 6, which is called a *two-degree-of-freedom* controller. In this architecture, the input to the derivative term $K_D s$ is the output signal θ , rather than the error signal



Robot Motion Control, Fig. 6 The system with a two-degree-of-freedom PID compensator

$\theta - \theta^r$, which avoids differentiating the step input θ^r at time $t = 0$, where the step function is discontinuous.

The Closed-Loop Systems

Whether we use the controller configuration in Fig. 5 or Fig. 6, we must analyze the appropriate transfer functions to assess the performance of the closed-loop systems. The transfer functions of relevance are, in each case, $G_1(s) = \frac{\Theta(s)}{\Theta^r(s)}$, the transfer function from the reference to the output, and $G_2(s) = \frac{\Theta(s)}{D(s)}$, the transfer function from the disturbance to the output. By the *Principle of Superposition* then, the overall system response, in each case, is given by

$$\Theta(s) = G_1(s)\Theta^r(s) + G_2(s)D(s)$$

Table 1 show the transfer functions $G_1(s)$ and $G_2(s)$ for the one- and two-degree-of-freedom architectures for both PID and PD compensators.

Set-Point Tracking

The *set-point tracking problem* refers to the case that the reference input θ^r is constant and arises in point-to-point motion. Since the reference input is a step signal, we will use the two-degree-of-freedom compensator in Fig. 6.

We first consider a PD compensator by setting the integral gain K_I equal to zero. In this case, we see from Table 1 that the closed-loop system is given by

$$\Theta(s) = \frac{K_P}{\Omega(s)}\Theta^r(s) + \frac{1}{\Omega(s)}D(s) \quad (4)$$

where $\Omega(s)$ is the closed-loop characteristic polynomial

$$\Omega(s) = Js^2 + (B + K_D)s + K_P \quad (5)$$

The closed-loop system will be stable for all positive values of K_P and K_D and bounded disturbances, and the tracking error $E(s)$ is given by

$$\begin{aligned} E(s) &= \Theta^r(s) - \Theta(s) \\ &= \frac{Js^2 + Bs}{\Omega(s)}\Theta^r(s) - \frac{1}{\Omega(s)}D(s) \end{aligned} \quad (6)$$

For a step reference input

$$\Theta^r(s) = \frac{\Theta^r}{s} \quad (7)$$

and a constant disturbance

$$D(s) = \frac{D}{s} \quad (8)$$

it follows directly from the final value theorem that the steady state error e_{ss} satisfies

$$e_{ss} = \lim_{s \rightarrow 0} sE(s) = \frac{-D}{K_P} \quad (9)$$

We see that the steady state error can be made arbitrarily small by making the position gain K_P large.

Robot Motion Control, Table 1 Closed-loop transfer functions using both the one- and two-degree-of-freedom architectures. a) and d) represent $G_1(s)$ in the one-degree-

of-freedom architecture. b) and e) represent $G_1(s)$ in the two-degree-of-freedom architecture. c) and f) represent $G_2(s)$

With PID Compensator	With PD Compensator
a) $\frac{K_D s^2 + K_P s + K_I}{J s^3 + (B + K_D) s^2 + K_P s + K_I}$	d) $\frac{K_D s + K_P}{J s^2 + (B + K_D) s + K_P}$
b) $\frac{K_P s + K_I}{J s^3 + (B + K_D) s^2 + K_P s + K_I}$	e) $\frac{K_P}{J s^2 + (B + K_D) s + K_P}$
c) $\frac{s}{J s^3 + (B + K_D) s^2 + K_P s + K_I}$	f) $\frac{1}{J s^2 + (B + K_D) s + K_P}$

Using a PD compensator, the closed-loop system is second order and hence the step response is determined by the closed-loop natural frequency ω and damping ratio ζ . Given a desired value for these quantities, the gains K_D and K_P can be found from the expression

$$s^2 + \frac{(B + K_D)}{J} s + \frac{K_P}{J} = s^2 + 2\zeta\omega s + \omega^2 \quad (10)$$

as

$$K_P = \omega^2 J, \quad K_D = 2\zeta\omega J - B \quad (11)$$

It is customary in robotics applications to take the damping ratio $\zeta = 1$ so that the response is critically damped. In this context ω determines the speed of response.

With $J = B = 1$, Fig. 7a shows step responses for several values of the natural frequency ω with no disturbance (left plot) and with a constant disturbance (right plot).

PID Control

Adding an integral term K_I/s to the above results in the closed-loop transfer functions b) and c) in Table 1. The PID control achieves exact steady tracking of step inputs while rejecting step disturbances, provided of course that the closed-loop system is stable.

The closed-loop system is now the third order system

$$\Theta(s) = \frac{(K_P s + K_I)}{\Omega_2(s)} \Theta^r(s) + \frac{s}{\Omega_2(s)} D(s) \quad (12)$$

with characteristic polynomial

$$\Omega_2 = J s^3 + (B + K_D) s^2 + K_P s + K_I \quad (13)$$

Applying the Routh-Hurwitz criterion to this polynomial, it follows that the closed-loop system is stable if the gains are positive, and in addition,

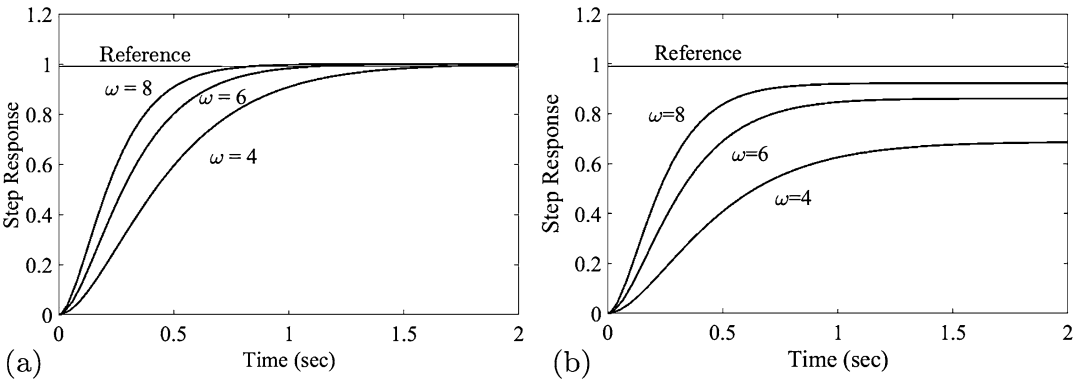
$$K_I < \frac{(B + K_D) K_P}{J} \quad (14)$$

A common design rule-of-thumb for PID control is to first set $K_I = 0$ and design the proportional and derivative gains, K_P and K_D , to achieve the desired transient behavior (rise time, settling time, and so forth) and then to choose K_I within the limits imposed by (14) to remove the steady state error.

Figure 8 shows the response of the system with a constant disturbance using a PID compensator. We see that the steady state error due to the disturbance is removed.

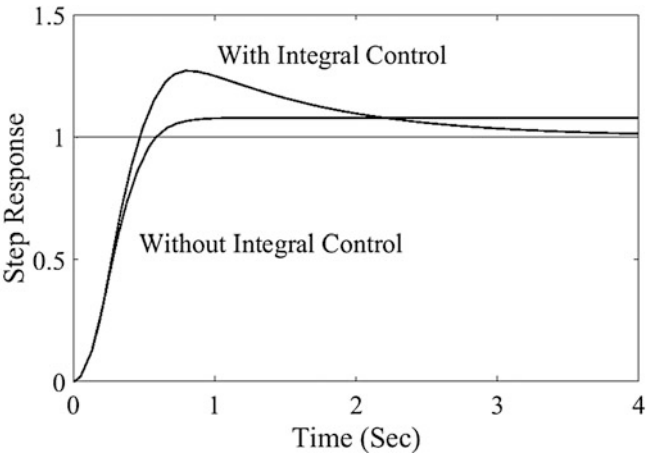
Feedforward Control

In order to track nonconstant reference signals, such as a cubic polynomial trajectory or trapezoidal velocity trajectory, a *feedforward* term may be superimposed on the PID control signal as shown in Fig. 9. Under the condition that the plant $P(s)$ is *minimum phase*, the feedforward transfer

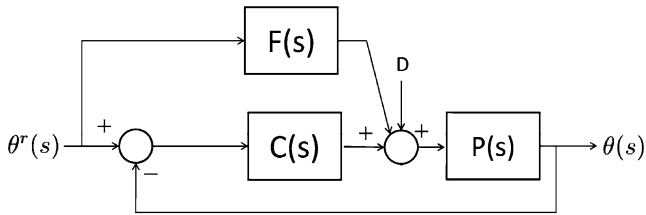


Robot Motion Control, Fig. 7 Second order step responses with PD control. The speed of response, as measured by rise time, is faster for larger values of the natural frequency ω . Likewise, the steady-state error due to a constant disturbance is smaller for larger values of ω

Robot Motion Control, Fig. 8 Response with integral control action showing that the steady state error to a constant disturbance has been removed



Robot Motion Control, Fig. 9 Feedforward control architecture



function $F(s)$ can be taken as $1/P(s)$, the inverse of the plant. This guarantees asymptotic tracking of any time-varying reference trajectory provided the closed-loop transfer function is stable.

Advanced Control Methods

Advanced control methods generally aim to take into account issues such as dynamic coupling and

elasticity; uncertainty in the inertia parameters; and robustness to sensor noise and other effects. A common model of the dynamics of n -link, rigid robots, without consideration of friction and elasticity, is given by the so-called *Euler-Lagrange* equations (Spong et al. 2006)

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = u \tag{15}$$

where θ is the n -dimensional vector of joint variables as in Fig. 1. The n -dimensional vectors,

$\dot{\theta}$ and $\ddot{\theta}$, are then the joint velocities and accelerations, respectively. The $n \times n$ matrix, $M(\theta)$, is called the inertia matrix. The vectors $C(\theta, \dot{\theta})\dot{\theta}$ and $g(\theta)$ are Coriolis and centrifugal forces and gravitational forces, respectively. Equation (15) is a system of n coupled, nonlinear, second-order equations.

Feedback Linearization Control

An intuitive method of control for the system (15) is the method of *inverse dynamics*, also called *feedback linearization*, which computes the input torque u according to

$$u = M(\theta)a + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) \quad (16)$$

where a is an additional (outer-loop) control. Choosing the outer-loop control a , for example, as

$$a = \ddot{\theta}^r + K_d(\dot{\theta}^r - \dot{\theta}) + K_p(\theta^r - \theta) \quad (17)$$

results in a closed-loop system, in terms of the tracking error, $e(t) = \theta(t) - \theta^r(t)$, that satisfies the linear equation

$$\ddot{e} + K_d\dot{e} + K_p e = 0 \quad (18)$$

and therefore, the tracking error converges exponentially to zero for diagonal matrices K_p and K_d with positive diagonal elements and any given reference trajectory, $\theta^r(t)$.

Robust and Adaptive Control

There are several practical challenges to the method of feedback linearization control discussed in the previous section. For example, in order to compute Equation (16), one must have exact knowledge of the parameters defining Equation (15). In addition, effects of compliance, friction, and so on are not modeled by Equation (15), and so the stability and performance of the system predicted by Equation (18) may not be achieved in practice. This has stimulated a great deal of research into robust and adaptive control, control of elasticity, and other issues.

In distinguishing between robust control and adaptive control, we follow the commonly accepted notion that a robust controller is a fixed controller designed to satisfy performance specifications over a given range of uncertainties, whereas an adaptive controller incorporates some sort of online parameter estimation. This distinction is important. For example, in a repetitive motion task, the tracking errors produced by a fixed robust controller would tend to be repetitive as well, whereas tracking errors produced by an adaptive controller might be expected to decrease over time as the plant and/or control parameters are updated based on runtime information. At the same time, adaptive controllers that perform well in the face of parametric uncertainty may not perform well in the face of other types of uncertainty such as external disturbances or unmodeled dynamics.

Robust Feedback Linearization

If we denote by $\hat{M}(\theta)$, $\hat{C}(\theta, \dot{\theta})$, and $\hat{g}(\theta)$ expressions for the terms $M(\theta)$, $C(\theta, \dot{\theta})$, and $g(\theta)$ in the equations of motion (15) based on nominal or estimated values of the true parameters, we can define a control input

$$u = \hat{M}(\theta)(a + \delta a) + \hat{C}(\theta, \dot{\theta})\dot{\theta} + \hat{g}(\theta) \quad (19)$$

where a is as defined in Equation (17) and δa represents an additional control intended to compensate for the parameter uncertainty. This leads to the state space model in terms of the tracking error e

$$\dot{e} = Ae + B\{\delta a + \eta\}$$

where η represents the *uncertainty* resulting from inexact cancellation of nonlinearities and

$$A = \begin{bmatrix} 0 & I \\ -K_0 & -K_1 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

Under the assumption that the uncertainty is bounded as $\|\eta\| \leq \rho(e, t)$, the control term δa can be chosen as

$$\delta a = \begin{cases} -\rho(e, t) \frac{B^T P e}{\|B^T P e\|} & ; \text{ if } \|B^T P e\| > \epsilon \\ -\frac{\rho(e, t)}{\epsilon} B^T P e & ; \text{ if } \|B^T P e\| \leq \epsilon \end{cases}$$

The Lyapunov function

$$V = e^T P e \quad (20)$$

where P is a symmetric, positive definite matrix satisfying a Lyapunov equation

$$A^T P + P A + Q = 0 \quad (21)$$

for a given symmetric positive definite matrix Q can be used to show uniform ultimate boundedness of all trajectories, where the size of the ultimate boundedness set depends on ϵ (Spong et al. 2006). This is a practical notion of asymptotic stability in the sense that the tracking errors can be made small.

Passivity-Based Control

Passivity-based control is an alternative to feedback linearization control and relies on some fundamental structural properties of the Euler-Lagrange equations, primarily *linearity in the parameters* and *passivity*.

The *passivity property* (Ortega and Spong 1989) of robot dynamics follows from the fact that the matrix $N(q, \dot{q}) = \dot{M}(q) - 2C(q, \dot{q})$ is skew symmetric (Spong et al. 2006). This property implies that the total energy E of the robot satisfies

$$\dot{E} = \dot{\theta}^T u \quad (22)$$

and can be used to design provably correct robust and adaptive control laws.

Linearity in the Parameters

The robot equations of motion are defined in terms of certain parameters, such as link masses, moments of inertia, etc. The complexity of the

dynamic equations makes the determination of these parameters a difficult task. Fortunately, the equations of motion are linear in these inertia parameters in the following sense: There exists an $n \times \ell$ matrix function $Y(q, \dot{q}, \ddot{q})$ and an ℓ -dimensional constant vector Φ such that the Euler-Lagrange equations can be written as

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = Y(\theta, \dot{\theta}, \ddot{\theta})\Phi \quad (23)$$

The function $Y(\theta, \dot{\theta}, \ddot{\theta})$ is called the *regressor* and $\Phi \in R^\ell$ is the *parameter vector*. The dimension of the parameter space, that is, the number of parameters needed to write the dynamics in this way, is not unique, and finding a minimal set of parameters that can parameterize the dynamic equations is difficult in general.

Passivity-Based Control

The passivity and linearity-in-the-parameters properties of the robot dynamics can be exploited to design robust and adaptive controllers for manipulators as follows. We define the control input u as

$$u = \hat{M}(\theta)a + \hat{C}(\theta, \dot{\theta})v + \hat{g}(\theta) - Kr \quad (24)$$

where the quantities v , a , and r are given as

$$\begin{aligned} v &= \dot{\theta}^r - \Lambda \tilde{\theta} ; \quad a = \dot{v} = \ddot{\theta}^r - \Lambda \dot{\tilde{\theta}} ; \\ r &= \dot{\theta} - v = \dot{\tilde{\theta}} + \Lambda \tilde{\theta} \end{aligned}$$

and K is a diagonal matrix of positive gains. Using the linearity-in-the-parameters property, the closed-loop system can be written as

$$M(\theta)\dot{r} + C(\theta, \dot{\theta})r + Kr = Y(\theta, \dot{\theta}, a, v)(\hat{\Phi} - \Phi) \quad (25)$$

In the robust passivity-based approach, the term $\hat{\Phi}$ is chosen as $\hat{\Phi} = \Phi_0 + \delta\Phi$ where Φ_0 is a fixed nominal parameter vector and $\delta\Phi$ is an additional control term. The additional term $\delta\Phi$ can be designed according to

$$\delta\Phi = \begin{cases} -\rho \frac{Y^T r}{\|Y^T r\|} & ; \text{ if } \|Y^T r\| > \epsilon \\ -\frac{\rho}{\epsilon} Y^T r & ; \text{ if } \|Y^T r\| \leq \epsilon \end{cases}$$

where ρ is a (constant) bound on the parameter uncertainty. Uniform ultimate boundedness of the tracking errors follows using the Lyapunov function

$$V = \frac{1}{2} r^T M(\theta) r + \tilde{\theta}^T \Lambda K \tilde{\theta}$$

where, as before, the size of the ultimate boundedness set depends on the parameter ϵ .

In the adaptive version of this approach, we consider again the control law (24) and the resulting closed-loop system

$$M(\theta)\dot{r} + C(\theta, \dot{\theta})r + Kr = Y(\hat{\Phi} - \Phi)$$

In this case, the term $\hat{\Phi}$ is taken as the output of an estimator

$$\dot{\hat{\Phi}} = -\Gamma^{-1} Y^T(\theta, \dot{\theta}, a, v) r \quad (26)$$

The Lyapunov function

$$V = \frac{1}{2} r^T M(\theta) r + \tilde{\theta}^T \Lambda K \tilde{\theta} + \frac{1}{2} \tilde{\Phi}^T \Gamma \tilde{\Phi}$$

can be used to show global convergence of the tracking errors to zero and boundedness of the parameter estimates (Spong et al. 2006).

Summary and Future Directions

We have discussed the commonly applied methods of PID control, feedback linearization control, and passivity-based control for motion control of robot manipulators. There is a large and relatively mature body of literature on these methods, and in fact, the material here is now contained in standard textbooks in robotics, such as (Spong et al. 2006).

Future directions in robot motion control include the full integration of vision, force, and position feedback, cooperative control of multiple arms, and advances in machine learning and human-robot interaction. Direct control of

robots through brain-machine interfaces is also an active area of research and will enable new areas of applications such as medical assistive robots.

Cross-References

- [Adaptive Control: Overview](#)
- [Classical Frequency-Domain Design Methods](#)
- [Feedback Linearization of Nonlinear Systems](#)
- [Passivity-Based Control](#)
- [PID Control](#)

Recommended Reading

Many of the fundamental theoretical problems in motion control of robot manipulators were solved during an intense period of research from about the mid-1980s until the early-1990s during which time researchers first began to exploit the structural properties of manipulator dynamics such as feedback linearizability, skew symmetry and passivity, multiple time-scale behavior, and other properties. For a more advanced treatment of some of these topics, the reader is referred to Spong et al. (1992) and Canudas de Wit et al. (1996). A survey of robust control of robots is found in Abdallah et al. (1991). The passivity-based robust control result here is due to Spong (1992). The first results in passivity-based adaptive control of manipulators were in Horowitz and Tomizuka (1986) and Slotine and Li (1987). The Lyapunov stability proof of passivity-based adaptive control is due to Spong et al. (1990). A unifying treatment of adaptive manipulator control from a passivity perspective was presented in Ortega and Spong (1989).

Bibliography

- Abdallah C, Dawson DM, Dorato P, Jamshidi M (1991) A survey of robust control of rigid robots. *IEEE Control Syst Mag* 11(2):24–30
- Canudas de Wit C et al (1996) *Theory of robot control*. Springer, Berlin
- Horowitz R, Tomizuka M (1986) An adaptive control scheme for mechanical manipulators – compensation

- of nonlinearities and decoupling control. *Trans ASME J Dyn Syst Meas Control* 108:127–135
- Hunt LR, Su R, Meyer G (1983) Design for multi-input nonlinear systems. In: Brockett RW et al (eds) *Differential geometric control theory*. Birkhauser, Boston/Basel/Stuttgart, pp 268–298
- Lavelle SM (2006) *Planning algorithms*. Cambridge University Press, Cambridge
- Markiewicz BR (1973) Analysis of the computed torque drive method and comparison with conventional position servo for a computer-controlled manipulator. NASA-JPL Technical Memo, pp 33–61
- Ortega R, Spong MW (1989) Adaptive motion control of rigid robots: a tutorial. *Automatica* 25(6):877–888
- Siciliano B, Sciavicco L, Villani L, Oriolo G (2010) *Robotics: modeling, planning, and control*. Springer, London
- Slotine J-JE, Li W (1987) On the adaptive control of robot manipulators. *Int J Robot Res* 6(3):147–157
- Spong MW (1987) Modeling and control of elastic joint robots. *Trans ASME J Dyn Syst Meas Control* 109:310–319
- Spong MW (1992) On the robust control of robot manipulators. *IEEE Trans Autom Control* AC-37(11):1782–1786
- Spong MW, Vidyasagar M (1987) Robust linear compensator design for nonlinear robotic control. *IEEE J Robot Autom* RA 3(4): 345–350
- Spong MW, Ortega R, Kelly R (1990) Comments on ‘adaptive manipulator control: a case study’. *IEEE Trans Autom Control* 35:761–762
- Spong MW, Lewis FL, Abdallah CT (1992) *Robot control: dynamics, motion planning, and analysis*. IEEE, New York
- Spong MW, Hutchinson S, Vidyasagar M (2006) *Robot modeling and control*. Wiley, New York
- Tarn TJ, Bejczy AK, Isidori A, Chen Y (1984) Nonlinear feedback in robot arm control. In: *Proceedings of the IEEE conference on decision and control*, Las Vegas, pp 736–751

Robot Teleoperation

Claudio Melchiorri
Dipartimento di Ingegneria dell’Energia
Elettrica e dell’Informazione, Alma Mater
Studiorum Università di Bologna, Bologna, Italy

Abstract

Robots may allow human beings to physically interact with remote objects and environments. This possibility is known as *robot teleoper-*

ation and permits to operate in conditions or environments dangerous for human operators. Although teleoperation was among the first developments in robotics back in the 1950’s, still nowadays there are important and difficult challenges for researchers and scientists, showing the intrinsic difficulties of this fascinating field of robotics.

Keywords

Bilateral control · Force reflection · Robot teleoperation · Time delay

Introduction

A robotic teleoperation system allows to reproduce the actions of a human operator and to interact physically with objects and environments placed at a distance. This possibility has always attracted the human being, and telemanipulation has been one of the first fields to be developed in robotics: the first modern applications of this type of technology are dated back to the 1940s and the early 1950s for handling radioactive material (Goertz and Thompson 1954), for underwater and space applications (Martin and Kuban 1985; Vertut and Coiffet 1986), and for human prostheses (Kobriniskii 1960). For an overview on applications, see Sheridan (1992), Hokayem and Spong (2006), Ferre et al. (2007) and the related references. Nevertheless, despite the research interest and the many existing devices, many challenging problems have still to be fully solved both from the technical and control point of view.

In these notes, an overview on robot teleoperation is presented. In particular, the following points are illustrated:

- General description of a telemanipulation system and of its key components: the “*master*,” the “*slave*,” and the “*communication channel*”
- Overview on applications and existing devices
- Some control techniques for telemanipulation systems: “traditional” force reflection, shared compliance control, Passivity-based control, predictive control, four-channel architecture

General Description of a Telemanipulation System

A telemanipulator is a complex mechatronic system in which the main elements are a *master* (or local) and a *slave* (or remote) device, interconnected by a *communication channel*. The overall system is interfaced on one side (the master) with a human operator and on the other (the slave) with the environment: see Fig. 1.

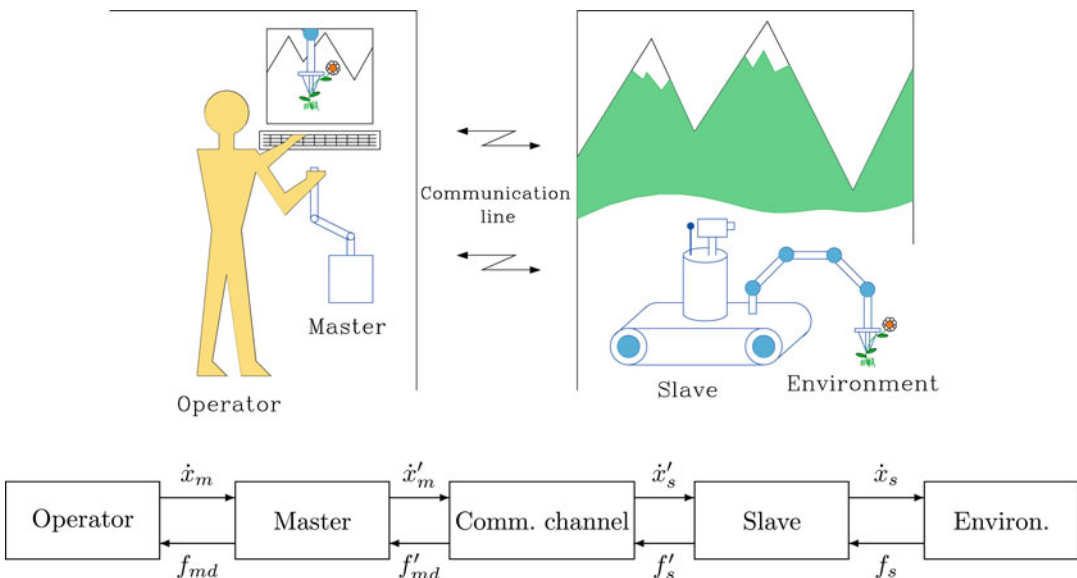
Both the master and slave devices have a local controller, with a hardware/software complexity that can be quite different depending on the system and task to be executed. Key features of this type of devices, usually not present in a typical robotic manipulation system, are:

1. A human operator is involved in the loop for the (high-level) control of the task execution.
2. It is necessary to provide to the operator, possibly in real time, data related to the task. This implies the presence of a suitable user interface and the selection of proper signals transmitted to the operator. These signals, e.g., related to forces applied to the environment, relevant positions of the slave, graphical video

data, and tactile or acoustic information, have strong implications on the control properties and performances of the system.

3. A communication channel is present between the master and the slave. This channel may represent a source of problems when time-delays are present since, as well known from the control theory, delays in a feedback loop may generate instability. Problems of this type, firstly observed in a force feedback scheme in 1965 (Ferrel 1966), arise, for example, in underwater or space operations. Note that even time-delays of the order of the tenth of a second may create instability problems.

In the block diagram of Fig. 1, some implicit choices have been made. The operator specifies a desired velocity ($\dot{x}_m$) to be applied to the environment through the master, the communication channel, and the slave and receives back a force signal (f_{md}). In the figure, the flow of the signals could be reversed as well, letting the operator specify a force to the environment and receiving back a velocity information. This is equivalent to reversing the roles of the master and slave



Robot Teleoperation, Fig. 1 A telemanipulation system and its block scheme representation. Subscripts m and s refer to variables at the master and slave site, respectively

devices. When this operation is possible, the teleoperation system is defined *bilaterally controlled* (Bejczy and Handlykken 1981).

One of the goals of the control system is to have, in steady state, the slave velocity equal to the master velocity, i.e., $\dot{x}_s = \dot{x}_m$, and similarly for the forces, $f_{md} = f_s$. When this is accomplished, the teleoperator is defined *transparent* (Lawrence 1993).

In this general framework, the main features of the components of a telemanipulator are the following.

The Master

The master, or local system, is the interface through which the operator specifies commands to the whole device. Typical features of the master are:

- Capability of assigning tasks to the slave and providing the operator with relevant information about the task development. In fact, an important feature of the master is its capability of providing the operator with the *telepresence*, i.e., the sensation of being in some manner involved with task execution. In this respect, several solutions have been adopted, varying from joysticks and/or consoles (Hirzinger et al. 1992) to exoskeletons (Smith et al. 1992; Bergamasco et al. 2007) and so on. In these devices, different types of signals may be reflected to the operator, from simple graphical data to full kinetostatic information.
- Capability of acquiring and processing data from both the operator and the slave. Typical elaborations are filtering, prediction, delay compensation, modeling of remote and local dynamics, and so on.

The Slave

The slave, or remote system, is the part of the teleoperator which directly interacts with the environment for task execution. Requirements similar to the master may be specified for the slave system:

- A robotic system for the interaction with the environment and the execution of the task planned by the operator. This part, usually provided with autonomous features, has to be in some way customized to operate in particular environments, e.g., submarine, outer space, and nuclear areas. Note that the kinematics and the dynamics of the remote manipulator may be different from those of the local one (when present), originating several problems when telepresence is needed for task execution (Colgate 1993).
- Signal acquisition and processing. Sensory capability is a main requirement for the slave device, which is often equipped with video cameras, force/tactile sensors, proximity sensors, and so on.
- Capability of data processing. Also the remote site must be able to elaborate the information needed for task execution. In fact, besides other considerations, the destabilizing effects caused by communication delays and/or restricted bandwidths of transmission must be compensated locally, providing the slave system with a certain degree of autonomy.

The Communication Line

The communication line represents the link between the master and slave sites. Different platforms may be used for this purpose, from radio connections by means of satellites to cables for underwater operations. The main drawback that can be introduced by this element is a delay, due both to a physical delay in the transmission line (e.g., in a long satellite communication) and to limited bandwidth of the hardware. This delay, that sometimes is not even constant, can cause noticeable instability problems if proper compensating actions are not taken.

An Overview on Applications

Use of telemanipulators, in the broader sense of the terminology, may be found in a number of different applications developed since the early

1950s. First examples of these devices have been designed and realized for operations in radioactive environments and for human limb prostheses. At the moment, this type of technology is applied in a number of different fields: space, underwater, medicine, hazardous environments, security, simulators, and so on.

Space Applications

Robotics is used in space for exploration, scientific experiments, and commercial activities. Main reasons of space telerobotics are the high costs and the hostile environment for human beings. For many years, the main example of teleoperation in space was applications in space shuttle activities where the operators had a direct control of the task executed by the manipulator. Nowadays, an important application of robot technology is for planetary missions, where autonomous telerobots are required and the operator has only a supervisory control of the task. Main directions of current research activity for space robotics are the development of arms for both intra-vehicular and extravehicular activities, free-flying platforms, and planetary rovers.

Among the most known examples of robot arms for space one can list the Canadian Remote Manipulator System (RMS), installed on the US space shuttles. The 6 degree-of-freedom (dof) arm, built by the Canadian firm SPAR, had a flexible, 15 m long structure and was capable of executing preprogrammed and/or teleoperated tasks. Five arms have been built, working on space shuttles from 1981 to 2011. Since 2001 the Canadarm 2 is used on the ISS (International Space Station). This 7 dof, 17.6 m long arm is used for assembly and maintenance purposes.

Concerning planetary exploration, a first successful space telerobotic program has been the Mars Viking Program, which performed scientific experiments on Mars in 1976. More recently, NASA has sent to Mars the rovers Sojourner in 1997 (working for about 3 months) and Spirit and Opportunity, which arrived in 2004. Opportunity is still working (January 2014), see <http://marsrovers.jpl.nasa.gov/home/index.html>. New missions on Mars with other,

more complex, rovers are currently planned by NASA.

With the current technological possibilities, further substantial developments in this field are slowed down by the large amount of money and time required to guarantee a successful mission. However, relevant technical problems still exist due to reliability requirements, weight constraints, hostile environments and communication time-delays (ranging from 1 s in earth orbits to 4–40 min or more for planetary missions).

Underwater

After the first successful military applications of underwater telerobotics (in 1966 the US Navy's CURV – Cable-controlled Underwater Recovery Vehicle – was successfully employed to retrieve a nuclear bomb from the ocean), extensive use of ROVs (remote operated vehicles) has started in the 1980s for offshore operations for oil/gas industry. At the moment, underwater telerobotics is mainly used for business, military missions, and scientific expeditions. Telerobotic (autonomous) tasks are usually limited to small routine tasks rather than complete activities, for example, simple tool switching operations, repetitive bolt/nut screwing, and piloting to new locations. First examples of underwater teleoperation were mainly based on manned submersibles, either free swimming or connected to a surface ship, and with teleoperated arms on the outer structure. In more recent operations, human operators remotely control the submersibles by long fiber-optic cables for data communication, increasing the costs and complexity of the missions.

Probably, the most important users are in the business field, where it is more convenient to use teleoperated devices rather than human divers to perform inspections and repairs on deep sea equipment. The main users of telesubmersibles are the oil and communication (telephone) industries, where underwater pipes and cables require routine operations. The scientific community uses this technology for marine biological, geological, and archeological missions, while the military have used telerobotics in many salvage operations, such as plane or watercraft recovery.

The conditions of the water environment, e.g., the high pressures, the poor visibility, and the communication difficulties, cause the major problems in this field. In order to solve the problems due to the high pressure, a very robust mechanical structure and (typically) hydraulic actuators are employed. On the other hand, vision problems are not so easily solved, being related to several factors of the environment. External lighting is necessary, and other technologies (e.g., sonar) are sometimes used. Computer graphic simulation may help the user during task execution in partially known environments. For references, see, e.g., Ridao et al. (2007).

Medical Telerobotics

Several teleoperated devices are found in the medical field. In fact, robotic manipulators are used to perform surgery, diagnose illnesses or injuries, help impaired people, and train specialized medical personnel.

Robotic systems of different complexities have been developed since the 1950s for aid to impaired people. Among the most common systems are automated wheelchairs, controlled by voice or by joysticks for hand, mouth, eye, or head movements.

At the moment, there is a relevant interest in applying teleoperated devices in microsurgery operations, e.g., eye surgery, where small precise movements are needed. The movements of the operator are scaled down by the mechanism so that very fine operations can be performed while maintaining a suitable telepresence effect. Another important class of surgical process consists of the so-called minimally invasive procedures. In this case, the surgeon operates through small insertions using thin medical instruments and small video cameras. The increased difficulties for the surgeon are partially compensated by computers, which are used to create virtual environments where the use of telepresence plays a fundamental role.

A very attractive application is the use of telemanipulators in remote surgery operations. Tele-diagnosis may also broaden the range of a single doctor by allowing to exam a patient visually or

viewing records on a computer interface. Finally, telepresence is becoming very important for the instruction of specialized doctors and to perform rehearsals before the actual operation.

Security

Applications in this area aim to employ telerobotic devices for the protection of persons and properties. Most systems used in this area are teleoperated devices since these tasks require decision capabilities and intelligence levels not currently possible for machines, although the use of autonomous systems is more and more frequent.

In the area of security, robots may be used for patrolling buildings and for protection purposes. These devices can either be autonomous or teleoperated. Military applications adopt principally teleoperation, mainly for locating enemies or dangerous equipment without direct risk for human personnel. Unmanned aeroplanes or teleoperated devices for the detection and destruction of mines or bombs are well-known examples of this technology. Teleoperation is also used for fire extinguishing, in order to spray water or chemical agents with remotely operated vehicles.

Telerobotics in Hazardous Environments

Robots may substitute human beings for operations in hazardous environments; as a matter of fact, nuclear industry was the first important user of modern teleoperating devices. Telerobotics is applied in several nuclear or chemical plants and also for military applications (e.g., for building military equipment and arms) in a variety of tasks. Besides direct handling of radioactive or chemical material, robots are used in waste cleanup/disposal and plant inspection. Ammunition disposal also makes use of telerobotic machines.

Telerobotics in Mining and Other Industries

Besides the typical use of robots in a number of industrial applications (assembly, welding, painting, and so on), other applications of robotic

systems in nonconventional production processes have been developed, for example, in mines, constructions, agriculture, warehousing, and many other activities.

Use of telemanipulators for mining applications, despite the relevant motivations such as high costs and risks of human work, finds difficulties and limitations caused by the particular environment and the relevant level of autonomy requested to operate in mines. As a matter of fact, the mining industry has only recently started to experiment teleoperated devices; see e.g., Duff et al. (2010). These machines are being developed to perform frame wall building, structure testing, hole drilling, wall blasting, mine digging, and so on.

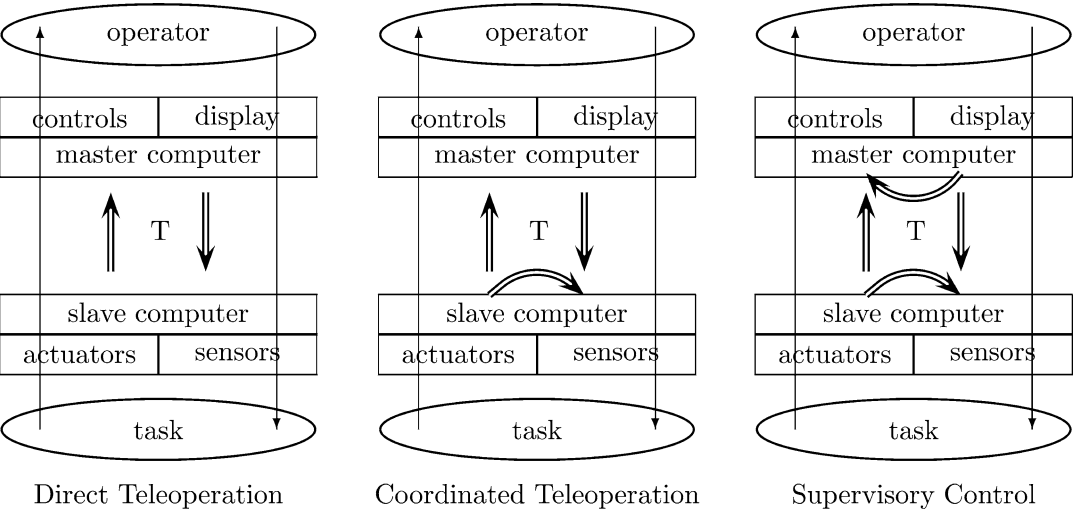
In construction tasks, not considering that all construction/destruction machinery controlled by a human can be regarded as examples of teleoperators (e.g., cranes and front-end loaders), applications of real telerobotic systems are not so numerous because of the unstructured environments and the nonrepetitive tasks. Current work in this area concerns the development of machines for earth movement, construction of structures, building window washing, bridge inspection and maintenance, and power line repair.

The Control Problem

For the development of a reliable teleoperation system, providing force feedback to the user, the problems caused by the interaction of the robotic device with the environment and the possible time-delays caused by the communication channel have to be properly considered and solved.

In telemanipulation without either force feedback to the operator or a local compliance control, the remote manipulator is strictly controlled according to the master position signal. As a consequence, the system results in being stiff, and errors between the master and slave positions may cause excessive and undesired contact forces.

In bilateral telemanipulation, it has been proven that a profitable manner for increasing system performances (e.g., in terms of task completion time, total contact time, and cumulative contact force) is to reflect back to the operator information about the force applied to the environment. On the other hand, it results that the force reflection gain, that gives to the operator the feeling of the interaction, destabilizes the system, especially when time-delays are present.



Robot Teleoperation, Fig. 2 Possible structures of bilateral control schemes for robotic teleoperation

Control schemes for robotic teleoperation devices can be classified according to the general structures reported in Fig. 2, showing the *direct teleoperation*, the *coordinated teleoperation*, and the *supervisory control* schemes. In the direct teleoperation scheme, possible only for negligible time delays, the operator has direct control of the slave robot and receives feedback in real time. In the coordinated teleoperation scheme, the operator still controls the remote robot, but low-level control loops in the slave system are present because time delays do not allow the operator to control directly the actuators. In the supervisory control scheme, the remote site has more autonomy and task execution is controlled locally, while the operator gives mainly high-level commands and acts as a supervisor. A local loop is present also at the master side, indicating the presence of (usually) a model (graphical, mathematical, etc.) of the slave site to improve performances in case of large time delays.

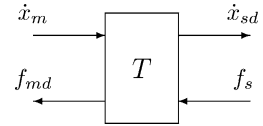
Some of the main control architectures for teleoperation devices presented in literature to deal with the problems of time-delay and force reflection are now briefly described and commented. The considered architectures are the “*traditional*” *force reflection*, the *shared compliance control*, the *passivity-based teleoperation*, the *predictive control*, and the *four-channel scheme*. However, many other control schemes have been presented in the literature; see, e.g., Arcara and Melchiorri (2002), Hirche et al. (2007), and therefore what is presented here is a brief, though significant, overview in order to focus on the major problems encountered in this field and on some of the approaches for their solutions.

Traditional Force Reflection Teleoperation

The simplest manner of transmitting the remote force to the operator is to reflect it directly, without any particular elaboration, as shown in Fig. 3. The resulting transmission equations are

$$\begin{cases} f_{md}(t) = f_s(t - T) \\ \dot{x}_{sd}(t) = \dot{x}_m(t - T) \end{cases} \quad (1)$$

Robot Teleoperation, Fig. 3 The “traditional force reflection” transmission scheme



where T is the time-delay introduced by the communication network and subscript d indicates the desired set point for the master (m) and slave (s) controllers.

This technique presents relevant instability problems due to time-delays. As a matter of fact, it is possible to verify that the communication channel does not present strictly passive properties, even for limited bandwidths of the input signals $\dot{x}_m$ and f_s . This result is valid also considering an attenuation between f_{md} and f_s , i.e., introducing a force reflection gain $G_{fr} < 1.0$ in (1) and computing therefore $f_{md}(t) = G_{fr} f_s(t - T)$. The attenuation reduces the telepresence sensation and degrades the performances, but still does not cause a passive (then stable) network. The nonpassive channel has the global effect of introducing in the overall system energy flows that, if not properly reduced by the local controllers, contribute to destabilize the telemanipulator.

The dynamics of the overall system may be described by the following two sets of equations: the first taking into account the master dynamics and the force transmitted by the communication channel and the second including the slave dynamics, the position signals of the channel, and the local position controller:

$$\begin{cases} M_m \dot{x}_m(t) = -f_{md}(t) - B_m \dot{x}_m(t) - K_h x_m(t) \\ f_{md}(t) = f_s(t - T) \\ M_s \dot{x}_s(t) = f_s(t) - B_s \dot{x}_s(t) \\ \dot{x}_{sd}(t) = S_p \dot{x}_m(t - T) \\ f_s(t) = K_p [x_{sd}(t) - x_s(t)] \end{cases}$$

In the above equations, M_i and B_i , $i = m, s$, are masses and damping factors at the master

and slave sites, K_h represents the operator (simply modeled as a stiffness) and K_p the slave position controller. The gain S_p has been added, with respect to Eq. (1), in order to scale velocity variables between the two robotic systems.

It can be shown that this control scheme does not guarantee stability in the presence of time delays, although in practical applications stability may still be achieved for small time delays due to dissipation introduced by friction and the local controllers.

Shared Compliance Control

As previously mentioned, both the interactions of the robotic device with the environment and the effects of time-delays have to be considered in the definition of control strategies for telemanipulation systems. The *position-error based force reflection* scheme deals with both these effects (Kim et al. 1992). This scheme is based on the computation of the feedback signal f_{md} as a force proportional to the error between master and slave positions:

$$f_{md}(t) = G_{fr} [x_m(t) - x_s(t - T)]$$

This signal gives to the operator a sensation related to the difference between the postures of the robotic devices caused either by interactions or delays. Note that in this manner an elastic (proportional) element is introduced between the positions of the robots. This allows to obtain a stable behavior of the overall system compre-

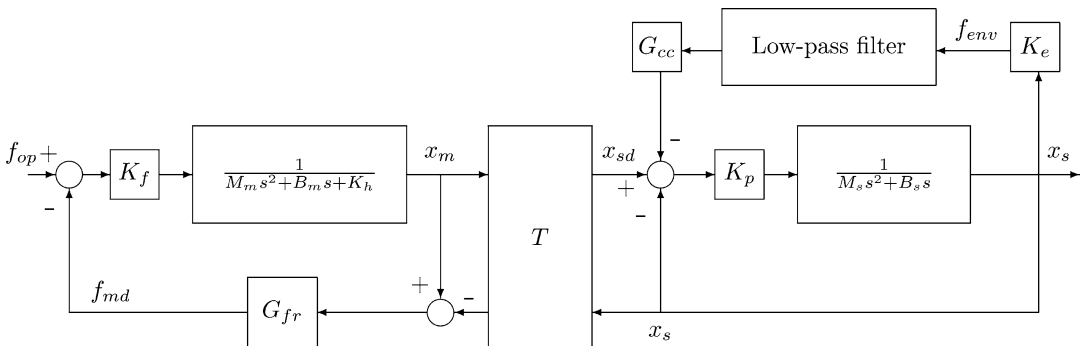
hensive of local controllers, at least for limited values of G_{fr} .

An additional feature for dealing with problems due to time-delays is the so-called *shared compliance control* (SCC). A local, autonomous force feedback is realized at the slave site in order to program active compliance and damping of the robotic device. This control action is important when compliance has to be realized between the (stiff) mechanical device and its environment and during the collision or contact phases. The overall control system is therefore based on sharing autonomous and human-driven control actions. A block diagram of the whole system, including the master and slave dynamics ($1/(M_ms^2 + B_ms + K_h)$ and $1/(M_ss^2 + B_ss)$ respectively), the force reflection gain (G_{fr}), the shared compliance controller (G_{cc}), and an environment model (K_e), is shown in Fig. 4.

For a given time-delay, the force reflection gain G_{fr} can be increased with respect to the traditional force reflection scheme. In any case, when the time-delay increases, the gain has to be correspondingly decreased to guarantee stability, i.e., the value of G_{fr} depends directly on the amount of time-delay. In fact, also this control scheme in general does not present passivity features, although it can be shown that it may be stable (for a limited range of time delays) with a proper choice of the control parameters.

Passivity-Based Teleoperation

A control scheme inspired by the passivity theory (Van der Schaft 2000) is now described. Basic



Robot Teleoperation, Fig. 4 Position-error based force reflection with SCC at the remote site

consideration is that the communication channel may represent, if proper actions are not taken, a non-passive element between the master and slave. With proper modifications, the transmission line presents passive properties, and therefore, the stability of the overall system may be achieved for any value of the time-delay T .

Lossless Transmission Line

Results of passivity and scattering theories can be used to show that in traditional force reflection teleoperation, Eq. (1), the instability of the overall system in presence of time-delays is caused by the non passive properties of the communication channel (Niemeyer and Slotine 1991). On the other hand, it has been shown that the definition of a communication network based on a lossless transmission line provides the system with passivity features for any time-delay (Anderson and Spong 1989), facilitating therefore the stability of the overall system.

For the definition of a lossless transmission line, it is convenient to refer, instead of the velocity and force variables $\dot{x}$, f at each port (see Fig. 3), to the equivalent *wave* variables u and v that are related to the passivity formalism and whose definition derives from the theory of electric circuits. By using these variables, it is possible to describe the power balance in a circuit as the difference of two positive terms which consider the input and the output power. In fact, by introducing the input wave $u = [u_m^T, u_s^T]^T$ and the output wave $v = [v_m^T, v_s^T]^T$, the power balance in the teleoperator can be expressed as

$$P = \frac{1}{2} (u^T u - v^T v) = f^T \dot{x} = [f_m^T, f_s^T] \begin{bmatrix} \dot{x}_m \\ -\dot{x}_s \end{bmatrix}$$

By considering a proper scaling factor b , defined as the *characteristic impedance* of the transmission line, the previous equation defines the following transformations between power and wave variables:

$$\begin{aligned} u_m &= \frac{1}{\sqrt{2b}}(f_m + b \dot{x}_m) & u_s &= \frac{1}{\sqrt{2b}}(f_s - b \dot{x}_s) \\ v_m &= \frac{1}{\sqrt{2b}}(f_m - b \dot{x}_m) & v_s &= \frac{1}{\sqrt{2b}}(f_s + b \dot{x}_s) \end{aligned}$$

The resulting network is described by

$$\begin{cases} f_{md}(t) = f_s(t - T) + b[\dot{x}_m(t) - \dot{x}_{sd}(t - T)] \\ \dot{x}_{sd}(t) = \dot{x}_m(t - T) + \frac{1}{b}[f_{md}(t - T) - f_s(t)] \end{cases}$$

In terms of wave variables, the passivity-based communication network is described as (see Fig. 5)

$$\begin{cases} f_{md}(t) = b \dot{x}_m(t) + \sqrt{2b} v_m(t) \\ \dot{x}_{sd}(t) = -\frac{1}{b}[f_s(t) - \sqrt{2b} v_s(t)] \end{cases}$$

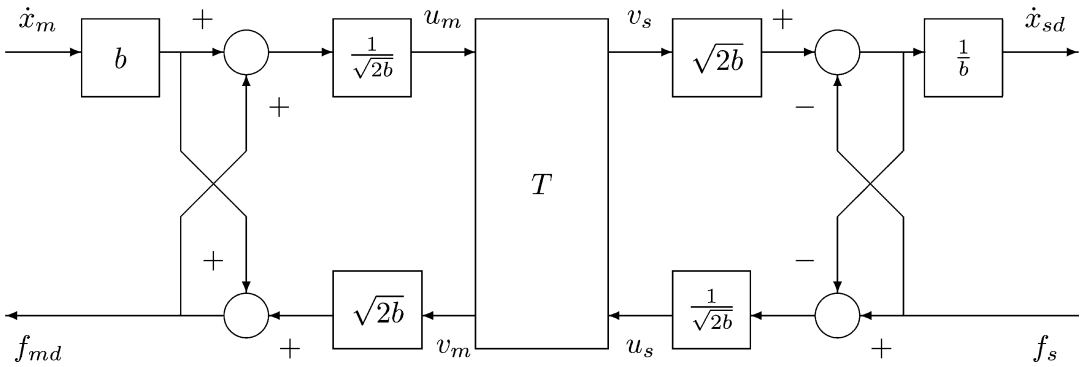
In analogy with electric networks, impedance adaptation should be added to both extremities of the transmission line, as described e.g., in Niemeyer and Slotine (1991).

Predictive Control

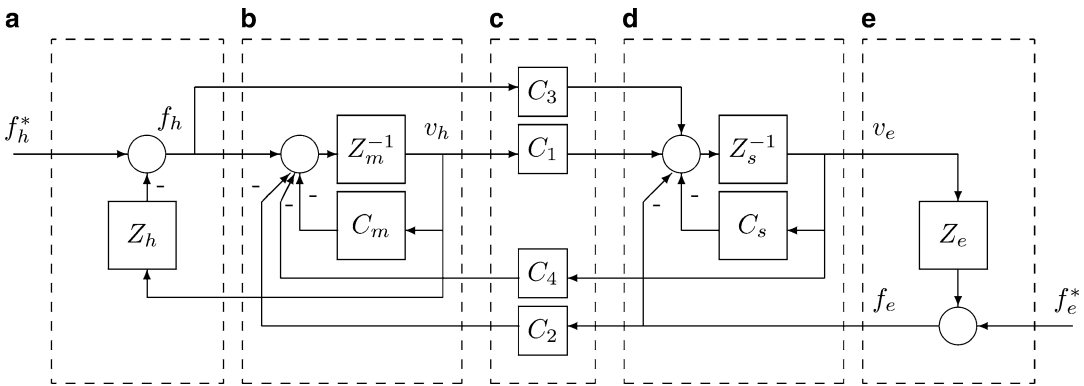
In a well-known example of space telerobotics, the ROTEX project (Hirzinger et al. 1992), the problems introduced by force feedback and time-delays have been solved in a different manner. In fact, in this case the force information is not transmitted to the operator, and an extensive use of graphic simulation and *telesensor programming* is made to help control of the task execution.

In particular, the *predictive display* technique (Sheridan 1992) has been employed for generating and extrapolating beforehand visual indications, such as cursors or wire frame models of the manipulator and its environment. These information are generated by the control system and assist the operator in driving the task execution more efficiently. In this case, a proper prediction algorithm has to be set on the basis of current initial conditions of the manipulator and, possibly, of current control variables.

In telerobotics, predictive displays have to be purposely designed in order to consider the prediction of motions of the manipulator. Usually, the task is graphically simulated in real time, without time-delay, exploiting a model of the remote environment and of the slave device. The operator can observe the task executed by the remote system on the screen, where a simulated copy (with $T = 0$ s) of the robotic device can



Robot Teleoperation, Fig. 5 Transmission line based on passivity



Robot Teleoperation, Fig. 6 Block scheme of the four-channel architecture. (a) The human operator. (b) Master controller. (c) Communication line. (d) Slave controller. (e) Environment

be superimposed on the real operating device in the scene of the remote site. In this manner, the operator may program appropriate actions for the interaction with the environment.

This type of task planning helps when a noticeable time-delay occurs. In fact, when operators deal with relevant time-delays (e.g., larger than 1 s), usually they operate with a “move and wait” strategy, conservatively specifying only small displacements to the remote robot. By using predictive display, the time required to execute complex tasks is greatly reduced. On the other hand, the operator has only visual information about the remote environment and the task execution.

Four-Channel Scheme

A generalization of the scheme of Fig. 1, the so-called four-channel architecture (Hirche et al. 2007; Lawrence 1993), is shown in Fig. 6. In this scheme, both the velocity and force signal of the master and slave are transmitted, and with a proper choice of the four blocks C_1, C_2, C_3 , and C_4 many design goals can be achieved, in particular concerning the stability and the transparency of the overall system. In particular, if $C_3 = C_4 = 0$, the standard velocity-force transmission scheme is obtained, while ideal transparency is achieved if $C_1 = Z_{cs}, C_2 = C_3 = I, C_4 = -Z_{cm}$. In the figure, the blocks Z_m and Z_s represent the master and the slave dynamics

(impedances), respectively, while C_m and C_s are the local master and slave controllers, f_h^* is an external force applied by the user, and f_e^* an exogenous force from the environment.

Summary and Future Directions

In these notes, an overview on telemanipulation has been presented with the aim of giving a general presentation of the impact of this area of robotics on both industry and research, of outlining typical problems encountered in dealing with remote manipulation systems, and of illustrating some approaches for their solution.

In this respect, it has to be pointed out that, besides the control schemes considered in these notes (purposely developed for telerobotic systems), many other schemes have been presented in the literature (see, e.g., Hokayem and Spong 2006, Ferre et al. 2007, and Arcara and Melchiorri 2002). More in general, however, a relevant literature exists, and important results have been presented from a methodological point of view to face control problems of time-delay systems: see for example, Gu et al. (2003).

There are, however, other important aspects of telemanipulation which, for space constraints, can only be mentioned here, such as the “impedance shaping” (typical in applications in which there is a relevant dynamic/mechanical difference between the master and slave mechanisms) or criteria for defining (and measuring) performance of teleoperator systems, such as the “time to completion,” criteria based on energy consumption, dexterity, and so on. Other interesting, and important, extensions are the possibility of controlling remote teams of robots cooperating for the execution of a common task (e.g., for aerial inspections, transport of heavy loads, etc.).

Future developments of robotic teleoperation systems will deal with the technological improvements of the user interface, giving to the operator more “realistic” feedback of the remote environment, the application of this type of technology to more complex situations, and the use of multi-robot systems controlled either by one or more

cooperating users. Control will play in any case a fundamental role in these scenarios.

Cross-References

- [Advanced Manipulation for Underwater Sampling](#)
- [Control of Linear Systems with Delays](#)
- [Disaster Response Robot](#)
- [Force Control in Robotics](#)
- [Model Predictive Control in Practice](#)
- [Redundant Robots](#)
- [Robot Visual Control](#)

Recommended Reading

Introductory and historical perspectives of telemanipulation, along with descriptions of several interesting applications of this technology, may be found in Ferre et al. (2007), Hokayem and Spong (2006), Sheridan (1992), and Vertut and Coiffet (1986). Specific applications, e.g., space, underwater, medical, and hazardous environment, are described in Duff et al. (2010), Hirzinger et al. (1992), <http://marsrovers.jpl.nasa.gov/home/index.html>, and Ridao et al. (2007). Some of the main control schemes specifically developed for this type of robotic devices are reported in Anderson and Spong (1989), Arcara and Melchiorri (2002), Colgate (1993), Hirche et al. (2007), Kim et al. (1992), and Niemeyer and Slotine (1991), while some basic background material on control theory is available in Gu et al. (2003) and Van der Schaft (2000).

Bibliography

- Anderson JA, Spong MW (1989) Bilateral control of teleoperators with time delay. *IEEE Trans Autom Control* 34(5):494–501
- Arcara P, Melchiorri C (2002) Control schemes for teleoperation with time delay: a comparative study. *Int J Robot Auton Syst* 38(1):49–64
- Bejczy AK, Handlykken M (1981) Generalization of bilateral force-reflecting control of manipulators.

- In: Proceedings of the 4th Rom-An-Sy, Warsaw, pp 242–255
- Bergamasco M, Frisoli A, Avizzano CA (2007) Exoskeletons as man-machine interface systems for teleoperation and interaction in virtual environments. In: Ferre M, Buss M, Aracil R, Melchiorri C, Balaguer C (eds) *Advances in telerobotics*. Springer, Berlin
- Colgate JE (1993) Robust impedance shaping telemanipulation. *IEEE Trans Robot Autom* 9(4):374–384
- Duff E, Caris C, Bonchis A, Taylor K, Gunn C, Adcock M (2010) The development of a telerobotic rock breaker. In: Howard A, Iagnemma K, Kelly A (eds) *Field and service robotics*. Springer tracts in advanced robotics, vol 62. Springer, Berlin/Heidelberg, pp 411–420
- Ferre M, Buss M, Aracil R, Melchiorri C, Balaguer C (eds) (2007) *Advances in telerobotics*. Springer, Berlin
- Ferrel WR (1966) Delayed force feedback. *IEEE Trans Hum Factors Electr HFE*-8:449–455
- Goertz RC, Thompson RC (1954) Electronically controlled manipulators. *Nucleonics* 12(11): 46–47
- Gu K, Kharitonov V, Chen J (2003) *Stability of time-delay systems*. Birkhauser, Boston
- Hirche S, Ferre M, Barrio J, Melchiorri C, Buss M (2007) Bilateral control architectures for telerobotics. In: Ferre M, Buss M, Aracil R, Melchiorri C, Balaguer C (eds) *Advances in telerobotics*. Springer, Berlin
- Hirzinger G, Grunwald G, Brunner B, Heindl J (1992) A sensor-based telerobotic system for the space robot experiment ROTEX. In: Workshop on teleoperation and orbital robotics, 1992 IEEE international conference on robotics and automation, Nice
- Hokayem PF, Spong MW (2006) Bilateral teleoperation: an historical survey. *Automatica* 42:2035–2057
<http://marsrovers.jpl.nasa.gov/home/index.html>
- Kim WS, Hannaford B, Bejczy AK (1992) Force-reflecting and shared compliant control in operating telemanipulators with time delay. *IEEE Trans Robot Autom* 8(2):176–185
- Kobrinskii A (1960) The thought control the machine: development of a bioelectric prosthesis. In: Proceedings of the 1st IFAC world congress on automatic control, Moscow
- Lawrence DA, (1993) Stability and transparency in bilateral teleoperation. *IEEE Trans Robot Autom* 9(5): 624–637
- Martin HL, Kuban DP (1985) Teleoperated robotics in hostile environments. *Robotics International of SME*, Dearborn
- Niemeyer G, Slotine JJE (1991) Stable adaptive teleoperation. *IEEE J Ocean Eng* 16(1):152–162
- Ridao P, Carreras M, Hernandez E, Palomeras N (2007) Underwater telerobotics for collaborative research. In: Ferre M, Buss M, Aracil R, Melchiorri C, Balaguer C (eds) *Advances in telerobotics*. Springer, Berlin
- Sheridan TB (1992) *Telerobotics, automation, and human supervisory control*. MIT, Cambridge
- Smith FM, Backman DK, Jacobsen SC (1992) Telerobotic manipulator for hazardous environments. *J Robot Syst* 9(2):251–260

- Van der Schaft AJ (2000) *L2-gain and passivity techniques in nonlinear control*. Springer communications and control engineering series, 2nd edn. Springer, London
- Vertut J, Coiffet P (1986) *Robot technology*, vol 3A: teleoperation and robotics: evolution and development. Prentice-Hall

Robot Visual Control

François Chaumette

Inria, Univ Rennes, CNRS, IRISA, Rennes, France

Abstract

This entry presents the basic concepts of vision-based control, that is, the use of visual data to control the motions of a robotics system. It details the modeling steps allowing to achieve a visual task, the list of possible visual features that can be used as input of the control scheme, the design of basic kinematics control schemes, and hints toward more advanced approaches. Applications are also described.

Keywords

Robot control · Robot vision · Visual servoing · Jacobian

Introduction

Basically, visual control, also named visual servoing, consists in using the data provided by one or several cameras as input of real-time closed-loop control schemes for controlling the motions of a dynamic system so that it achieves a task specified by visual constraints (Hutchinson et al. 1996; Chaumette et al. 2016). Such systems are usually robot arms or mobile robots, but can also be virtual robots, or even a virtual camera. A large variety of positioning tasks, or target tracking tasks, can be considered by controlling from one to all the degrees of freedom (DoF) of the system (Marchand et al.

2005). Whatever the sensor configuration, which can vary from one on-board camera located on the robot end-effector to several free-standing cameras, a set of visual features has to be selected at best from the image measurements available, allowing to control the desired DoF. A control law has then to be designed so that these visual features reach a desired value, defining a correct achievement of the task. A desired planned trajectory can also be tracked. The control principle is to regulate the error between the current and desired values of the visual features to zero or, in other terms, to minimize an objective function from which Lyapunov-based stability analysis can be performed. With a vision sensor providing 2D measurements, potential visual features are numerous, since 2D data (coordinates of particular points in the image, parameters related to geometrical shapes, intensity levels of set of pixels, etc.) as well as 3D data provided by a localization algorithm exploiting the extracted 2D measurements can be considered. It is also possible to combine 2D and 3D visual features to take the advantages of each approach while avoiding their respective drawbacks.

Typically, an iteration of the control scheme consists of the following successive steps:

- acquire an image;
- extract some useful image measurements;
- compute the current value of the visual features used as inputs of the control scheme;
- compute the error between the current and the desired values of the visual features;
- update the control outputs, which are usually the robot velocity, to regulate that error to zero, i.e., to minimize its norm.

For instance, for the first example depicted on Fig. 1, the image processing part consists in extracting and tracking the center of gravity of the moving people, the visual features are composed of the two Cartesian coordinates of this center of gravity, and the control scheme computes the camera pan and tilt velocities so that the center of gravity is as near as possible of

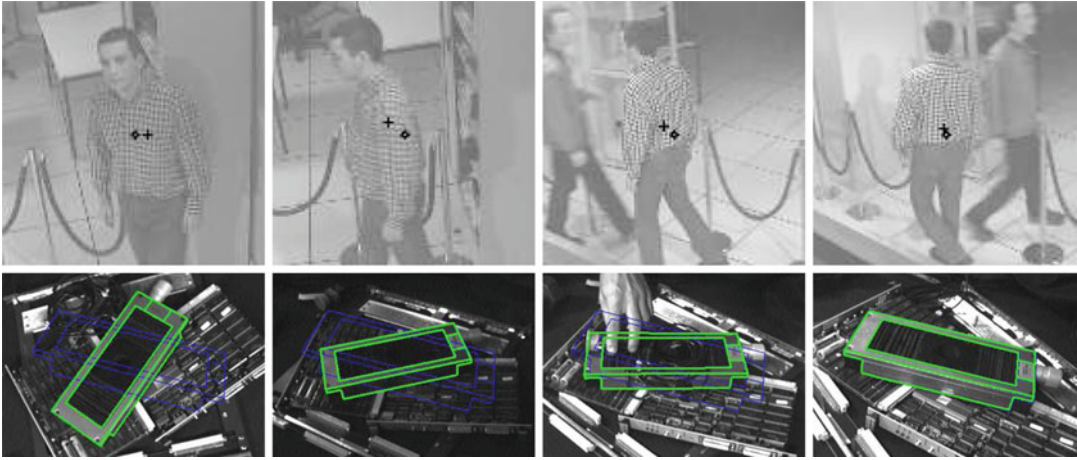
the image center despite the unknown motion of the people. In the second example where a camera mounted on a six DoF robot arm is considered, the image measurements are a set of segments that are tracked in the image sequence. From these measurements and the knowledge of the 3D object model, the pose from the camera to the object is estimated and used as visual features. The control scheme now computes the six components of the robot velocity so that this pose reaches a particular desired value corresponding to the object position depicted in blue on the images.

Basic Theory

Most if not all visual servoing tasks can be expressed as the regulation to zero of an error $e(t)$ defined by

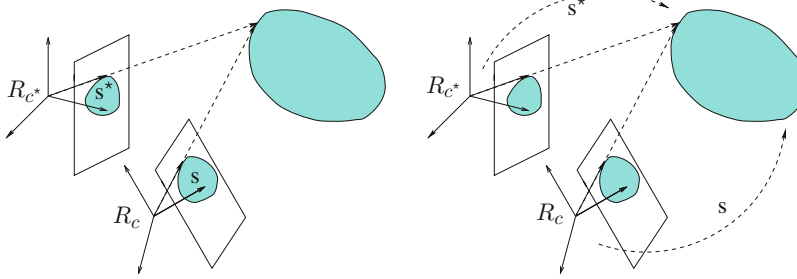
$$e(t) = s(m(r(t)), a) - s^*(t). \quad (1)$$

The parameters in (1) are defined as follows (Chaumette et al. 2016): the vector $m(r(t))$ is a set of image measurements (e.g., the image coordinates of points, or the area, the center of gravity, and other geometric characteristics of an object). These image measurements depend on the pose $r(t)$ between the camera and the environment, this pose varying with time t . They are used to compute a vector $s(m(r(t)), a)$ of visual features, in which a is a set of parameters that represent potential additional knowledge about the system (e.g., coarse camera intrinsic parameters or 3D model of objects). The vector $s^*(t)$ contains the desired value of the features, which can be either constant in the case of a fixed goal or varying if the task consists in following a specified trajectory. Visual servoing schemes mainly differ in the way that the visual features are designed. As represented in Fig. 2, the two most classical approaches are named image-based visual servoing (IBVS), in which s consists of a set of 2D parameters that are directly expressed in the image (Weiss et al. 1987; Espiau et al. 1992), and pose-based visual servoing (PBVS), in which s consists of a set of 3D parameters related



Robot Visual Control, Fig. 1 Few images acquired during two visual servoing tasks: on the top, pedestrian tracking using a pan-tilt camera; on the bottom, controlling the

6 degrees of freedom of an eye-in-hand system so that an object appears at a particular position in the image (shown in blue)



Robot Visual Control, Fig. 2 If the goal is to move the camera from frame R_c to the desired frame R_{c^*} , two main approaches are possible: IBVS on the left, where the

features s and s^* are expressed in the image, and PBVS on the right, where the features s and s^* are related to the pose between the camera and the observed object

to the pose between the camera and the target (Weiss et al. 1987; Wilson et al. 1996; Thuilot et al. 2002). In that case, the 3D parameters have to be estimated from the image measurements either through a pose estimation process using the knowledge of the 3D target model (Marchand et al. 2016) or through a triangulation process if a stereovision system is considered. Inside IBVS and PBVS approaches, many possibilities exist depending on the choice of the features. Each choice will induce a particular behavior of the system. There also exist hybrid approaches, named 2 1/2D visual servoing, which combine 2D and 3D parameters in s in order to benefit from the advantages of IBVS and PBVS while

avoiding their respective drawbacks (Malis and Chaumette 2000).

The Features Jacobian

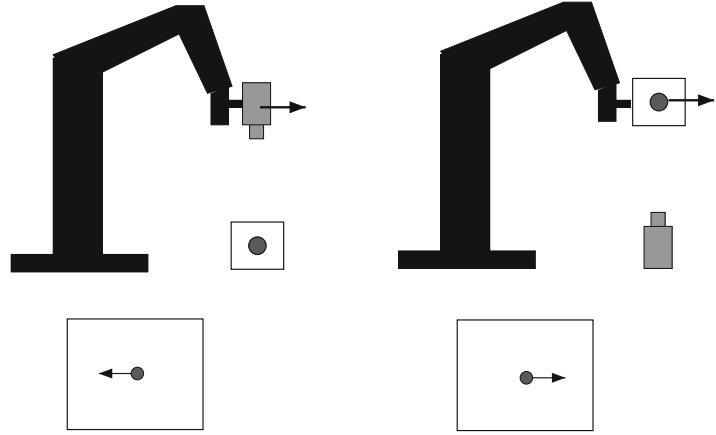
The design of the control scheme is based on the link between the time variation $\dot{s}$ of the features and the robot control inputs, which are usually the velocity $\dot{q}$ of the robot joints. This relation is given by

$$\dot{s} = J_s \dot{q} + \frac{\partial s}{\partial t} \quad (2)$$

where J_s is the features Jacobian matrix, defined from the equation above similarly as the classical robot Jacobian. For an eye-in-hand system (see the left part of Fig. 3), the term $\frac{\partial s}{\partial t}$, represents

Robot Visual Control,

Fig. 3 In visual servoing, the vision sensor can be mounted either near the robot end-effector (eye-in-hand configuration) or outside and observing the end-effector (eye-to-hand configuration). For the same robot motion, the motion produced in the image will be opposite from one configuration to the other



the time variation of s due to a potential object motion, while for an eye-to-hand system (see the right part of Fig. 3) it represents the time variation of s due to a potential sensor motion.

As for the features Jacobian, in the eye-in-hand configuration, it can be decomposed as (Chaumette et al. 2016)

$$J_s = L_s {}^cV_e J(q) \quad (3)$$

where:

- $J(q)$ is the robot Jacobian such that $\dot{v}_e = J(q)\dot{q}$ where v_e is the robot end-effector velocity;
- cV_e is the spatial motion transform matrix from the vision sensor to the end-effector. It is given by Khalil and Dombre (2002)

$${}^cV_e = \begin{bmatrix} {}^cR_e & [{}^ct_e]_{\times} {}^cR_e \\ \mathbf{0} & {}^cR_e \end{bmatrix} \quad (4)$$

where cR_e and ct_e are, respectively, the rotation matrix and the translation vector between the sensor frame and the end-effector frame and where $[{}^ct_e]_{\times}$ is the skew symmetric matrix associated to ct_e . Matrix cV_e is constant when the vision sensor is rigidly attached to the end-effector, which is usually the case. Thanks to the robustness of closed-loop control schemes with respect to calibration errors, a coarse approximation of cR_e and ct_e is generally sufficient in practice

to serve as a satisfactory estimation of cV_e to be injected in the control law. If needed, an accurate estimation is possible through classical hand-eye calibration methods (Tsai and Lenz 1989).

- L_s is the interaction matrix of s defined such that $\dot{s} = L_s v$ where v is the relative velocity between the camera and the environment.

In the eye-to-hand configuration, the features Jacobian J_s is composed of (Chaumette et al. 2016)

$$J_s = -L_s {}^cV_f {}^fV_e J(q) \quad (5)$$

where:

- fV_e is the spatial motion transform matrix from the robot reference frame to the end-effector frame. It is known from the robot kinematics model.
- cV_f is the spatial motion transform matrix from the camera frame to the reference frame. It is constant as long as the camera does not move. In that case, similarly as for the eye-in-hand configuration, a coarse approximation of cR_f and ct_f is usually sufficient.

The Interaction Matrix

A lot of works have concerned the modeling of various visual features s and the determination of the analytical form of their interaction matrix L_s . To give just an example, in the case of an image point with normalized Cartesian coordinates

$\mathbf{x} = (x, y)$ and whose 3D corresponding point has depth Z in the camera frame, the interaction matrix $\mathbf{L}_x$ of $\mathbf{x}$ is given by Espiau et al. (1992)

$$\mathbf{L}_x = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \quad (6)$$

where the three first columns contain the elements related to the three components of the translational velocity and where the three last

columns contain the elements related to the three components of the rotational velocity.

By just changing the parameters representing the same image point, that is, by using the cylindrical coordinates defined by $\boldsymbol{\gamma} = (\rho, \theta)$ with $\rho = \sqrt{x^2 + y^2}$ and $\theta = \text{Arctan}(y/x)$, the interaction matrix of these parameters has a completely different form (Iwatsuki and Okiyama 2005):

$$\mathbf{L}_\gamma = \begin{bmatrix} -\cos \theta / Z & -\sin \theta / Z & \rho / Z & (1 + \rho^2) \sin \theta & -(1 + \rho^2) \cos \theta & 0 \\ \sin \theta / (\rho Z) & -\cos \theta / (\rho Z) & 0 & \cos \theta / \rho & \sin \theta / \rho & -1 \end{bmatrix} \quad (7)$$

This implies that using the Cartesian coordinates or the cylindrical coordinates as visual features will induce a different behavior, that is, a different trajectory of the point in the image and, consequently, a different robot trajectory. The main objective in designing a visual servoing control scheme is thus to select the best set of visual features in terms of stability, global behavior (adequate trajectories in both the image plane and 3D space), and robustness to noise and to modeling and calibration errors from the task to be achieved, the environment observed, and the available image measurements. All these aspects can be studied from the interaction matrix of the potential visual features.

Currently, the analytical form of the interaction matrix is available for most basic features resulting from the perspective projection of simple geometrical primitives such as circles, spheres, and cylinders (Espiau et al. 1992). It is also available for image moments related to planar and almost-planar objects of any shape (Chaumette 2004), as well as for features selected from the epipolar geometry (Silveira and Malis 2012) and, of course, also for coordinates of 3D points, parameters of 3D geometrical primitives, and pose parameters, assuming these features are perfectly estimated.

In the recent years, following the seminal works of Nayar et al. (1996) and Deguchi (2000), a new trend has concerned the use of direct image content as input of the control scheme (Han et al. 2010; Collewet and Marchand 2011). The main

objective of these works is to avoid the extraction, tracking, and matching of geometrical measurements, such as points of interest or edges, so that the system is extremely accurate and robust with respect to image processing errors. The basic idea is to consider the intensity of a set of pixels as visual features ($\mathbf{s} = \mathbf{I}$). From the classical assumption in computer vision stating that the intensity level of a moving point does not change (i.e., $I(\mathbf{x}, t) = I(\mathbf{x} + d\mathbf{x}, t + dt)$), it is possible to determine the interaction matrix corresponding to the intensity level of a pixel:

$$\mathbf{L}_I = -\left[\frac{\partial I}{\partial x} \quad \frac{\partial I}{\partial y}\right] \mathbf{L}_x \quad (8)$$

where $\left[\frac{\partial I}{\partial x} \quad \frac{\partial I}{\partial y}\right]$ is the spatial gradient of the intensity along the x and y directions. Proceeding so leads to control a highly nonlinear system, with the drawback of a relatively small convergence domain, and, in general, not expected robot trajectory, plus the potential issue of robustness with respect to lighting variations. That is why the idea of direct photometric visual servoing has been expended either by considering other objective functions than $\|\mathbf{I} - \mathbf{I}^*\|$, such as the mutual information between the current and desired images (Dame and Marchand 2011), or other global image representations (Duflot et al. 2019; Crombez et al. 2019), or by designing photo-geometric visual features (Bakthavatchalam et al. 2018).

All the works mentioned above have considered a classical vision sensor modeled by a perspective projection. It is possible to generalize the approach to any sort of sensors, such as omnidirectional cameras (Hadj-Abdelkader et al. 2008; Caron et al. 2013), RGB-D sensors (Teulière and Marchand 2014), the coupling between a camera and structured light (Motyl et al. 1992; Pagès et al. 2006), and even 2D echographic probes (Mebarki et al. 2010). A large variety of visual features is thus available for many vision sensors.

Finally, methods also exist to estimate offline or online a numerical value of the interaction matrix, by using neural networks, for instance (Suh 1993; Wells et al. 1996), or the Broyden update (Hosoda and Asada 1994; Jägersand et al. 1997). These methods are useful when the analytical form of the interaction matrix cannot be determined, but any a priori analysis of the properties of the system is unfortunately impossible.

Control

Once the modeling step has been performed, the design of the control scheme can be quite simple. The most basic control scheme has the following form (Chaumette et al. 2016):

$$\dot{\mathbf{q}} = -\lambda \widehat{\mathbf{J}}_s^+ \mathbf{e} + \widehat{\mathbf{J}}_s^+ \frac{\partial \mathbf{s}^*}{\partial t} - \widehat{\mathbf{J}}_s^+ \frac{\partial \widehat{\mathbf{s}}}{\partial t} \quad (9)$$

where, in the first feedback term, $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ as defined in Eq. (1), λ is a positive (possibly varying) gain tuning the time-to-convergence of the system and $\widehat{\mathbf{J}}_s^+$ is the Moore-Penrose pseudoinverse of an approximation or an estimation of the features Jacobian. The exact value of all its elements is indeed generally unknown since it depends on the intrinsic and extrinsic camera parameters, as well as on some 3D parameters such as the depth of the point in Eqs. (6) and (7). Methods for estimating these 3D parameters exist, either using the knowledge of the robot motion (De Luca et al. 2008) or, up to a scalar factor, from partial pose estimation using the properties of the epipolar geometry between the current and the desired images (Malis and Chaumette 2000).

The second term of the control scheme anticipates for the variation of $\mathbf{s}^*$ in the case of a varying desired value. The third term compensates as much as possible a possible target motion in the eye-in-hand case and a possible camera motion in the eye-to-hand case. They are both null in the case of a fixed desired value and a motionless target or camera. They serve as feedforward terms for removing the tracking error in the other cases (Corke and Good 1996).

Following the Lyapunov theory, the stability of the system can be studied (Chaumette et al. 2016). Generally, visual servoing schemes can be demonstrated to be locally asymptotically stable (i.e., the robot will converge if it starts from a local neighborhood of the desired pose) if the errors introduced in $\widehat{\mathbf{J}}_s$ are not too strong. Some particular visual servoing schemes can be demonstrated to be globally asymptotically stable (i.e., the robot will converge whatever its initial pose) under similar conditions. This is, for instance, the case for the pan-tilt camera control depicted on Fig. 1, for PBVS assuming the 3D parameters involved are perfectly estimated, and for well-designed IBVS schemes.

Finally, when the visual features do not constrain all the DoF, it is possible to combine the visual task with supplementary tasks such as joint limits avoidance or the visibility constraint (to be sure that the target considered will always remain in the camera field of view). In that case, the redundancy framework (Nakamura et al. 1987) can be applied and the new error to be regulated to zero has the following form:

$$\mathbf{e}_n = \widehat{\mathbf{J}}_s^+ \mathbf{e} + (\mathbf{I} - \widehat{\mathbf{J}}_s^+ \widehat{\mathbf{J}}_s) \mathbf{e}_2 \quad (10)$$

where $(\mathbf{I} - \widehat{\mathbf{J}}_s^+ \widehat{\mathbf{J}}_s)$ is a projection operator on the null space of the visual task $\mathbf{e}$ so that the supplementary task $\mathbf{e}_2$ will be achieved at best under the constraint that it does not perturb the visual task. A similar control scheme to (9) is now given by

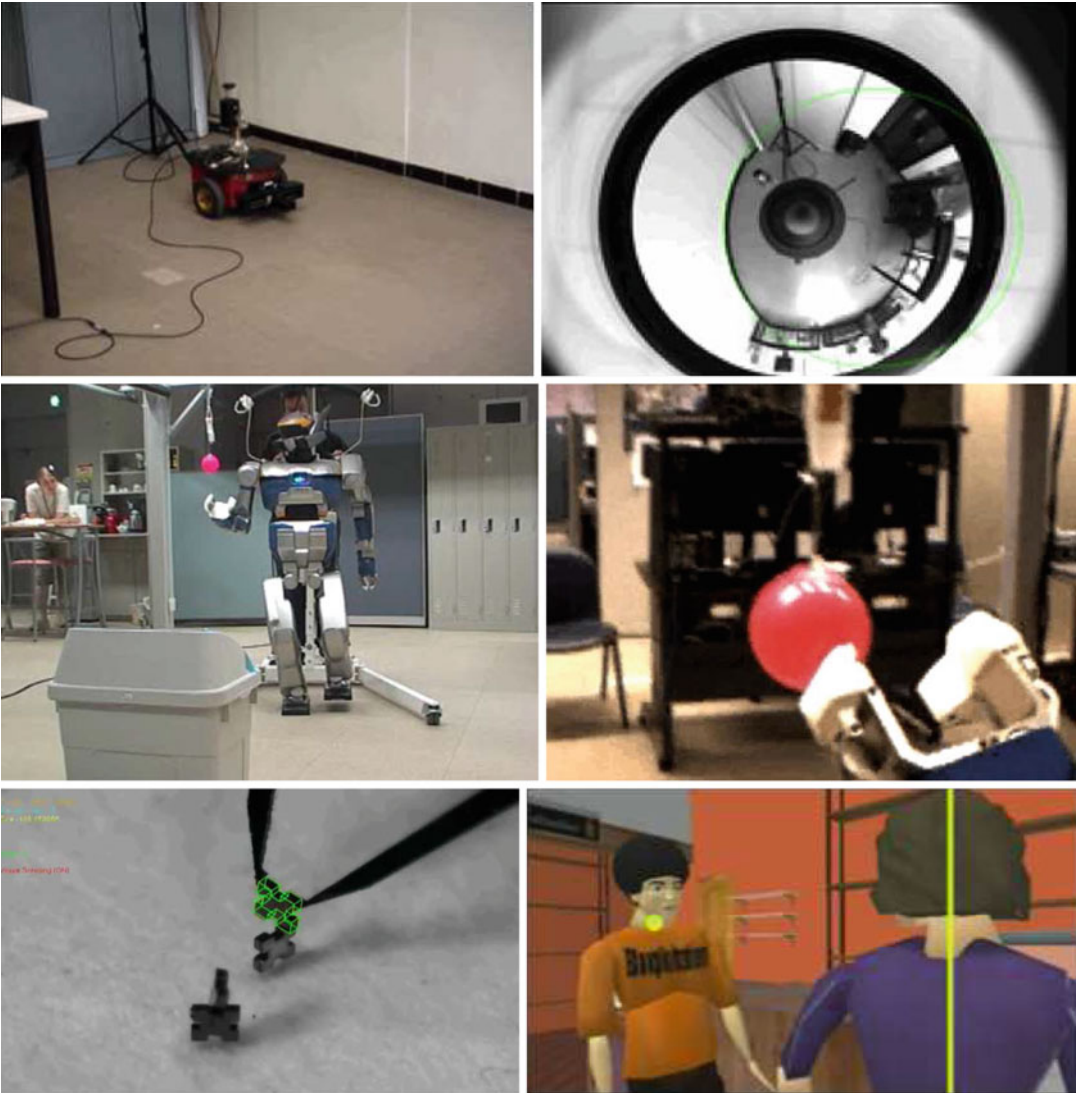
$$\dot{\mathbf{q}} = -\lambda \mathbf{e}_n - \frac{\partial \widehat{\mathbf{e}}_n}{\partial t} \quad (11)$$

This scheme has, for instance, been applied for the first example depicted on Fig. 4 where the

rotational motion of the mobile robot is controlled by vision while its translational motion is controlled by the odometry to move at a constant velocity.

Any other more advanced control strategy can be applied such as optimal control (Nelson and Khosla 1995; Hashimoto et al. 1996), coupling path planning and visual servoing (Mezouar and Chaumette 2002; Chesi 2009; Kazemi et al. 2013), model predictive control

(Ginhoux et al. 2005; Allibert et al. 2010), or quadratic programming (Agravante et al. 2016) when visual tasks and visual constraints have to be simultaneously handled with other tasks and constraints. Particular care has to be considered for underactuated and nonholonomic systems for which adequate control laws have to be designed (Hamel and Mahony 2002; Mebarki et al. 2015; Mariottini et al. 2007; Lopez-Nicolas et al. 2010).



Robot Visual Control, Fig. 4 Few applications of visual servoing: navigation of a mobile robot to follow a wall using an omnidirectional vision sensor (top line), grasping

a ball with a humanoid robot (middle line), and assembly of MEMS and film of a dialogue within the constraints of a script in animation (bottom line)

Applications

Potential applications of visual servoing are numerous. It can be used as soon as a vision sensor is available and a task is assigned to a dynamic system. A non-exhaustive list of examples is (see also Fig. 4):

- the control of a pan-tilt-zoom camera, as illustrated in Fig. 1 for the pan-tilt case;
- grasping using a robot arm;
- locomotion and dexterous manipulation with a humanoid robot;
- micro- or nano-manipulation of MEMS or biological cells;
- pipe inspection by an underwater autonomous vehicle;
- autonomous navigation of a mobile robot in indoor or outdoor environment;
- aircraft landing;
- autonomous satellite rendezvous;
- biopsy using ultrasound probes or heart motion compensation in medical robotics.
- virtual cinematography in animation.

Summary and Future Directions

Visual servoing is a mature area. It is basically a nonlinear control problem for which numerous modeling works have been achieved to design visual features so that the control problem is transformed as much as possible to a linear control problem. On the one hand, improvements on this topic are still expected for instantiating this general approach to particular applications. On the other hand, designing new control strategies is another direction for improvements, especially when supplementary data coming from other sensors (force, tactile, proximity sensors) are available. Finally, the current expansion of deep learning may rejuvenate the field (Levine et al. 2016; Bateux et al. 2018; Pandya et al. 2019), especially for the dense direct methods that use the same input and end-to-end approach.

Cross-References

- [Image-Based Formation Control of Mobile Robots](#)
- [Image-Based Robot Control](#)
- [Intermittent Image-Based Estimation](#)
- [Networked Control Systems: Estimation and Control over Lossy Networks](#)
- [Robot Motion Control](#)

Recommended Reading

In addition to the references already cited in text, including the classical tutorials by Hutchinson et al. (1996) and Chaumette et al. (2016), the book by Corke (2011) and the collection of papers by Hashimoto (1993), Kriegman et al. (1998), and Chesi and Hashimoto (2010) provide a good overview of past works in the field.

Bibliography

- Agravante DJ, Claudio G, Spindler F, Chaumette F (2016) Visual servoing in an optimization framework for the whole-body control of humanoid robots. *IEEE Robot Autom Lett* 2(2):608–615
- Allibert G, Courtial E, Chaumette F (2010) Predictive control for constrained image-based visual servoing. *IEEE Trans Robot* 26(5):933–939
- Bakthavatchalam M, Tahri O, Chaumette F (2018) A direct dense visual servoing approach using photometric moments. *IEEE Trans Robot* 34(5):1226–1239
- Bateux Q, Marchand E, Leitner J, Chaumette F, Corke P (2018) Training deep neural networks for visual servoing. In: *IEEE International Conference on Robotics and Automation (ICRA'18)*, pp 3307–3314
- Caron G, Marchand E, Mouaddib EM (2013) Photometric visual servoing for omnidirectional cameras. *Auton Robot* 35(2):177–193
- Chaumette F (2004) Image moments: a general and useful set of features for visual servoing. *IEEE Trans Robot* 20(4):713–723
- Chaumette F, Hutchinson S, Corke P (2016) Visual servoing. In: *Handbook of robotics*, 2nd edn, Chapter 34. Springer, Berlin, pp 841–866
- Chesi G (2009) Visual servoing path planning via homogeneous forms and LMI optimizations. *IEEE Trans Robot* 25(2):281–291
- Chesi G, Hashimoto K (eds) (2010) *Visual servoing via advanced numerical methods*. LNCIS, vol 401. Springer, Berlin

- Collewet C, Marchand E (2011) Photometric visual servoing. *IEEE Trans Robot* 27(4):828–834
- Corke P (2011) Robotics, vision and control. Springer tracts in advanced robotics, vol 73. Springer, Berlin
- Corke P, Good M (1996) Dynamic effects in visual closed-loop systems. *IEEE Trans Robot Autom* 12(5):671–683
- Crombez N, Mouaddib EM, Caron G, Chaumette F (2019) Visual servoing with photometric Gaussian mixtures as dense features. *IEEE Trans Robot* 35(1):49–63
- Dame A, Marchand E (2011) Mutual information-based visual servoing. *IEEE Trans Robot* 27(5):958–969
- Deguchi K (2000) A direct interpretation of dynamic images with camera and object motions for vision guided robot control. *Int J Comput Vis* 37(1):7–20
- De Luca A, Oriolo G, Robuffo Giordano P (2008) Feature depth observation for image-based visual servoing: theory and experiments. *Int J Robot Res* 38(4):422–450, 27(10):1093–1116
- Dufloy LA, Reisenhofer R, Tmadazte B, Andreff N, Krupa A (2019) Wavelet and shearlet-based image representations for visual servoing. *Int J Robot Res* 38(4):422–450
- Espiau B, Chaumette F, Rives P (1992) A new approach to visual servoing in robotics. *IEEE Trans Robot Autom* 8(3):313–326
- Ginhoux R, Gangloff J, de Mathelin M, Soler L, Sanchez MA, Marescaux J (2005) Active filtering of physiological motion in robotized surgery using predictive control. *IEEE Trans Robot* 21(1):67–79
- Hadj-Abdelkader H, Mezouar Y, Martinet P, Chaumette F (2008) Catadioptric visual servoing from 3D straight lines. *IEEE Trans Robot* 24(3):652–665
- Hamel T, Mahony R (2002) Visual servoing of an under-actuated dynamic rigid-body system: an image-based approach. *IEEE Trans Robot Autom* 18(2):187–198
- Han S, Censi A, Straw A, Murray R (2010) A bio-plausible design for visual pose stabilization. In: *International Conference on Intelligent Robots and Systems (IROS'10)*, pp 5679–5686
- Hashimoto K (ed) (1993) Visual servoing: real-time control of robot manipulators based on visual sensory feedback. World Scientific, Singapore
- Hashimoto K, Ebine T, Kimura H (1996) Visual servoing with hand-eye manipulator – optimal control approach. *IEEE Trans Robot Autom* 12(5):766–774
- Hosoda K, Asada M (1994) Versatile visual servoing without knowledge of true jacobian. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, pp 186–193
- Hutchinson S, Hager G, Corke P (1996) A tutorial on visual servo control. *IEEE Trans Robot Autom* 12(5):651–670
- Iwatsuki M, Okiyama N (2005) A new formulation of visual servoing based on cylindrical coordinate system. *IEEE Trans Robot Autom* 21(2):266–273
- Jägersand M, Fuentes O, Nelson R (1997) Experimental evaluation of uncalibrated visual servoing for precision manipulation. In: *IEEE International Conference on Robotics and Automation (ICRA'97)*, pp 2874–2880
- Kazemi M, Gupta K, Mehrandezh M (2013) Randomized kinodynamic planning for robust visual servoing. *IEEE Trans Robot* 29(5):1197–1211
- Khalil W, Dombre E (2002) Modeling, identification, and control of robots. CRC Press, Stanmore, UK
- Kriegman D, Hager G, Morse S (eds) (1998) The confluence of vision and control. LNCIS, vol 237. Springer, London
- Levine S, Finn C, Darrell T, Abbeel P (2016) End-to-end training of deep visuomotor policies. *J Mach Learn Res* 17(1):1334–1373
- Lopez-Nicolas G, Guerrero JJ, Sagues C (2010) Visual control through the trifocal tensor for nonholonomic robots. *Robot Auton Syst* 58(2):216–226
- Malis E, Chaumette F (2000) 2-1/2D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *Int J Comput Vis* 37(1):79–97
- Marchand E, Spindler F, Chaumette F (2005) ViSP for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robot Autom Mag* 12(4):40–52
- Marchand E, Uchiyama H, Spindler F (2016) Pose estimation for augmented reality: a hands-on survey. *IEEE Trans Vis Comput Graph* 22(12):2633–2651
- Mariottini GL, Oriolo G, Prattichizzo D (2007) Image-based visual servoing for nonholonomic mobile robots using epipolar geometry. *IEEE Trans Robot* 23(1):87–100
- Mebarki R, Krupa A, Chaumette F (2010) 2D ultrasound probe complete guidance by visual servoing using image moments. *IEEE Trans Robot* 26(2):296–306
- Mebarki R, Lippiello V, Siciliano B (2015) Nonlinear visual control of unmanned aerial vehicles in GPS-denied environments. *IEEE Trans Robot* 31(4):1004–1017
- Mezouar Y, Chaumette F (2002) Path planning for robust image-based control. *IEEE Trans Robot* 22(10):781–804
- Motyl G, Chaumette F, Gallice J (1992) Coupling a camera and laser stripe in sensor based control. In: *Second International Symposium on Measurement and Control in Robotics*, pp 685–692
- Nakamura Y, Hanafusa H, Yoshikawa T (1987) Task-priority based redundancy control of robot manipulators. *Int J Robot Res* 6(2):3–15
- Nayar S, Nene S, Murase H (1996) Subspace methods for robot vision. *IEEE Trans Robot Autom* 12(5):750–758
- Nelson B, Khosla P (1995) Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance. *Int J Robot Res* 14(3):225–269
- Pagès J, Collewet C, Chaumette F, Salvi J (2006) Optimizing plane-to-plane positioning tasks by image-based visual servoing and structured light. *IEEE Trans Robot* 22(5):1000–1010
- Pandya H, Gaud A, Kumar G, Madhava Krishna K (2019) Instance invariant visual servoing framework for part-aware autonomous vehicle inspection using MAVs. *J Field Rob* 36(5):892–918

- Silveira G, Malis E (2012) Direct visual servoing: vision-based estimation and control using only non-metric information. *IEEE Trans Robot* 28(4):974–980
- Suh I (1993) Visual servoing of robot manipulators by fuzzy membership function based neural networks. In: *Visual servoing. World scientific series on robotics and automated systems*, vol 7. World Scientific, Singapore, pp 285–315
- Teulière C, Marchand E (2014) A dense and direct approach to visual servoing using depth maps. *IEEE Trans Robot* 30(5):1242–1249
- Thiulot B, Martinet P, Cordesses L, Gallice J (2002) Position based visual servoing: keeping the object in the field of vision. In: *IEEE International Conference on Robotics and Automation (ICRA'02)*, pp 1624–1629
- Tsai R, Lenz R (1989) A new technique for fully autonomous and efficient 3D robotics hand-eye calibration. *IEEE Trans Robot Autom* 5(3):345–358
- Weiss L, Sanderson A, Neuman C (1987) Dynamic sensor-based control of robots with visual feedback. *IEEE J Robot Autom* 3(5):404–417
- Wells G, Venaille C, Torras C (1996) Vision-based robot positioning using neural networks. *Image Vis Comput* 14(10):715–732
- Wilson W, Hulls C, Bell G (1996) Relative end-effector control using cartesian position-based visual servoing. *IEEE Trans Robot Autom* 12(5):684–696

Robust Adaptive Control

Anuradha M. Annaswamy and
Joseph E. Gaudio

Active-adaptive Control Laboratory, Department
of Mechanical Engineering, Massachusetts
Institute of Technology, Cambridge, MA, USA

Abstract

Robust adaptive control pertains to the satisfactory behavior of adaptive control systems in the presence of nonparametric perturbations such as disturbances, unmodeled dynamics, and time delays. This entry covers the highlights of robust adaptive controllers, methods used, and results obtained. Both methods of achieving robustness, which include modifications in the adaptive law and persistent excitation in the reference input, are presented.

In both cases, results obtained for robustness to disturbances and unmodeled dynamics are discussed.

Keywords

Dead zone · Global boundedness ·
Parameter projection · Persistent excitation ·
Robustness · $|e|$ -modification · Magnitude
limits

Introduction

The central problem in adaptive control pertains to regulation and tracking of systems in the presence of parametric uncertainties. The classical adaptive control problem solved in 1980 assumed that the underlying transfer function had unknown parameters, but no other uncertainties. No disturbances, delays, time variations in parameters, or unmodeled dynamics were assumed to be present. Under these ideal conditions, it was shown that an adaptive controller can be designed so that the closed-loop system has bounded signals and that asymptotic regulation and tracking were possible.

With asymptotic regulation and tracking achieved under such ideal conditions, the goal of robust adaptive control was to ensure globally bounded signals in the closed-loop adaptive system when the plant was subjected to a variety of nonparametric perturbations such as external disturbances, time-varying parameters, unmodeled dynamics, and time delays. With adaptation in the control parameters in the ideal case accommodating parametric uncertainties, the approaches developed in robust adaptive control focused on developing solutions in the perturbed case to accommodate nonparametric uncertainties and improve on the classical adaptive controller which either underperformed or even exhibited instability with the introduction of nonparametric perturbations.

We briefly present the adaptive control solutions for the ideal case before proceeding with robust adaptive control.

Classical Adaptive Control

Adaptive Control of Plants with State Feedback

One of the very first problems where stable adaptive control was solved was for the case when states are accessible (Narendra and Kudva 1974), with the plant given by (the argument t is suppressed for the sake of convenience, except for emphasis)

$$\dot{x}_p = A_p x_p + b \lambda u \quad (1)$$

where $A_p \in \mathbb{R}^{n \times n}$ and the scalar λ are unknown parameters with b and the sign of λ known and (A_p, b) controllable. An adaptive controller that ensures global boundedness and asymptotic regulation and tracking for such plants is of the form

$$u = \theta_x^T(t) x_p + \theta_r(t) r, \quad (2)$$

and the adaptive laws for adjusting the unknown parameters are given by

$$\dot{\theta} = -\text{sign}(\lambda) \Gamma \omega b^T P e, \quad (3)$$

where $\omega = [x_p^T, r]^T$, $\theta = [\theta_x^T, \theta_r]^T$, $e = x - x_m$, and x_m is the state of a reference model

$$\dot{x}_m = A_m x_m + b r \quad (4)$$

with A_m Hurwitz, and P being the solution of the Lyapunov equation $A_m^T P + P A_m = -Q$, $Q > 0$, and $\Gamma \in \mathbb{R}^{(n+1) \times (n+1)}$ a diagonal positive definite matrix. The reference model in (4) is to be chosen so that certain matching conditions are satisfied, which are of the form

$$A_p + b \lambda \theta_x^{*T} = A_m, \quad \lambda \theta_r^* = 1 \quad (5)$$

for some $\theta^* = [\theta_x^{*T}, \theta_r^*]^T$. In such a case, the controller in (2) and (3) guarantees stability and ensures that $x(t)$ tracks $x_m(t)$. The underlying Lyapunov function is quadratic in e and the parameter error $\tilde{\theta} = \theta - \theta^*$, given by

$$V = \frac{1}{2} (e^T P e + \lambda \tilde{\theta}^T \Gamma^{-1} \tilde{\theta}) \quad (6)$$

with a negative semi-definite time derivative $\dot{V}$ given by

$$\dot{V} \leq -e^T Q e. \quad (7)$$

Adaptive Control of Plants with Output Feedback

Consider the *single-input single-output* (SISO) system of equations

$$y(t) = W(s)u(t) \quad (8)$$

where $u \in \mathbb{R}$ is the input, $y \in \mathbb{R}$ the measurable output, and s the differential operator. The transfer function of the plant is parameterized as

$$W(s) \triangleq k_p \frac{Z(s)}{P(s)} \quad (9)$$

where k_p is a scalar and $Z(s)$ and $P(s)$ are monic polynomials with $\deg(Z(s)) < \deg(P(s))$. The following assumptions will be made throughout:

Assumption 1 $W(s)$ is minimum phase.

Assumption 2 The sign of k_p is known.

Assumption 3 The relative degree n^* and the order of $W(s)$ are known.

The goal is to design a control input u so that the output y in (8) tracks the output y_m of the reference system

$$y_m(t) = W_m(s)r(t) \triangleq k_m \frac{Z_m(s)}{P_m(s)} r(t) \quad (10)$$

where k_m is a scalar and $Z_m(s)$ and $P_m(s)$ are monic polynomials with $W_m(s)$ relative degree n^* .

The structure of the adaptive controller is now presented:

$$\dot{\omega}_1(t) = \Lambda \omega_1 + b_\lambda u(t) \quad (11)$$

$$\dot{\omega}_2(t) = \Lambda \omega_2 + b_\lambda y(t) \quad (12)$$

$$\omega(t) \triangleq [r(t), \omega_1^T(t), y(t), \omega_2^T(t)]^T \quad (13)$$

$$\theta(t) \triangleq [k(t), \theta_1^T(t), \theta_0(t), \theta_2^T(t)]^T \quad (14)$$

$$u(t) = \theta^T(t)\omega(t) \quad (15) \quad \text{function candidate given by}$$

where $\Lambda \in \mathbb{R}^{(n-1) \times (n-1)}$ is Hurwitz, $b_\lambda \in \mathbb{R}^{n-1}$, $\omega_1, \omega_2 \in \mathbb{R}^{n-1}$, and $\theta \in \mathbb{R}^{2n}$ is an adaptive gain vector with $k(t) \in \mathbb{R}$, $\theta_1(t) \in \mathbb{R}^{n-1}$, $\theta_2(t) \in \mathbb{R}^{n-1}$, and $\theta_0(t) \in \mathbb{R}$.

The update law for the adaptive parameter differs depending on whether the relative degree of $W_m(s)$ is unity or greater than one and can be described as follows:

$$\dot{\theta}(t) = -\text{sign}(k_p)\Gamma e_y \omega \quad n^* = 1 \quad (16)$$

and

$$\dot{\theta}(t) = -\text{sign}(k_p)\Gamma \frac{e_a \zeta}{1 + \zeta^T \zeta} \quad n^* \geq 2 \quad (17)$$

where $e_y = y - y_m$, e_a is an augmented error, and ζ is a modified regressor, both of which are determined by the following equations:

$$\zeta = W_m(s)I(s)\omega, \quad \omega = [r, \omega_1^T, y, \omega_2^T]^T, \quad (18)$$

$$e_2 = \theta^T \zeta - W_m(s)[\theta^T \omega] \quad (19)$$

$$e_a = e_y + k_1(t)e_2 \quad (20)$$

$$\dot{k}_1 = -\frac{e_a e_2}{1 + \zeta^T \zeta} \quad (21)$$

The results of Narendra and Annaswamy (2005) guarantee that the above adaptive controller in Eqs. (11)–(21) will guarantee that $e_y(t)$ tends to zero as $t \rightarrow \infty$ with all signals remaining bounded in both the $n^* = 1$ and $n^* \geq 2$ cases.

Need for Robust Adaptive Control

When a disturbance η is present, the plant dynamics often is of the form

$$\dot{x}_p = A_p x_p + b\lambda(u + \eta(t)) \quad (22)$$

while the reference model and the controller remain the same as in (4) and (2), respectively. This in turn necessitates new tools for the analysis and synthesis of adaptive systems. The main reason for this is the fact that the standard Lyapunov

$$V = \frac{1}{2}e^T P e + \frac{1}{2}\lambda \tilde{\theta}^T \Gamma^{-1} \tilde{\theta} \quad (23)$$

together with the parameter adjustment as in (3) yields a time derivative

$$\dot{V} \leq -\frac{1}{2}e^T Q e + k_1 \|e\| d_0 \quad k_1 > 0, \quad (24)$$

where d_0 is an upper bound on the perturbation η . The second term on the right-hand side of (24) causes $\dot{V}$ to be sign indefinite. This is because V is a function of both e and $\tilde{\theta}$, and therefore, the second term can be large compared to the first with the second argument of V , $\tilde{\theta}$, which can be arbitrary, causing $\dot{V}$ to be sign indefinite. This property causes $\dot{V}$ to be semi-definite in the ideal case. Hence, in this perturbed case, no guarantees of boundedness can be provided. In fact, it can be shown that if $\eta(t)$ is chosen in a particular manner, the closed-loop signals can actually be shown to become unbounded, either in the presence of bounded disturbances (Narendra and Annaswamy 2005) or with unmodeled dynamics (Rohrs et al. 1985). This in turn led to the area of robust adaptive control.

Various approaches that have been developed under the rubric of robust adaptive control can be grouped into two categories. The first of these is related to modifications in the adaptive laws so as to ensure boundedness. These changes consist of modifications in the adaptive law (3) as

$$\dot{\theta} = \text{sign}(\lambda) - \Gamma \omega b^T P e - \sigma g(\theta, e) \quad (25)$$

The problem then reduces to finding a suitable $g(\theta, e)$. This is discussed in detail in the next section. The second approach used in adaptive control pertains to the use of a persistently exciting reference signal r . The latter ensures parameter convergence of the adaptive system and therefore exponential stability. This in turn ensures robustness of the overall system. These details are addressed in section “[Robust Adaptive Control with Persistently Exciting Reference Input](#)”.

Robust Adaptive Control with Modifications in the Adaptive Law

Robustness to Bounded Disturbances

When a bounded input disturbance η is present, the plant dynamics is changed as

$$\dot{x}_p = A_p x_p + b\lambda(u + \eta(t)), \quad (26)$$

$$g(\theta, e) = \begin{cases} \theta & \text{Ioannou and Sun (2013)} \\ \|e\|\theta & \text{Narendra and Annaswamy (1987)} \\ \theta \left(1 - \frac{\|\theta\|}{\theta_{\max}}\right)^2 & \text{Kreisselmeier and Narendra (1982)} \end{cases} \quad (27)$$

where $\theta_{\max}$ is a known bound on the parameter θ . (One can choose to set σ to zero if $\|\theta\| \leq \theta_{\max}$, as is done in Ioannou and Sun (2013), Tsakalis and Ioannou (1987), and many other references in the literature.) An alternate approach that is different from (25) is to not have an additive term $g(\cdot, \cdot)$ but rather set $\dot{\theta} = 0$ whenever the error e is small in some sense. Such a dead zone approach was suggested, for example, in Egardt (1979) and Peterson and Narendra (1982). It can be shown that each one of these choices leads to boundedness, which is described below. Without loss of generality, we assume that $\lambda > 0$.

With the same Lyapunov function candidate as in (23), its time derivative now becomes

$$\dot{V} \leq -\frac{1}{2}e^T Q e + k_1 \|e\| \|\eta\| - k_2 \tilde{\theta}^T g(\theta, e), \quad (28)$$

for some $k_1, k_2 > 0$.

The property of $g(\cdot, \cdot)$, together with the fact that η is bounded, ensures that $\dot{V} < 0$ outside a compact set Ω in the $(e, \tilde{\theta})$ space. This ensures global boundedness of both e and $\tilde{\theta}$. Boundedness of x_p follows.

In all of the above methods, the idea behind adding the term $g(e, \theta)$ is this: the parameter θ can drift away from the correct direction due to the term $k_1 \|e\| \|\eta\|$, and the construction of

while the reference model and the controller remain the same as in (4) and (2), respectively. As mentioned above, a modification to the adaptive law as in (25) is needed. Over the years, different choices have been suggested for the nonlinear function $g(\theta, e)$. For example, these are chosen as

$g(e, \theta)$ is such that it counteracts this drift and keeps the parameter in check, by adding a negative quadratic term in $\tilde{\theta}$. The boundedness of both e and θ is simultaneously assured in the above since V has a time derivative $\dot{V}$ that is nonpositive outside a compact set in the $(e, \tilde{\theta})$ space. It should be noted however that this was possible to a large extent because η was bounded, and as a result, the sign-indefinite term remained linear in $\|e\|$.

An alternative procedure, originally proposed in Pomet and Praly (1992) and revised and refined in Khalil (2001) and Lavretsky (2011), proceeds in a slightly different manner. Here, the boundedness of θ is first established, independent of the error equation. It should be noted that a similar approach is adopted in the context of output feedback in plants with higher relative degree by using normalization and an augmented error approach (Narendra and Annaswamy 2005). In Khalil (2001) and Lavretsky (2011), no normalization is used but a projection algorithm. This is described below.

The projection algorithm for adjusting the parameter θ is given by

$$\dot{\theta} = \text{Proj}(\theta, y), \quad (29)$$

where

$$\text{Proj}(\theta, y) = \begin{cases} y - \frac{\nabla f(\theta)(\nabla f(\theta))^T}{\|\nabla f(\theta)\|^2} y f(\theta) & \text{if } [f(\theta) > 0 \wedge y^T \nabla f(\theta) > 0] \\ y & \text{otherwise} \end{cases} \quad (30)$$

$$y = -\text{sign}(\lambda)\omega b^T P e \quad (31)$$

$$f(\theta) = \frac{\|\theta\|^2 - \theta_{\max}^2}{\varepsilon^2 + 2\varepsilon\theta'_{\max}} \quad (32)$$

where $\theta'_{\max}$ and ε are arbitrary positive constants and Ω_0 and Ω_1 are defined as

$$\begin{aligned} \Omega_0 &= \{\theta \in \mathbb{R}^n | f(\theta) \leq 0\} \\ \Omega_1 &= \{\theta \in \mathbb{R}^n | f(\theta) \leq 1\}. \end{aligned} \quad (33)$$

From the above relations, one can show that

$$\theta(0) \in \Omega_0 \implies \theta(t) \in \Omega_1.$$

In addition,

$$\theta'_{\max} = \max_{\theta \in \Omega_0} (\|\theta\|), \quad \theta_{\max} = \max_{\theta \in \Omega_1} (\|\theta\|) \quad (34)$$

where $\theta_{\max} = \theta'_{\max} + \varepsilon$ (Matsutani et al. 2011).

Robustness to Unmodeled Dynamics

One of the major observations in the early 1980s was the stark difference between the system signals in the ideal adaptive system and the perturbed adaptive system when the perturbation was due to commonly present unmodeled dynamics such as those of an actuator used for control implementation. Among other references, the publication in Rohrs et al. (1985) pointed out the fact that when an adaptive controller prescribed for a first-order plant is evaluated with unmodeled dynamics present, instability occurs readily and for a wide range of command signals. A number of solutions have been suggested to alleviate this instability and form the subject matter of this section.

We consider the plant in (26) with an additional unmodeled dynamics so that

$$\begin{aligned} \dot{x}_p &= A_p x_p + b \lambda v \\ \dot{x}_\eta &= A_\eta x_\eta + b_\eta u, \quad v = \tilde{c}_\eta^T x_\eta. \end{aligned} \quad (35)$$

where A_η is a Hurwitz matrix. If $\eta = v - u$, then the plant dynamics can be rewritten as

$$\dot{x}_p = A_p x_p + b \lambda (u + \eta) \quad (36)$$

Unlike the bounded disturbance case, no upper bound d_0 can be assumed to exist as η is a state-dependent disturbance. It is this that causes a huge difference between deriving boundedness in section “[Robustness to Bounded Disturbances](#)” and here in section “[Robustness to Unmodeled Dynamics](#)”. Significant effort has been extended in the adaptive control community in this regard. These results fall into two categories (i) that assure global boundedness for a narrow class of unmodeled dynamics and (ii) that assure semi-global boundedness for a slightly larger class of unmodeled dynamics. More recently, some results have been obtained that are able to establish global boundedness with minimal restrictions on the unmodeled dynamics. In what follows, we give examples of each of the above two cases as well as the recent results.

Global Boundedness in the Presence of a Small Class of Unmodeled Dynamics

For the plant in (26), under assumptions in (5), the plant can be rewritten as

$$\dot{x} = A_m x + b \lambda (u + \theta_x^{*T} x + \eta) \quad (37)$$

where λ and θ_x^* are unknown, A_m and b are known, and $\eta = v - u$ whose state-space representation can be shown to be of the form

$$\dot{x}_\eta = A_\eta x_\eta + b_\eta u, \quad \eta = c_\eta^T x_\eta \quad (38)$$

for some vector c_η .

For a class of unmodeled dynamics $\{c_\eta, A_\eta, b_\eta\}$, if the control input in (2) and the projection algorithm in (29) with y and $f(\theta)$ chosen as in (31) and (32) are used, one can guarantee boundedness. In particular, if the inequality

$$k\theta_{x,\max}\lambda_{\max}\left(\frac{b_0}{\sigma_{A_\eta}}\right) < 1 \quad (39)$$

is satisfied, where b_0 is an upper bound on $\|b_\eta\|$ and σ_A denotes the singular value of the matrix A , then boundedness follows. That is, robustness of adaptive controllers can be ensured if the

unmodeled dynamics are fast and their zeros are restricted in some sense.

A specific example of such an unmodeled dynamics is given by

$$c_\eta^T (sI - A_\eta)^{-1} b_\eta = \frac{-2\mu s}{1 + \mu s} \quad (40)$$

for a given $\mu > 0$.

Global Boundedness for a Large Class of Unmodeled Dynamics: A First-Order Example

A different approach can be taken for the problem of unmodeled dynamics which allows a global result, for a class of adaptive systems (Hussain et al. 2013). The main idea here is to use the projection algorithm and use properties of adaptive systems in conjunction with linear time-varying systems. This is presented in this section using a first-order plant.

We consider the control of

$$\dot{x}_p(t) = a_p x_p(t) + k_p v(t) \quad (41)$$

where a_p is unknown and k_p is known. It is assumed that $|a_p| \leq \bar{a}$, where $\bar{a}$ is a known positive constant. The unmodeled dynamics is given by (38) with

$$G_\eta(s) \triangleq c_\eta^T (sI_{n \times n} - A_\eta)^{-1} b_\eta. \quad (42)$$

The goal is to design the control input such that $x_p(t)$ follows $x_m(t)$ which is specified by the reference model

$$\dot{x}_m(t) = a_m x_m(t) + k_m r(t) \quad (43)$$

where $a_m < 0$ and reference input is $r(t)$. The proposed adaptive controller is of the form

$$u(t) = \theta(t)x_p(t) + \frac{k_m}{k_p} r(t) \quad (44)$$

where the parameter $\theta(t)$ is updated using a projection algorithm of the form

$$\begin{aligned} \dot{\theta}(t) &= \gamma \text{Proj}(\theta(t), -x_p(t)(x_p(t) - x_m(t))), \\ \gamma &> 0 \end{aligned} \quad (45)$$

and

$$\text{Proj}(\theta, y) = \begin{cases} \frac{\theta_{\max}^2 - \theta^2}{\theta_{\max}^2 - \theta_{\min}^2} y & [\theta \in \Omega_A, y > 0] \\ y & \text{otherwise} \end{cases} \quad (46)$$

$$\Omega_0 = \{\theta \in \mathbb{R}^1 \mid -\theta'_{\max} \leq \theta \leq \theta'_{\max}\}$$

$$\Omega_1 = \{\theta \in \mathbb{R}^1 \mid -\theta_{\max} \leq \theta \leq \theta_{\max}\} \quad (47)$$

$$\Omega_A = \Omega_1 \setminus \Omega_0$$

with positive constants $\theta'_{\max}$ and $\theta_{\max}$ given by

$$\theta'_{\max} > \frac{\bar{a} + |a_m|}{k_p}, \quad \theta_{\max} = \theta'_{\max} + \varepsilon_0, \quad (48)$$

where ε_0 is an arbitrary positive constant. It can be shown that if $\theta_{\max}$ is chosen as in (48), then the closed adaptive system specified by Eqs. (41)–(48) always has guaranteed bounded solutions for a class of unmodeled dynamics $G_\eta(s)$. Additionally, there is an optimal value of ε_0 for which the largest class of $G_\eta(s)$ can be found.

It should be noted that in the Rohrs example in Rohrs et al. (1985), the plant is first order, with $a_p = -1$, and

$$G_\eta = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \quad (49)$$

for $\zeta = 1$, $\omega_n = 15$. It is easy to show that for these values of ζ and ω_n , if $\theta_{\max} = 17$, then Eq. (48) is satisfied and that the closed-loop system is robust to G_η .

In general, for a first-order plant as in (41), it can be shown that the adaptive system is robust for G_η for all (ζ, ω_n) that satisfy the following inequalities for all $|a_p| \leq \bar{a}$:

$$\begin{aligned} -a_p \zeta^2 + f(a_p, \omega_n) \zeta - \frac{k_p \theta_{\max}}{4} &> 0 \\ \omega_n &> \omega_{n_{\min}} \end{aligned} \quad (50)$$

where

$$f(a_p, \omega_n) = \frac{a_p^2 + \omega_n^2}{2\omega_n}$$

$$\omega_{n_{\min}} = \max \left(\frac{\bar{a}}{2\zeta}, 2\zeta\bar{a}, \sqrt{\bar{a}k_p\theta_{\max}} \right. \\ \left. \left\{ 1 + \sqrt{1 - \frac{\bar{a}}{k_p\theta_{\max}}} \right\} \right) \quad (51)$$

When a time delay τ is present in the plant to be controlled, the plant under consideration can be represented as in (37) where

$$\eta(t) = u(t - \tau) - u(t).$$

Similar results of boundedness can be derived in this case as well (Matsutani 2013; Matsutani et al. 2012, 2013), with a comprehensive derivation available in Hussain et al. (2017).

Adaptive Control in the Presence of Magnitude Constraints on the Input

Physical plant models commonly have input magnitude constraints which need to be explicitly accounted for in the design of robust adaptive control systems. The first such results which design adaptive update laws for a general class of output feedback plants was described in Kárason and Annaswamy (1994). Similar to (8), consider the single-input single-output plant of the form

$$y_p(t) = W_p(s)u(t) = \left[k_p \frac{Z_p(s)}{R_p(s)} \right] u(t) \quad (52)$$

where the input $u(t)$ is subjected to a magnitude constraint as

$$|u(t)| \leq u_0, \quad u_0 > 0. \quad (53)$$

It is assumed that $Z_p(s)$ is Hurwitz and the sign of k_p , the order, and the relative degree of plant are all known. In the presence of magnitude saturation, the goal is for the plant output $y_p(t)$ to track a reference trajectory $y_m(t)$ which is

generated from

$$y_m(t) = W_m(t)r(t) = \left[k_m \frac{Z_m(s)}{R_m(s)} \right] r(t) \quad (54)$$

where $W_m(s)$ is designed to be asymptotically stable, minimum phase and with known relative degree n^* , similar to (10). Additionally the reference signal $r(t)$ is assumed to be piecewise-continuous and satisfy $|r(t)| \leq r_0$ for a finite $r_0 > 0$.

Then as in Kárason and Annaswamy (1994), the following controller modified from Eqs. (11), (12), (13), (14), and (15) may then be chosen:

$$\dot{\omega}_1(t) = \Lambda \omega_1(t) + b_\lambda u(t) \quad (55)$$

$$\dot{\omega}_2(t) = \Lambda \omega_2(t) + b_\lambda y_p(t) \quad (56)$$

$$\omega(t) = [r(t), \omega_1^T(t), y_p(t), \omega_2^T(t)] \quad (57)$$

$$\theta(t) = [k(t), \theta_1^T(t), \theta_0(t), \theta_2^T(t)] \quad (58)$$

$$v(t) = \theta^T(t)\omega(t) \quad (59)$$

$$u(t) = \begin{cases} v(t), & \text{if } |v(t)| \leq u_0 \\ u_0 \operatorname{sgn}(v(t)), & \text{if } |v(t)| > u_0 \end{cases} \quad (60)$$

where $\theta_1(t), \theta_2(t), \omega_1(t), \omega_2(t) \in \mathbb{R}^{n-1}$, $k(t), \theta_0(t) \in \mathbb{R}$, $b_\lambda \in \mathbb{R}^{n-1}$, and $\Lambda \in \mathbb{R}^{(n-1) \times (n-1)}$ is Hurwitz. The parameter estimation errors may then be stated as: $\psi(t) = k(t) - k^*$, $\phi_0(t) = \theta_0(t) - \theta_0^*$, $\phi_1(t) = \theta_1(t) - \theta_1^*$, $\phi_2(t) = \theta_2(t) - \theta_2^*$, $\phi(t) = [\psi(t), \phi_1^T(t), \phi_0(t), \phi_2^T(t)]^T$.

With the output tracking error defined as $e_1 = y_p - y_m$ and input disturbance given by the control deficiency defined as $\Delta u = u - v$, the error model may be stated as

$$e_1(t) = \frac{k_p}{k_m} W_m(s) [\phi^T(t)\omega(t) + \Delta u(t)]. \quad (61)$$

Given the additional disturbance term Δu in the error model, an additional signal is generated as

$$e_\Delta(t) = k_\Delta(t) W_m(s) \Delta u(t) \quad (62)$$

where $k_\Delta(t)$ is an additional parameter for which an adaptive law will be designed. Defining $e_{u1} =$

$e_1 - e_\Delta$:

$$e_{u1}(t) = \frac{1}{k^*} W_m(s) [\phi^T(t) \omega(t)] + \psi_\Delta(t) W_m(s) \Delta u(t) \quad (63)$$

with $\psi_\Delta(t) = \frac{1}{k^*} - k_\Delta(t)$. The auxiliary and augmented errors may then be defined as

$$e_2(t) = [\theta^T W_m(s) I - W_m(s) \theta^T(t)] \omega(t) \quad (64)$$

$$\varepsilon_{1u}(t) = e_{u1}(t) + k_1(t) e_2(t). \quad (65)$$

The adaptive update laws for this error model may then be stated of the form

$$\dot{\theta}(t) = \dot{\phi}(t) = -\text{sgn}(k_p) \frac{\varepsilon_{1u}(t) \xi(t)}{1 + \xi^T(t) \xi(t)} \quad (66)$$

$$\dot{k}_1(t) = \dot{\psi}_1(t) = -\frac{\varepsilon_{1u}(t) e_2(t)}{1 + \xi^T(t) \xi(t)} \quad (67)$$

$$\dot{k}_\Delta(t) = -\dot{\psi}_\Delta(t) = \frac{\varepsilon_{1u}(t) \xi_\Delta(t)}{1 + \xi^T(t) \xi(t)} \quad (68)$$

where $\xi(t) = W_m(t) I \omega(t)$, $\xi_\Delta(t) = W_m(s) \Delta u(t)$, $\psi_1(t) = k_1(t) - (1/k^*)$ (Káráson and Annaswamy 1994).

It can be noted that the additional components of the modified adaptive update laws directly account for the control deficiency Δu . Due to the constraints on the input, for general plant models $W_p(s)$, stability results are local in nature. If the plant $W_p(s)$ is asymptotically stable, then due to the presence of a saturated input, stability follows due to the bounded-input bounded-output property. For open-loop unstable plants, regions of attraction may be defined which are proportional to the level of saturation u_0 and upper bounds on the uncertainty of the system. It can be noted that for open-loop unstable plants, there always exist initial conditions for which the closed-loop system can be unstable, thus the need for derivations of regions of attraction as provided in Káráson and Annaswamy (1994).

Adaptive Augmentation of Robust Controllers

Commonly adaptive controllers are employed as augmentations to robust output feedback controllers. One such example which has been employed in aircraft flight platforms is the observer-based loop transfer recovery (OBLTR) method as described in Lavretsky and Wise (2013). Consider the open-loop plant model of the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B\Lambda^* \left(u(t) + \Theta^{*T} \Phi(x(t)) \right) \\ &\quad + B_z z_{cmd}(t) \end{aligned} \quad (69)$$

$$y(t) = Cx(t) \quad (70)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $B_z \in \mathbb{R}^{n \times n_z}$, and $C \in \mathbb{R}^{p \times n}$ are known dynamics, input, reference, and output matrices, respectively. The unknown diagonal positive definite input control effectiveness matrix is denoted as $\Lambda^* \in \mathbb{R}^{m \times m}$, and the unknown additive uncertainty is $\Theta^* \in \mathbb{R}^{N \times m}$. The additive uncertainty acts upon a known continuously differentiable regressor $\Phi(x(t)) \in \mathbb{R}^{N \times 1}$. It is assumed that the pair (A, B) is controllable and the pair (A, C) is observable. Given that this system is in an output feedback setting, a state observer, which will function additionally as a closed-loop reference model, is designed as

$$\dot{\hat{x}}(t) = A_m \hat{x}(t) + L_v (y(t) - \hat{y}(t)) + B_z z_{cmd}(t) \quad (71)$$

$$\hat{y}(t) = C \hat{x}(t) \quad (72)$$

where $A_m = A - BK_{LQR}^T$. A systematic loop transfer recovery-based procedure is provided by Lavretsky and Wise (2013) to design the state feedback gain $K_{LQR} \in \mathbb{R}^{n \times m}$ and observer gain $L_v \in \mathbb{R}^{n \times p}$.

The control input may then be selected as a combination of a robust output feedback contribution and adaptive augmentation as

$$u(t) = u_{bl}(t) + u_{ad}(t) \quad (73)$$

where $u_{bl}(t) = -K_{LQR}^T \hat{x}(t)$ is the robust baseline control contribution and $u_{ad}(t)$ is an adap-

tive augmentation. Based on the design of the output feedback robust baseline control $u_{bl}(t)$, Lavretsky and Wise (2013) provides a method to synthesize the adaptive input $u_{ad}(t)$ given the robust output feedback design parameters K_{LQR} , L_v and the known plant matrices.

Robustness Modifications to Track Time-Varying Parameters

Similar to the discussion in Narendra and Annaswamy (2005), a scalar plant of the following form will be considered to elucidate the essential problem of tracking time-varying parameters:

$$\dot{x}_p(t) = a_p(t)x_p(t) + u(t) \quad (74)$$

where $a_p(t)$ is the unknown time-varying parameter. The reference model for the plant to track is of the form

$$\dot{x}_m(t) = a_m x_m(t) + r(t) \quad (75)$$

where $a_m < 0$ and $r(t)$ is a selected reference input. The control input is chosen as

$$u(t) = \theta(t)x_p(t) + r(t). \quad (76)$$

With a model tracking error $e = x_p - x_m$, the error model is of the form

$$\dot{e}(t) = a_m e(t) + (a_p(t) + \theta(t) - a_m(t))x_p(t). \quad (77)$$

If $a_p(t)$ were to be known, then the parameter estimate could be chosen as $\theta(t) = \theta^*(t)$, with $\theta^*(t) = a_m - a_p(t)$. Such a choice of $\theta(t)$ would result in model tracking with $\lim_{t \rightarrow \infty} e(t) = 0$. Given however that $a_p(t)$ (and thus $\theta^*(t)$) is unknown, an adaptive update law must be designed using available input-output data to achieve model tracking. The time-varying nature of the unknown parameter requires additional assumptions regarding the time variation in order to arrive at a tractable problem statement. One such assumption regarding the time variation is bounded magnitude and time derivative of the unknown parameter, i.e., $|a_p(t)| \leq a_0$ and $|\dot{a}_p(t)| \leq d_0$.

With a parameter error $\phi(t) = \theta(t) - \theta^*(t)$ and standard adaptive law $\dot{\theta} = -e x_p$, the model tracking and parameter error equations may be expressed of the form

$$\begin{aligned} \dot{e}(t) &= a_m e(t) + \phi(t)x_p(t) \\ \dot{\phi}(t) &= -e(t)x_p(t) - \dot{\theta}^*(t) \end{aligned} \quad (78)$$

It can be noted that (78) is nonlinear and nonautonomous, and due to the presence of $\dot{\theta}^*(t)$, additional modifications are necessary to ensure boundedness of $e(t)$ and $\phi(t)$. One such modification is the sigma-modification (Ioannou and Sun 2013) in (27) for which the adaptive law is then of the form

$$\dot{\phi}(t) = -e(t)x_p(t) - \sigma\theta(t) - \dot{\theta}^*(t) \quad (79)$$

where $\sigma > 0$ is a user-designed parameter. A Lyapunov function $V(e(t), \phi(t)) = e^2(t) + \phi^2(t)$ can be employed to yield a time derivative $\dot{V}(e(t), \phi(t))$ which is negative definite outside of a compact region in the (e, ϕ) space containing the origin (Narendra and Annaswamy 2005).

Robust Adaptive Control with Persistently Exciting Reference Input

We return to the plant in (26) with the control input as in (2) and the adaptive law as in (3). When $\eta(t)$ is bounded with a finite upper bound d_0 , it can be shown that no modifications are necessary in the adaptive law to ensure boundedness if the reference input is persistently exciting. It can be shown that if the reference input $r(t)$ is such that the vector ω^* defined as $\omega^* = [x_m^T, r]^T$ is persistently exciting with

$$\left| \frac{1}{T} \int_t^{t+T} \omega^{*T}(\tau) w d\tau \right| \geq k d_0 \quad \forall t \geq t_0$$

for every unit vector $w \in \mathbb{R}^n$, where $k > 0$, T are finite constants, then the adaptive system is well behaved, i.e., has globally bounded solutions (Narendra and Annaswamy 2005).

An alternative approach for achieving robustness has been addressed in Anderson et al. (1986) that addresses local stability in the presence of persistently exciting signals. The starting point for this investigation is (35) but when not all states are accessible. Assuming that an output $y = c_p^T x_p$ is measurable and a controller as in (11), (12), (13), (14), and (15) and a reference model as in (10) are used, the underlying error equation can be written as

$$e_1 = \bar{W}_m(s) (\tilde{\theta}^T \omega + \bar{v}) \quad (80)$$

where $\bar{W}_m(s)$ is asymptotically stable, $\tilde{\theta}$ is the parameter error vector, and $\bar{v}$ is the effect of the unmodeled dynamics η at the input. Suppose the standard adaptive law is used and as a first step, the perturbation $\bar{v}$ is ignored, the underlying error equation and the adaptive law are given by

$$e_1 = \bar{W}_m(s) \tilde{\theta}^T \omega \quad (81)$$

$$\dot{\tilde{\theta}} = -\mu e_1 \omega, \quad \mu > 0. \quad (82)$$

If the origin in the $(e_1, \tilde{\theta})$ space of (81) and (82) is exponentially stable, all solutions of (80) are bounded for sufficiently small initial conditions and $\bar{v}(t)$. Therefore, the question that is of interest is the set of conditions of persistent excitation that will assure such an exponential stability. This is addressed in Anderson et al. (1986). The underlying tool is the method of averaging (Krylov and Bogoliuboff 1943; Hale 1969; Arnold 1988) used in the study of nonlinear oscillations and addresses the stability property of the differential equation

$$\dot{x} = \mu f(x, t, \mu), \quad x(0) = x_0 \quad (83)$$

where μ is a small parameter. By a process of averaging, the nonautonomous system in (83) is approximated by an autonomous differential equation in x_{av} , an averaged value of x . This autonomous system, which is easier to analyze, can be used to derive stability properties of (83).

In order to use the method of averaging for robust adaptive control, we write Eqs. (81) and (82) as

$$\begin{bmatrix} \dot{e} \\ \dot{\tilde{\theta}} \end{bmatrix} = \begin{bmatrix} A & b\omega^T \\ -\mu\omega h^T & 0 \end{bmatrix} \begin{bmatrix} e \\ \tilde{\theta} \end{bmatrix} \quad (84)$$

Theorem 1 *Let $\omega(t)$ be bounded, almost periodic, and persistently exciting. Then there exists a $c^* > 0$ such that for all $\mu \in (0, c^*]$, the origin of (84) is exponentially stable if*

$$\Re \left[\lambda_i \left(\int_0^T \omega(t) \bar{W}_m(s) \omega^T(t) dt \right) \right] > 0, \quad \forall i = 1, \dots, n \quad (85)$$

and is unstable if

$$\Re \left[\lambda_j \left(\int_0^T \omega(t) \bar{W}_m(s) \omega^T(t) dt \right) \right] < 0, \quad \text{for some } j = 1, \dots, n \quad (86)$$

In Kokotovic et al. (1985), it is further shown that $\omega(t)$ can be expressed as $\omega(t) = \sum_{k=-\infty}^{\infty} \Omega(iv_k) \exp(iv_k t)$, and the inequality in (85) can be satisfied if the condition

$$\sum_{k=-\infty}^{\infty} \Re [\bar{W}_m(iv_k)] \Re [\Omega(iv_k) \bar{\Omega}^T(iv_k)] > 0 \quad (87)$$

is satisfied, where $\bar{\Omega}(iv_k)$ is the complex conjugate of $\Omega(iv_k)$. Given a general transfer function $\bar{W}_m(s)$, there exists a large class of functions ω that satisfies (87), even when $\bar{W}_m(s)$ is not SPR.

ω in Theorem 1 is not an independent variable but rather an internal variable of the nonlinear system in (84). Hence, it cannot be shown to be bounded or persistently exciting. If ω_* represents the signal corresponding to ω in the reference model, it can be made to satisfy (85) by the proper choice of the reference input. Expressing $\omega = \omega_* + \omega_e$, ω will also be bounded, persistently exciting, and satisfy (85) if ω_e is small. This can be achieved by choosing the initial conditions $e(t_0)$ and $\tilde{\theta}(t_0)$ in (84) to be sufficiently small. The conditions of Theorem 1 are then verified, and for a sufficiently small μ , exponential stability of the origin of (84) follows.

Theorem 1 provides conditions for exponential stability and instability when the solutions of the adaptive system are sufficiently close to the tuned solutions. These are very valuable in understanding the stability and instability mechanisms peculiar to adaptive control in the presence of different types of perturbations. Many of these results have been summarized and presented in a unified fashion in Anderson et al. (1986).

Cross-References

- [Adaptive Control: Overview](#)
- [History of Adaptive Control](#)
- [Nonlinear Adaptive Control](#)
- [Stochastic Adaptive Control](#)

Bibliography

- Anderson BDO, Bitmead RR, Johnson CR Jr, Kokotovic PV, Kosut RL, Mareels IM, Praly L, Riedle BD (1986) Stability of adaptive systems: passivity and averaging analysis. MIT, Cambridge
- Arnold VI (1988) Geometric methods in the theory of differential equations. Springer, New York
- Egardt B (1979) Stability of adaptive controllers. Springer, New York
- Hale JK (1969) Ordinary differential equations. Wiley–Interscience, New York
- Hussain H, Matsutani M, Annaswamy A, Lavretsky E (2013) Adaptive control of scalar plants in the presence of unmodeled dynamics. In: 11th IFAC international workshop, ALCOSP, Caen, July 2013
- Hussain HS, Yildiray Y, Matsutani M, Annaswamy AM, Lavretsky E (2017) Computable delay margins for adaptive systems with state variables accessible. IEEE Trans Autom Control 62:5039–5054
- Ioannou P, Sun J (2013) Robust adaptive control. Dover, Mineola
- Káráson SP, Annaswamy AM (1994) Adaptive control in the presence of input constraints. IEEE Trans Autom Control 39:2325–2330
- Khalil H (2001) Nonlinear systems, ch. 14.5. Prentice Hall, Upper Saddle River
- Kokotovic P, Riedle B, Praly L (1985) On a stability criterion for continuous slow adaptation. Syst Control Lett 6:7–14
- Kreisselmeier G, Narendra KS (1982) Stable model reference adaptive control in the presence of bounded disturbances. IEEE Trans Autom Control 27:1169–1175
- Krylov AN, Bogoliuboff NN (1943) Introduction to nonlinear mechanics. Princeton University Press, Princeton
- Lavretsky E (2011) Adaptive output feedback design using asymptotic properties of LQG/LTR controllers. IEEE Trans Autom Control 57:1587–1591
- Lavretsky E, Wise KA (2013) Robust adaptive control with aerospace applications. Springer, London
- Matsutani M (2013) Robust adaptive flight control systems in the presence of time delay. Ph.D. dissertation, Massachusetts Institute of Technology
- Matsutani M, Annaswamy A, Gibson T, Lavretsky E (2011) Trustable autonomous systems using adaptive control. In: 50th IEEE conference on decision and control and European control conference, Orlando
- Matsutani M, Annaswamy A, Lavretsky E (2012) Guaranteed delay margins for adaptive control of scalar plants. In: 2012 IEEE 51st annual conference on decision and control (CDC), Maui, pp 7297–7302
- Matsutani M, Annaswamy A, Lavretsky E (2013) Guaranteed delay margins for adaptive systems with state variables accessible. In: American control conference, Washington, DC, pp 3362–3369
- Narendra KS, Annaswamy AM (1987) A new adaptive law for robust adaptation without persistent excitation. IEEE Trans Autom Control 32:134–145
- Narendra KS, Annaswamy AM (2005) Stable adaptive systems. Dover, Mineola
- Narendra KS, Kudva P (1974) Stable adaptive schemes for system identification and control – parts I & II. IEEE Trans Syst Man Cybern 4:542–560
- Peterson B, Narendra K (1982) Bounded error adaptive control. IEEE Trans Autom Control 27:1161–1168
- Pomet J, Praly L (1992) Adaptive nonlinear regulation: estimation from the Lyapunov equation. IEEE Trans Autom Control 37(6):729–740
- Rohrs C, Valavani L, Athans M, Stein G (1985) Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics. IEEE Trans Autom Control 30(9):881–889
- Tsakalis K, Ioannou P (1987) Adaptive control of linear time-varying plants. Automatica 23(4):459–468

Robust Control in Gap Metric

Li Qiu¹ and Di Zhao²

¹Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong, China

²Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong SAR, China

Abstract

Robust control needs to start with a model of system uncertainty. What is a good uncertainty model? First it needs to capture the

possible system perturbations and uncertainties. Second it needs to be mathematically tractable. The gap metric was introduced by Zames and El-Sakkary for this purpose. Its study climaxed in an award-winning paper by Georgiou and Smith. A modified gap, called the ν -gap, was later discovered by Vinnicombe and was shown to have advantages. When the plant and controller communicate through a non-ideal bidirectional channel, cascaded two-port networks can be utilized to model such a channel whose uncertainties are measured by the $\mathcal{H}_\infty$ norm. With these descriptions of uncertainties in hand, robust stabilization issues can be nicely addressed.

Keywords

Gap metric · H-infinity control · ν -gap metric · Robust stabilization · Uncertain system · Networked control system · Two-port network

Introduction

The gap is rooted in mathematical literature for the purpose of measuring the distance between unbounded operators (Kato 1976). It is introduced to control theory by Zames and El-Sakkary (1980) to measure the distance between systems and subsequently to model an uncertain system, with the recognition that a possibly unstable system is simply a possibly unbounded operator. Here only continuous-time systems will be treated. Discrete-time systems can be treated in an analogous way. Let us identify a linear time-invariant (LTI) system with its transfer function. The set of m -input p -output finite-dimensional LTI systems is then identified with the set $\mathcal{R}^{p \times m}$ of $p \times m$ real rational matrices. Such a system can be considered as a linear operator from input space $\mathcal{H}_2^m$ to output space $\mathcal{H}_2^p$, defined by the input-output relation $y = Pu$. Here $\mathcal{H}_2$ is the collection of all bounded-energy signals $x(s)$ satisfying

$$\|x\|_2 := \sup_{\sigma > 0} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} |x(\sigma + j\omega)|^2 d\omega \right)^{1/2} < \infty.$$

This operator is possibly unbounded since for an input $u \in \mathcal{H}_2^m$, the corresponding output $y = Pu$ is not necessarily in $\mathcal{H}_2^p$. It is bounded if and only if P is stable, i.e., if and only if $P \in \mathcal{RH}_\infty^{p \times m}$, the set of $p \times m$ real rational matrices bounded over $\text{Re } s > 0$. In this case, the induced operator norm is the $\mathcal{H}_\infty$ norm of P :

$$\|P\|_\infty = \sup_{\text{Re } s > 0} \bar{\sigma}[P(s)] = \sup_{\omega \in \mathbb{R}} \bar{\sigma}[P(j\omega)].$$

No matter whether or not P is stable, we define the graph of P as

$$\mathcal{G}_P = \left\{ \begin{bmatrix} u \\ y \end{bmatrix} \in \mathcal{H}_2^{m+p} : y = Pu \right\},$$

i.e., the graph is the set of all finite energy input-output pairs. It is easy to see that $\mathcal{G}_P$ is a linear subspace of $\mathcal{H}_2^{m+p}$, and a little more effort shows that it is closed. Hence it uniquely corresponds to a bounded linear operator on $\mathcal{H}_2^{m+p}$, called the orthogonal projection onto $\mathcal{G}_P$, denoted by $\Pi_{\mathcal{G}_P}$. Now with two systems $P_1, P_2 \in \mathcal{R}^{p \times m}$, the gap in between is defined as

$$\delta(P_1, P_2) = \|\Pi_{\mathcal{G}_{P_1}} - \Pi_{\mathcal{G}_{P_2}}\|.$$

That the gap is a metric in $\mathcal{R}^{p \times m}$ follows from the fact that the induced operator norm used above defines a metric on the set of all orthogonal projections.

With the gap between two systems, an uncertain system described by the gap is simply a gap metric ball with a center P , called the nominal system, and a radius r , qualifying the amount of uncertainty:

$$\mathcal{B}(P, r) = \{\tilde{P} \in \mathcal{R}^{p \times m} : \delta(\tilde{P}, P) < r\}.$$

Gap Computation and Robust Stabilization

With the basic definitions constructed above, the following questions are then asked:

Computation: How can the gap between two systems be computed?

Analysis: How much stability robustness does a stable feedback system have against gap uncertainty in the plant or in both the plant and the controller?

Synthesis: How can a feedback controller be designed so that the feedback system has optimal robustness against gap uncertainty?

For the question on computation, it is rather easy to see that if P_1 and P_2 are static, also said to be memoryless, systems, i.e., $P_1(s) = K_1$ and $P_2(s) = K_2$, then

$$\delta(P_1, P_2) = \bar{\sigma}[(I + K_1 K_1^T)^{-1/2}(K_1 - K_2)(I + K_2^T K_2)^{-1/2}].$$

In the single-input single-output case, this is exactly the chordal distance between two numbers K_1 and K_2 . Hence the expression above generalizes the chordal distance between two complex numbers to constant matrices. What if P_1 and P_2 are dynamic systems? It is not until (Georgiou 1988) when the computation of the gap was settled by using the coprime factorization.

For each $P \in \mathcal{R}^{p \times m}$, there are

$$\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}, \begin{bmatrix} M & U \\ N & V \end{bmatrix} \in \mathcal{RH}_\infty^{(m+p) \times (m+p)}$$

such that $P = NM^{-1} = \tilde{M}^{-1}\tilde{N}$ and

$$\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} M & U \\ N & V \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}.$$

These matrices are said to give a doubly coprime factorization of P . Also $P = NM^{-1}$ and $P = \tilde{M}^{-1}\tilde{N}$ are said to be right and left coprime factorizations, respectively. In the doubly coprime factorization, we can further require

$$M^T(-s)M(s) + N^T(-s)N(s) = I \text{ and} \\ \tilde{M}(s)\tilde{M}^T(-s) + \tilde{N}(s)\tilde{N}^T(-s) = I.$$

In this case, the coprime factorizations are said to be normalized.

Theorem 1 (Computation of the gap) Let $P_i = N_i M_i^{-1}$, $i = 1, 2$, be normalized right coprime factorizations. Then

$$\delta(P_1, P_2) = \max \{\delta(P_1, P_2), \delta(P_2, P_1)\},$$

where

$$\delta(P_1, P_2) = \inf_{Q \in \mathcal{RH}_\infty^{m \times m}} \left\| \begin{bmatrix} M_1 \\ N_1 \end{bmatrix} - \begin{bmatrix} M_2 \\ N_2 \end{bmatrix} Q \right\|_\infty.$$

The problems of finding the two infima above are $\mathcal{H}_\infty$ model-matching problems, special forms of $\mathcal{H}_\infty$ control problems. See article ► [“Optimal Control via Factorization and Model Matching”](#) and article ► [“H-Infinity Control”](#) in this encyclopedia. In principle, they can be solved using the standard ways.

The analysis and synthesis questions are satisfactorily answered by Georgiou and Smith (1990). Let us consider the feedback system shown in Fig. 1.

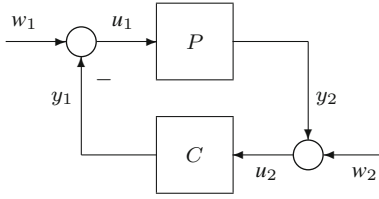
Such a feedback system is denoted by a plant and controller pair or simply a feedback pair $P \# C$ with $P \in \mathcal{R}^{p \times m}$ and $C \in \mathcal{R}^{m \times p}$. This closed-loop system is stable if the transfer matrix from $\begin{bmatrix} w_1 \\ -w_2 \end{bmatrix}$ to $\begin{bmatrix} u_1 \\ y_2 \end{bmatrix}$, nicknamed the Gang of Four matrix by Åström and Murray (2008),

$$P \# C = \begin{bmatrix} (I + PC)^{-1} & (I + PC)^{-1}C \\ C(I + PC)^{-1} & P(I + PC)^{-1}C \end{bmatrix} \\ = \begin{bmatrix} I \\ P \end{bmatrix} (I + PC)^{-1} \begin{bmatrix} I & C \end{bmatrix}$$

is stable. Here $P \# C$ denotes both the feedback system and its Gang of Four matrix.

Theorem 2 (Stability margin) Assume $P \# C$ is a stable closed-loop system. All feedback systems $\tilde{P} \# C$ with $\tilde{P} \in \mathcal{B}(P, r)$ are stable if and only if $r \leq \|P \# C\|_\infty^{-1}$.

It follows from Theorem 2 that $\|P \# C\|_\infty^{-1}$ is the stability margin of the closed-loop system in Fig. 1. The natural design problem is



Robust Control in Gap Metric, Fig. 1 An uncertain feedback system

then to design a controller C for a given P such that $\|P \# C\|_{\infty}^{-1}$ is maximized or equivalently $\|P \# C\|_{\infty}$ is minimized. Such a problem again is an $\mathcal{H}_{\infty}$ control problem, which is the topic of article ▶ “[H-Infinity Control](#)” in this encyclopedia. It is realized by Georgiou and Smith (1990) that this particular $\mathcal{H}_{\infty}$ control problem has some unique features. Let P have a normalized doubly coprime factorization, and let

$$R(s) = M^T(-s)U(s) + N^T(-s)V(s).$$

Then

$$\inf_C \|P \# C\|_{\infty} = \left(1 + \inf_{Q \in \mathcal{RH}_{\infty}^{m \times p}} \|R - Q\|_{\infty} \right)^{1/2}.$$

The minimization over Q above is a one-block $\mathcal{H}_{\infty}$ model-matching problem. It can be solved rather easily, much more easily than the $\mathcal{H}_{\infty}$ model-matching problem arising in the computation of gap. After finding an optimal Q , an optimal controller is obtained as

$$C = -(U - MQ)(V - NQ)^{-1}.$$

Qiu and Davison (1992a) extended Theorem 2 to the case when both the plant and controller are subject to uncertainty.

Theorem 3 (The arcsin theorem) Assume $P \# C$ is a stable closed-loop system. All feedback systems $\tilde{P} \# \tilde{C}$ with $\tilde{P} \in \mathcal{B}(P, r_P)$ and $\tilde{C} \in \mathcal{B}(C, r_C)$ are stable if and only if

$$\arcsin r_P + \arcsin r_C \leq \arcsin \|P \# C\|_{\infty}^{-1}.$$

This theorem further strengthens the role of $\|P \# C\|_{\infty}^{-1}$ as the robust stability margin of the feedback system $P \# C$.

The ν -Gap

Partly because of the lack of efficient ways in computing the gap, there were efforts in seeking other metrics on $\mathcal{R}^{p \times m}$ with better numerical and analytical properties. Several such metrics were proposed, including the graph metric by Vidyasagar (1984), pointwise gap metric by Qiu and Davison (1992b), and ν -gap metric by Vinnicombe (1993). The winner is the ν -gap which is defined by ingeniously exploring the special structures and properties of rational matrices in $\mathcal{R}^{p \times m}$. For $P_1, P_2 \in \mathcal{R}^{p \times m}$, let $P_i = N_i M_i^{-1}$, $i = 1, 2$, be normalized right coprime factorizations. Define the ν -gap metric as

$$\delta_{\nu}(P_1, P_2) = \sup_{\omega \in \mathbb{R}} \bar{\sigma} \left\{ [I + P_1(j\omega)P_1(j\omega)^*]^{-1/2} [P_1(j\omega) - P_2(j\omega)][I + P_2(j\omega)^*P_2(j\omega)]^{-1/2} \right\}$$

if $\det[M_2^T(-s)M_1(s) + N_2^T(-s)N_1(s)]$ has equal number of unstable poles and zeros and $\delta_{\nu}(P_1, P_2) = 1$ otherwise. Apparently ν -gap is easier to compute than the gap. When the pole-zero number condition is satisfied, the ν -gap is the peak of the chordal distance between the system frequency responses. The ν -gap is no greater than the gap, i.e.,

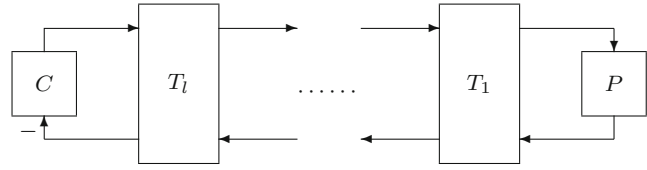
$$\delta_{\nu}(P_1, P_2) \leq \delta(P_1, P_2).$$

Hence the ν -gap ball

$$\mathcal{B}_{\nu}(P, r) = \{\tilde{P} \in \mathcal{R}^{p \times m} : \delta_{\nu}(\tilde{P}, P) < r\}$$

is a superset of the gap ball with the same center and radius. Theorems 2 and 3 can be restated with the gap balls $\mathcal{B}$ replaced by the new gap balls $\mathcal{B}_{\nu}$. Consequently the restated Theorems 2 and 3 are less conservative than the original versions for the gap. The optimal robust stabilization problems

Robust Control in Gap Metric, Fig. 2 An uncertain cascaded two-port networked control system



for the gap and the ν -gap are the same: design C to maximize $\|P \# C\|_{\infty}^{-1}$ for a given P .

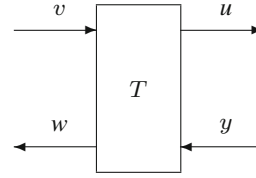
Networked Robust Stabilization

In the era of network, a feedback system is expected to incorporate communication networks that are typically noisy and uncertain. Recently, two-port networks are adopted to model a bidirectional communication channel between the plant and controller in Zhao et al. (2020). As shown in Fig. 2, a networked control system (NCS) is described as a feedback interconnection of a plant P and a controller C communicating through a bidirectional channel modeled by cascaded two-port networks $\{T_k\}_{k=1}^l$. This model is sufficiently rich to capture various properties of a real-world communication channel subject to distortions, interferences, and even malicious attacks.

The network T in Fig. 3 has two external ports, with one port composed of signals u, y and the other of signals v, w . Naturally, we regard u, v as the downlink signals both with dimension m and w, y as the uplink signals both with dimension p . We describe a two-port network with its transmission (representation) matrix defined as

$$T = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \in \mathcal{R}^{(m+p) \times (m+p)} \quad \text{and} \quad \begin{bmatrix} v \\ w \end{bmatrix} = T \begin{bmatrix} u \\ y \end{bmatrix}.$$

Here T stands for both the two-port network itself and its transmission matrix. It is worth noting that the transmission representation of a two-port network is also called the chain-scattering representation and has been used to solve the $\mathcal{H}_{\infty}$ control problems by Kimura (1996).



Robust Control in Gap Metric, Fig. 3 A two-port network T

When a two-port network describes ideal communication, it admits an identity transmission matrix; i.e.,

$$T = \begin{bmatrix} I_m & 0 \\ 0 & I_p \end{bmatrix}.$$

When a two-port network admits both distortions and interferences, its transmission matrix can be modeled by

$$T = I + \Delta = \begin{bmatrix} I_m + \Delta_{\div} & \Delta_{-} \\ \Delta_{+} & I_p + \Delta_{\times} \end{bmatrix},$$

$$\text{where } \Delta = \begin{bmatrix} \Delta_{\div} & \Delta_{-} \\ \Delta_{+} & \Delta_{\times} \end{bmatrix} \in \mathcal{RH}_{\infty}^{(m+p) \times (m+p)}.$$

The four-block matrix Δ is called the uncertainty quartet. The subscripts $+, -, \times, \div$ for Δ , which represent different types of uncertainties, were firstly used by Halsey and Glover (2005) for the purpose to describe various connections of uncertainties vividly.

The stability of an NCS is defined as usual. The two-port NCS in Fig. 2 is said to be stable if for all injected energy-bounded exogenous signals, the endogenous signals are energy-bounded. A necessary and sufficient robust stability condition of the two-port NCS was obtained by Zhao et al. (2020), which is a generalization of Theorem 3.

Theorem 4 (The information constrained arcsin theorem) Assume $P \# C$ is a stable

closed-loop system. For $r_P, r_C, r_k \in [0, 1)$, the NCS in Fig. 2 is stable for all $\tilde{P} \in \mathcal{B}(P, r_P)$, $\tilde{C} \in \mathcal{B}(C, r_C)$ and $\Delta_k \in \mathcal{RH}_\infty^{(m+p) \times (m+p)}$ with $\|\Delta_k\|_\infty \leq r_k, k = 1, 2, \dots, l$, if and only if

$$\arcsin r_P + \arcsin r_C + \sum_{k=1}^l \arcsin r_k < \arcsin \|P \# C\|_\infty^{-1}.$$

Clearly, the above theorem reduces to Theorem 3 when $r_k = 0, k = 1, 2, \dots, l$. It is noteworthy that when the gap balls in the above theorem are replaced by the ν -gap balls, the robust stability condition is still valid.

Summary and Future Directions

The gap, as well as the ν -gap, and the associated robust control theory can be extended to infinite dimensional systems as in Georgiou and Smith (1992) and Ball and Sasane (2012); time-varying systems as in Foias et al. (1993) and Feintuch (1998); and even nonlinear systems as in Georgiou and Smith (1997), Anderson et al. (2002), James et al. (2005), and Bian and French (2005), in varying degrees. There are still research opportunities in these extensions. The use of normalized coprime factorizations seems to be an obstacle in these extensions.

For a plant P , the controller optimizing $\|P \# C\|_\infty$ is not always a good controller. This gives another example where “optimal” is not always equal to “good.” One reason is that the actual plant uncertainty cannot necessarily be well described by a gap ball or a ν -gap ball. Another reason is that performance issues other than the stability robustness, such as tracking and disturbance rejection, are not taken into account in the optimization. The actual plant uncertainty might be better described by a gap ball centered at a frequency-shaped plant $\tilde{P} = W_o P W_i$ where W_i and W_o are real rational weighting matrices which can also be chosen to address tracking and disturbance rejection requirements. In this case, an optimal controller $\tilde{C}$ can then be designed to

optimize the transfer matrix $\tilde{P} \# \tilde{C}$ corresponding to the shaped plant $\tilde{P}$. Finally $C = W_i \tilde{C} W_o$ is used as a designed controller for the original plant P . With the proper choice of W_i and W_o , it is more likely that a good controller will result. This loop-shaping design method was proposed in McFarlane and Glover (1992) and further developed in Vinnicombe (2001).

In the process of obtaining the arcsin theorems for standard closed-loop systems and two-port NCSs, it has been realized that the gap and even more so the ν -gap are closely related to the canonical angles between linear subspaces. In fact the gap is the sin of the largest canonical angle between certain subspaces, and the largest canonical angle itself is also a metric, a better one in some geometric sense. For the latest development on canonical angles, see Qiu et al. (2008) and Zhang and Qiu (2010).

In addition to the effort in deepening and expanding the notion of gap and its use in robust control, there is also effort in making it more accessible and more closely related to classical frequency response analysis; see Qiu and Zhou (2013) and Zhao et al. (2018). It again appears that the use of coprime factorizations in the current theory is hindering this effort. Hence, circumventing the use of coprime factorizations, normalized or not, in the development of the gap would help its extension and popularization.

Cross-References

- [H-Infinity Control](#)
- [Optimal Control via Factorization and Model Matching](#)
- [Robust Synthesis and Robustness Analysis Techniques and Tools](#)

Recommended Reading

The most authoritative work on gap, ν -gap, and the associated robust stabilization theory is the

comprehensive monograph (Vinnicombe 2001). This theory is inherently an input-output frequency domain theory. However many related computations, such as those of doubly normalized coprime factorizations, $\mathcal{H}_\infty$ model matching, and the optimal controller synthesis, can be done using state-space formulas and further using MATLAB programs. Vinnicombe (2001) contains a list of such state-space formulas. This theory provides a good example of the once popular and successful philosophy behind the linear multivariable control theory: thinking in terms of transfer functions and computing in term of state-space equations.

The system and control background needed to understand and study the gap, the ν -gap, and robust stabilization, in particular the coprime factorization and frequency domain stabilization theory, can be found in Vidyasagar (1985).

The book Zhou and Doyle (1998) also contains considerable content on gap based robust control.

Bibliography

- Anderson BDO, Brinsmead TS, De Bruyne F (2002) The Vinnicombe metric for nonlinear operators. *IEEE Trans Autom Control* 47:1450–1465
- Åström KJ, Murray RM (2008) *Feedback fundamentals – control & dynamical systems*. Princeton University Press, Princeton
- Ball JA, Sasane AJ (2012) Extension of the ν -metric. *Compl Anal Oper Theory* 6:65–89
- Bian W, French M (2005) Graph topologies, gap metrics, and robust stability for nonlinear systems. *SIAM J Control Optim* 44:418–443
- El-Sakkary AK (1985) The gap metric: robustness of stabilization of feedback systems. *IEEE Trans Autom Control* 30:240–247
- Feintuch A (1998) *Robust control theory in Hilbert space*. Springer, New York
- Foias C, Georgiou TT, Smith MC (1993) Robust stability of feedback systems: a geometric approach using the gap metric. *SIAM J Control Optim* 31:1518–1537
- Georgiou TT (1988) On the computation of the gap metric. *Syst Control Lett* 11:253–257
- Georgiou TT, Smith MC (1990) Optimal robustness in the gap metric. *IEEE Trans Autom Control* 35:673–687
- Georgiou TT, Smith MC (1992) Robust stabilization in the gap metric: controller design for distributed plants. *IEEE Trans Autom Control* 37:1133–1143
- Georgiou TT, Smith MC (1997) Robustness analysis of nonlinear feedback systems: an input-output approach. *IEEE Trans Autom Control* 42:1200–1221
- Glover K, McFarlane DC (1989) Robust stabilization of normalized coprime factor plant descriptions with $\mathcal{H}_\infty$ bounded uncertainties. *IEEE Trans Autom Control* 34:821–830
- Halsey KM, Glover K (2005) Analysis and synthesis of nested feedback systems. *IEEE Trans Autom Control* 50:984–996
- James MR, Smith MC, Vinnicombe G (2005) Gap metrics, representations and nonlinear robust stability. *SIAM J Control Optim* 43:1535–1582
- Kato T (1976) *Perturbation theory for linear operators*, 2nd edn. Springer, Berlin
- Kimura H (1996) *Chain-scattering approach to $\mathcal{H}_\infty$ control*. Springer Science & Business Media, New York
- McFarlane DC, Glover K (1992) A loop shaping design procedure using $\mathcal{H}_\infty$ -synthesis. *IEEE Trans Autom Control* 37:759–769
- Qiu L, Davison EJ (1992a) Feedback stability under simultaneous gap metric uncertainties in plant and controller. *Syst Control Lett* 18:9–22
- Qiu L, Davison EJ (1992b) Pointwise gap metrics on transfer matrices. *IEEE Trans Autom Control* 37:741–758
- Qiu L, Zhou K (2013) Preclassical tools for postmodern control. *IEEE Control Syst Mag* 33(4):26–38
- Qiu L, Zhang Y, Li CK (2008) Unitarily invariant metrics on the Grassmann space. *SIAM J Matrix Anal* 27:501–531
- Vidyasagar M (1984) The graph metric for unstable plants and robustness estimates for feedback stability. *IEEE Trans Autom Control* 29:403–418
- Vidyasagar M (1985) *Control system synthesis: a factorization approach*. MIT, Cambridge
- Vinnicombe G (1993) Frequency domain uncertainty and the graph topology. *IEEE Trans Autom Control* 38:1371–1383
- Vinnicombe G (2001) Uncertainty and feedback: $\mathcal{H}_\infty$ loop-shaping and the ν -gap metric. Imperial Collage Press, London
- Zames G, El-Sakkary AK (1980) Unstable systems and feedback: the gap metric. In: *Proceedings of the 16th Allerton conference*, Illinois, pp 380–385
- Zhang Y, Qiu L (2010) From subadditive inequalities of singular values to triangular inequalities of canonical angles. *SIAM J Matrix Anal Appl* 31:1606–1620
- Zhao D, Chen C, Khong SZ, Qiu L (2018) Robust control against uncertainty quartet: a polynomial approach. In: Başar T (ed) *Uncertainty in complex networked systems*. Springer International Publishing, Cham, pp 149–178
- Zhao D, Qiu L, Gu G (2020, to appear) Stabilization of two-port networked systems with simultaneous uncertainties in plant, controller, and communication channels. *IEEE Trans Autom Control*. <https://doi.org/10.1109/TAC.2019.2918121>
- Zhou K, Doyle JC (1998) *Essentials of robust control*. Prentice Hall, Upper Saddle River

Robust Control of Infinite-Dimensional Systems

Hitay Özbay

Department of Electrical and Electronics
Engineering, Bilkent University, Ankara, Turkey

Abstract

Basic robust control problems are studied for the feedback systems where the underlying plant model is infinite dimensional. The $\mathcal{H}_\infty$ optimal controller formula is given for the mixed sensitivity minimization problem with rational weights. Key steps of the numerical computations required to determine the controller parameters are illustrated with an example where the plant model includes time delay terms.

Keywords

Coprime factorizations · Direct design methods · Inner-outer factorizations

Introduction

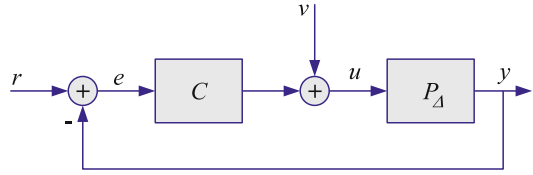
Robust control deals with the feedback system shown in Fig. 1, where P_Δ represents the uncertain physical plant and C is a fixed controller to be designed.

Here, it is assumed that the controller and the plant are linear time-invariant (LTI) systems and they are represented by their transfer functions. Furthermore, P_Δ satisfies the following conditions:

$$P_\Delta(s) = P(s) + \Delta(s)$$

where P is the nominal plant model, with $P(s)$ and $P_\Delta(s)$ having the same number of poles in $\mathbb{C}_+$; and there is a known uncertainty bound $W(s)$ satisfying

$$|\Delta(j\omega)| < |W(j\omega)| \quad \forall \omega \in \mathbb{R}.$$



Robust Control of Infinite-Dimensional Systems, Fig. 1 Feedback system $\mathcal{F}(C, P_\Delta)$ with fixed controller C and uncertain plant P_Δ

Definition 1 All P_Δ satisfying the above conditions are said to be in the set of uncertain plants $\mathcal{P}$, which is characterized by the given functions $P(s)$ and $W(s)$.

Depending on physical system modeling, other forms of uncertainty representations can be more convenient than the additive unstructured uncertainty model taken here; see, e.g., Doyle et al. (1992), Özbay (2000), and Zhou et al. (1996) for the examples of multiplicative, coprime factor, parametric, and structured uncertainty descriptions. Note that for notational convenience and simplicity of the presentation, single-input-single-output (SISO) plants are considered here; for extensions to multi-input-multi-output (MIMO) plants, see, e.g., Curtain and Zwart (1995).

When the plant under consideration is infinite dimensional, the transfer function $P(s)$ is irrational, i.e., it cannot be expressed as a ratio of two polynomials (it does not admit a finite-dimensional state-space representation). Typical examples of such systems are spatially distributed parameter systems modeled by partial differential equations, fractional-order systems, and systems with time delays. The reader is referred to Curtain and Morris (2009) for examples of transfer functions of distributed parameter systems. There are many interesting industrial applications where fractional-order transfer functions are used for modeling and control, see, e.g., Monje et al. (2010); typically, such functions are rational in s^α , where α is a rational number in the open interval $(0, 1)$. Transfer functions of systems with time delays involve terms like e^{-hs} where $h > 0$ is the delay; see Sipahi et al. (2011) for various real-life examples where time-delay mod-

els appear. Transfer functions considered here are functions of the complex variable s with real coefficients, so $\overline{P(\bar{s})} = P(s)$ where $\bar{s}$ denotes the complex conjugate of s .

Definition 2 A linear time-invariant system H is said to be *stable* if its transfer function $H(s)$ is bounded and analytic in $\mathbb{C}_+$. In this case, the *system norm* is

$$\|H\| = \|H\|_\infty = \sup_{\operatorname{Re}(s) > 0} |H(s)|,$$

which is equivalent to the *energy amplification* through the system H ; see Doyle et al. (1992) and Foias et al. (1996).

Definition 2 is sometimes called the $\mathcal{H}_\infty$ -stability, and in this setting, the set of all stable plants is the function space $\mathcal{H}_\infty$. It is worth noting that for infinite-dimensional systems, there are other definitions of stability (Curtain and Zwart 1995; Desoer and Vidyasagar 2009), leading to different measures of the system norm.

Robust Control Design Objectives

Let $\mathcal{F}(C, P_\Delta)$ denote the feedback system shown in Fig. 1. This system is said to be *robustly stable* if all the transfer functions from external inputs (r, v) to internal signals (e, u) are in $\mathcal{H}_\infty$ for all $P_\Delta \in \mathcal{P}$. In the controller design, robust stability of the feedback system is the primary constraint.

The feedback system $\mathcal{F}(C, P_\Delta)$ is robustly stable if and only if the following conditions hold; see, e.g., Doyle et al. (1992) and Foias et al. (1996),

$$(a) \quad S, CS, PS \in \mathcal{H}_\infty, \text{ where } S = (1 + PC)^{-1},$$

and

$$(b) \quad \|WCS\|_\infty \leq 1.$$

In order to illustrate these design constraints for robustly stabilizing controller, as an example, consider a strictly proper stable plant, i.e.,

$$P \in \mathcal{H}_\infty \quad \text{with} \quad \lim_{|s| \rightarrow \infty} |P(s)| = 0.$$

In this case, all controllers in the form $C = Q/(1 - PQ)$ satisfy condition (a) for any $Q \in \mathcal{H}_\infty$ (moreover, any controller C satisfying (a) must be in this form for some $Q \in \mathcal{H}_\infty$). Now consider a rational $W(s)$ with a stable Q such that $|Q(j\omega)|$ is a continuous function of $\omega \in \mathbb{R}$. Then, condition (b) becomes

$$\|WQ\|_\infty \leq 1 \iff |Q(j\omega)| \leq |W(j\omega)|^{-1} \quad \forall \omega \in \mathbb{R}.$$

So, whenever the modeling uncertainty is “large” on a frequency band $\omega \in \Omega$, the magnitude of Q should be “small” in this region.

When the plant is unstable, say $p \in \mathbb{C}_+$ is a pole of $P(s)$ of multiplicity one, conditions (a) and (b) impose a restriction on the controller, which leads to

$$1 \geq \|WCS\|_\infty \geq \left| \frac{W(p)}{N(p)} \right| \quad \text{where } N(p) = \lim_{s \rightarrow p} \frac{(s - p)}{(s + \bar{p})} P(s).$$

So, a necessary condition, for (b) to hold in this case, is $|W(p)| \leq |N(p)|$, which means that the modeling uncertainty at the unstable pole of the plant should be small enough for the existence of a robustly stabilizing controller. This is one of the fundamental quantifiable limitations of feedback systems with unstable plants; see Stein (2003) for further discussions on other limitations.

Many other performance-related design objectives, such as reference tracking and disturbance attenuation, are captured by the *sensitivity minimization*, which is defined as finding a controller satisfying (a) and achieving

$$(c) \quad \|W_1 S\|_\infty \leq \gamma$$

for the smallest possible $\gamma > 0$, for a given stable sensitivity weight $W_1(s)$. Selection of W_1 depends on the class of reference signals and disturbances considered; see Doyle et al. (1992), Özbay (2000), and Stein (2003) for general guidelines. Stability robustness and performance objectives defined above can be

blended to define a single $\mathcal{H}_\infty$ -optimization problem, known as the *mixed sensitivity minimization*: given W_1 , W_2 , P , find a controller C satisfying (a) and achieving

$$(d) \left\| \begin{bmatrix} W_1 S \\ W_2 T \end{bmatrix} \right\|_\infty := \sup_{\operatorname{Re}(s) > 0} \left(|W_1(s)S(s)|^2 + |W_2(s)T(s)|^2 \right)^{\frac{1}{2}} \leq \gamma$$

for the smallest possible $\gamma > 0$, where $T(s) := 1 - S(s)$ and $W_2(s)$ represents the multiplicative uncertainty bound, with $|W_2(j\omega)| = |W(j\omega)|/|P(j\omega)|$, $\forall \omega \in \mathbb{R}$. The smallest achievable γ is the optimal performance level γ_{opt} , and the corresponding controller is denoted by C_{opt} . Typically, when P is infinite dimensional, so is the optimal controller.

Design Methods

Approximation of the Plant

One possible way to design a robust controller for an infinite-dimensional plant P is to design a robust controller C_a for an approximate finite-dimensional plant P_a ; (for a frequency domain approximation technique for infinite-dimensional systems, see Gu et al. (1989)). When W_1 , W_2 , and P_a are finite dimensional, standard state-space methods, Zhou et al. (1996), can be used to find a $\mathcal{H}_\infty$ controller C_a achieving

$$\left\| \begin{bmatrix} W_1 S_a \\ W_2 T_a \end{bmatrix} \right\|_\infty \leq \gamma_a$$

for the smallest possible γ_a , where $S_a := (1 + P_a C_a)^{-1}$ and $T_a = (1 - S_a)$. Then, the controller $C = C_a$ satisfies (a) and achieves the performance objective (d) with

$$\gamma = (\gamma_a + \varepsilon) \frac{1}{1 - \varepsilon}, \quad \varepsilon := \|C_a S_a (P - P_a)\|_\infty,$$

where it is assumed that the approximation of the plant is made in such a way that $\varepsilon < 1$. Clearly, if $\gamma_a \rightarrow \gamma_{\text{opt}}$ as $\varepsilon \rightarrow 0$, then $\gamma \rightarrow \gamma_{\text{opt}}$ as

$\varepsilon \rightarrow 0$. The conditions under which $\gamma_a \rightarrow \gamma_{\text{opt}}$ are discussed in Morris (2001).

Direct Design Methods

The classical two-Riccati equation approach, Zhou et al. (1996), developed for finite-dimensional systems, has been extended to various classes of infinite-dimensional systems by using the state-space techniques where semigroup theory plays an important role; see van Keulen (1993) for further details.

In order to illustrate some of the key steps of a frequency domain method developed in Foias et al. (1996), consider a specific example where the plant is given as

$$P(s) = \frac{(s-1)(s+2)e^{-hs}}{(s^2+2s+2)(s+1-3e^{-2hs})}, \quad h = \ln(2) \approx 0.693. \quad (1)$$

First, compute the location of the poles in $\mathbb{C}_+$ using available numerical tools for finding the roots of quasi-polynomials; see, e.g., Sipahi et al. (2011) for references. For the simple example chosen here, $P(s)$ has only one pole in $\mathbb{C}_+$, at $s = 0.5$ (for larger values of h , the number of unstable poles of P may be higher). Now, the plant can be factored as follows:

$$P(s) = \frac{M_N(s)N_O(s)}{M_D(s)} \quad (2)$$

where

$$M_N(s) = \frac{s-1}{s+1} e^{-hs} \quad M_D(s) = \frac{s-0.5}{s+0.5}$$

are all-pass (inner) transfer functions and

$$N_O(s) = \frac{(s+2)(s+1)}{(s^2+2s^2+2)(s+0.5)} \left(\frac{s-0.5}{s+1-3e^{-2hs}} \right)$$

is a minimum-phase (outer) transfer function. Note that

$$\frac{s - 0.5}{s + 1 - 3e^{-2hs}} = \frac{1}{1 + H_F(s)},$$

$$H_F(s) = 1.5 \frac{1 - e^{-2h(s-0.5)}}{s - 0.5}. \quad (3)$$

The impulse response of H_F is $h_F(t) = 1.5e^{t/2}$ when $t \in [0, 2h]$ and $h_F(t) = 0$ otherwise. Stability of N_O can also be verified from the Nyquist graph of H_F . Also, note that $N_O(s)$ can be factored as $N_O(s) = N_1(s)N_2(s)$ where

$$N_1(s) = \frac{(s+2)(s+1)}{(s^2+2s^2+2)} \left(\frac{1}{1+H_F(s)} \right),$$

$$N_2(s) = \frac{1}{s+0.5}$$

with $N_1, N_1^{-1} \in \mathcal{H}_\infty$, and N_2 is finite-dimensional (first order in this example).

The above steps illustrate *coprime factorizations* and *inner-outer factorizations* for systems with time delays (retarded case). For systems represented by PDEs or integrodifferential equations, plant transfer function can be factored similarly, provided that the poles and zeros in $\mathbb{C}_+$ can be computed numerically.

When the plant is in the form (2) given above and the weights W_1 and W_2 are rational, the optimal performance level and the corresponding optimal controller are obtained by the following procedure (see Foias et al. (1996) for details).

- *Controller parameterization* transforms the mixed sensitivity minimization to a problem of finding the smallest $\gamma > 0$ for which there exists $Q \in \mathcal{H}_\infty$ such that

$$\left\| \begin{bmatrix} W_1 \\ 0 \end{bmatrix} - \begin{bmatrix} W_1 N_2 \\ -W_2 N_2 \end{bmatrix} M_N(R + M_D Q) \right\|_\infty \leq \gamma$$

where $R(s)$ is a rational function (whose order is one less than the order of M_D) satisfying certain interpolation conditions at the zeros of $M_D(s)$.

- A *spectral factorization* determines $W_0 \in \mathcal{H}_\infty$ such that $W_0^{-1} \in \mathcal{H}_\infty$ and

$$\begin{aligned} & \left(|W_1(j\omega)|^2 + |W_2(j\omega)|^2 \right) |N_2(j\omega)|^2 \\ &= |W_0(j\omega)|^2 \quad \forall \omega \in \mathbb{R}, \end{aligned}$$

(here, it is assumed that $W_2 N_2$ and $(W_2 N_2)^{-1}$ are in $\mathcal{H}_\infty$).

- By using the norm-preserving property of the unitary matrices and the *commutant lifting theorem*, it has been shown that

$$\gamma_{\text{opt}} = \left\| \begin{bmatrix} \Gamma \\ \Upsilon \end{bmatrix} \right\|$$

where Γ is the *Hankel operator* whose symbol is

$$\begin{aligned} & M_D(-s) \left(M_N(-s) W_0^{-1}(-s) N_2(-s) \right. \\ & \left. W_1(-s) W_1(s) - W_0(s) R(s) \right) \end{aligned}$$

and Υ is the *Toeplitz operator* whose symbol is $W_1(s) W_2(s) N_2(s) W_0^{-1}(s)$. Moreover, under mild technical assumptions, the optimal controller is obtained from a nonzero $\psi_o \in \mathcal{H}_2$ satisfying

$$\left(\gamma_{\text{opt}}^2 - (\Gamma^* \Gamma + \Upsilon^* \Upsilon) \right) \psi_o = 0$$

The operator $(\Gamma^* \Gamma + \Upsilon^* \Upsilon)$ is in the form of a *skew-Toeplitz operator* that gives the name to this approach. See Foias et al. (1996) for a detailed exposition.

Optimal $\mathcal{H}_\infty$ -Controller

The above steps have been implemented, and the final optimal controller expression has been obtained in a simplified form described below.

Let $\alpha_1, \dots, \alpha_\ell \in \mathbb{C}_+$ be the zeros of $M_D(s)$, i.e., unstable poles of the plant (for simplicity of the exposition, they are assumed to be distinct). The sensitivity weight can be written as $W_1(s) = n W_1(s)/d W_1(s)$, for two coprime polynomials $n W_1$ and $d W_1$ with $\deg(n W_1) \leq \deg(d W_1) =: n_1 \geq 1$. Define

$$E_\gamma(s) := \left(\frac{W_1(-s) W_1(s)}{\gamma^2} - 1 \right)$$

and let $\beta_1, \dots, \beta_{2n_1}$ be the zeros of $E_\gamma(s)$, enumerated in such a way that $-\beta_{n_1+k} = \beta_k \in \overline{\mathbb{C}}_+$, for $k = 1, \dots, n_1$. For notational convenience, assume that the zeros of E_γ are distinct for $\gamma = \gamma_{\text{opt}}$.

Now define a rational function depending on $\gamma > 0$ and the weights W_1 and W_2 ,

$$F_\gamma(s) := \gamma \frac{dW_1(-s)}{nW_1(s)} G_\gamma(s)$$

where $G_\gamma \in \mathcal{H}_\infty$ is an outer function determined from the spectral factorization

$$G_\gamma(-s)G_\gamma(s) = \left(1 + \frac{W_2(-s)W_2(s)}{W_1(-s)W_1(s)} - \frac{W_2(-s)W_2(s)}{\gamma^2}\right)^{-1}.$$

With the above definitions, under certain mild conditions (satisfied generically in most practical cases), the optimal controller can be expressed as

$$C_{\text{opt}}(s) = \frac{E_\gamma(s)M_D(s)F_\gamma(s)L(s)}{1 + M_N(s)F_\gamma(s)L(s)} N_O^{-1}(s) \quad (4)$$

where $\gamma = \gamma_{\text{opt}}$ and $L(s) = L_2(s)/L_1(s)$ with polynomials L_1 and L_2 , of degree $n_1 + \ell - 1$, determined from the interpolation conditions:

$$L_1(\beta_k) + M_N(\beta_k)F_\gamma(\beta_k)L_2(\beta_k) = 0$$

$$k = 1, \dots, n_1$$

$$L_1(\alpha_k) + M_N(\alpha_k)F_\gamma(\alpha_k)L_2(\alpha_k) = 0$$

$$k = 1, \dots, \ell$$

$$L_2(-\beta_k) + M_N(\beta_k)F_\gamma(\beta_k)L_1(-\beta_k) = 0$$

$$k = 1, \dots, n_1$$

$$L_2(-\alpha_k) + M_N(\alpha_k)F_\gamma(\alpha_k)L_1(-\alpha_k) = 0$$

$$k = 1, \dots, \ell.$$

The above system of equations can be rewritten in the matrix form

$$\mathcal{R}_\gamma \Phi = 0 \quad (5)$$

where the $2(n_1 + \ell) \times 1$ vector Φ contains the coefficients of L_1 and L_2 and $\mathcal{R}_\gamma$ is a $2(n_1 + \ell) \times 2(n_1 + \ell)$ matrix which can be computed numerically when γ is fixed. The optimal performance level γ_{opt} is the largest γ which makes $\mathcal{R}_\gamma$ singular. The corresponding nonzero Φ gives $L(s)$, and hence, all the components of C_{opt} are computed.

Example 1 Consider the weighted sensitivity minimization for the plant (1) with the following first-order weights:

$$W_1(s) = \frac{1}{s}, \quad W_2(s) = k s \quad (6)$$

where $k > 0$ represents the relative importance of the multiplicative uncertainty with respect to the tracking performance under steplike reference inputs (see Doyle et al. 1992; Özbay 2000). With (6) the functions $E_\gamma(s)$ and $F_\gamma(s)$ are computed as

$$E_\gamma(s) = \frac{1 + \gamma^2 s^2}{-\gamma^2 s^2}, \quad F_\gamma(s) = \frac{-\gamma s}{ks^2 + k_\gamma s + 1},$$

$$\text{where } k_\gamma = \sqrt{2k - \frac{k^2}{\gamma^2}}. \quad (7)$$

In this example $\ell = 1$ and $n_1 = 1$, with $\alpha_1 = 0.5$, $\beta_1 = j/\gamma$. For $k = 0.1$, the largest γ which makes $\mathcal{R}_\gamma$ singular is $\gamma_{\text{opt}} = 7.452$, and the coefficients of the corresponding $L(s)$ are computed from the SVD of $\mathcal{R}_{\gamma_{\text{opt}}}$,

$$L(s) = \frac{-0.0867 - 0.99623 s}{-0.0867 + 0.99623 s} = \frac{0.087 + s}{0.087 - s}.$$

Note that zeros of $E_\gamma(s)M_D(s)$ in $\overline{\mathbb{C}}_+$ appear as roots of the equation

$$1 + M_N(s)F_\gamma(s)L(s) = 0.$$

Hence, there are *internal* unstable pole-zero cancelations in the representation (4). An internally stable implementation of this controller is shown in Gumussoy (2011) using a realization similar to (3).

The above approach can also be extended to a class of infinite-dimensional plants with infinitely many poles in $\mathbb{C}_+$; see Gumussoy and Özbay (2004) for technical details.

A recent book Özbay et al. (2018) includes many examples from distributed parameters and discusses internally stable realization of the optimal controller and its approximations.

Summary and Future Directions

This entry briefly summarized robust control problems involving linear time-invariant infinite-dimensional plants with dynamic uncertainty models. Salient features of these robust control problems are captured by the mixed sensitivity minimization problem, for which a numerical computational procedure is outlined under the assumption that the weights are rational functions. Note that different types of plant models involving probabilistic, parametric, or structured (MIMO case) uncertainty are left out in this entry. Other robust control problems that are not discussed here include simultaneous stabilization (control of finitely many plant models by a single robust controller) and strong stabilization (robust control with the added restriction that the controller must be stable) of infinite-dimensional systems. Stable robust controller design techniques for different types of systems with time delays are illustrated in Özbay (2010) and Wakaiki et al. (2013); see also their references.

For practical implementation of infinite-dimensional robust controllers, it is important to find low-order approximations of stable irrational transfer functions with prescribed $\mathcal{H}_\infty$ error bound. There exist many different approximation techniques for various types of transfer functions, but there is still need for computationally efficient algorithms in this area. Another interesting topic along the same lines is direct computation of fixed-order $\mathcal{H}_\infty$ controllers for infinite-dimensional plants. In fact, computation of $\mathcal{H}_\infty$ -optimal PID controllers is still a challenging problem for infinite-dimensional plants, except

for some time-delay systems satisfying certain simplifying structural assumptions. Advances in numerical optimization tools will play critical roles in the computation of low (or fixed)-order robust controller design for infinite-dimensional plants; see, e.g., Gumussoy and Michiels (2011) for recent results along this direction.

In the past, robust control of infinite-dimensional systems found applications in many different areas such as chemical processes, flexible structures, robotic systems, transportation systems, and aerospace. Robust control problems involving systems with time-varying and uncertain time delays appear in control of networks and control over networks. Ongoing research in the networked systems area include generalization of these problems to more complex and interconnected systems.

There are also many interesting robust control problems in biological systems, where typical underlying plant models are nonlinear and infinite dimensional. Some of these problems are solved under simplifying assumptions; it is expected that robust control theory will make significant contributions to this field by extensions of the existing results to more realistic plant and uncertainty models.

Cross-References

- [Control of Linear Systems with Delays](#)
- [Flexible Robots](#)
- [H-Infinity Control](#)
- [Model Order Reduction: Techniques and Tools](#)
- [Networked Systems](#)
- [Optimal Control via Factorization and Model Matching](#)
- [Optimization-Based Robust Control](#)
- [PID Control](#)
- [Robust Control in Gap Metric](#)
- [Spectral Factorization](#)
- [Stability and Performance of Complex Systems Affected by Parametric Uncertainty](#)
- [Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems](#)

Bibliography

- Curtain R, Morris K (2009) Transfer functions of distributed parameter systems: a tutorial. *Automatica* 45:1101–1116
- Curtain R, Zwart HJ (1995) An introduction to infinite-dimensional linear systems theory. Springer, New York
- Desoer CA, Vidyasagar M (2009) Feedback systems: input-output properties. SIAM, Philadelphia
- Doyle JC, Francis BA, Tannenbaum AR (1992) Feedback control theory. Macmillan, New York
- Foias C, Özbay H, Tannenbaum A (1996) Robust control of infinite-dimensional systems: frequency domain methods. Lecture notes in control and information sciences, vol 209. Springer, London
- Gu G, Khargonekar PP, Lee EB (1989) Approximation of infinite-dimensional systems. *IEEE Trans Autom Control* 34:610–618
- Gumussoy S (2011) Coprime-inner/outer factorization of SISO time-delay systems and FIR structure of their optimal $\mathcal{H}_\infty$ controllers. *Int J Robust Nonlinear Control* 22:981–998
- Gumussoy S, Michiels W (2011) Fixed-order H -infinity control for interconnected systems using delay differential algebraic equations. *SIAM J Control Optim* 49(5):2212–2238
- Gumussoy S, Özbay H (2004) On the mixed sensitivity minimization for systems with infinitely many unstable modes. *Syst Control Lett* 53:211–216
- Meinsma G, Mirkin L, Zhong Q-C (2002) Control of systems with I/O delay via reduction to a one-block problem. *IEEE Trans Autom Control* 47:1890–1895
- Monje CA, Chen YQ, Vinagre BM, Xue D, Feliu V (2010) Fractional-order systems and controls, fundamentals and applications. Springer, London
- Morris KA (2001) $\mathcal{H}_\infty$ -output feedback of infinite-dimensional systems via approximation. *Syst Control Lett* 44:211–217
- Özbay H (2000) Introduction to feedback control theory. CRC Press LLC, Boca Raton
- Özbay H (2010) Stable $\mathcal{H}_\infty$ controller design for systems with time delays. In: Willems JC et al (eds) Perspectives in mathematical system theory, control, and signal processing. Lecture notes in control and information sciences, vol 398. Springer, Berlin/Heidelberg, pp 105–113
- Özbay H, Gumussoy S, Kashima K, Yamamoto Y (2018) Frequency domain techniques for H_∞ control of distributed parameter systems. SIAM, Philadelphia
- Sipahi R, Niculescu S-I, Abdallah CT, Michiels W, Gu K (2011) Stability and stabilization of systems with time delay. *IEEE Control Syst Mag* 31(1):38–65
- Stein G (2003) Respect the unstable. *IEEE Control Syst Mag* 23(4):12–25
- van Keulen B (1993) $\mathcal{H}_\infty$ -control for distributed parameter systems: a state space approach. Birkhäuser, Boston
- Wakaiki M, Yamamoto Y, Özbay H (2013) Stable controllers for robust stabilization of systems with infinitely many unstable poles. *Syst Control Lett* 62:511–516
- Zhong QC (2006) Robust control of time-delay systems. Springer, London
- Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice-Hall, Upper Saddle River

Robust Fault Diagnosis and Control

Steven X. Ding
University of Duisburg-Essen, Duisburg,
Germany

Abstract

Aiming at increasing system reliability and availability, integration of fault diagnosis into feedback control systems, and integrated design of control and diagnosis receive considerable attention in research and industrial applications. In the framework of robust control, integration of diagnosis and control systems can improve system robustness and fault tolerance. The core of such integrated systems is an observer that delivers needed information for a robust fault detection and feedback control.

Keywords

Observer-based fault diagnosis and control · Residual generation

Introduction

Advanced automatic control systems are marked by the high integration degree of digital electronics, intelligent sensors, and actuators. In parallel to this development, a new trend of integrating model-based fault detection and isolation (FDI) into the control systems can be observed (Patton et al. 2000; Blanke et al. 2006; Isermann 2006; Ding 2013), which is strongly driven by the increasing demands for system reliability and availability.

A critical issue surrounding the integration of a diagnostic module into a feedback control

loop is the interaction between the control and diagnosis. Initiated by Nett et al. (1988), study on the integrated design of control and diagnosis has received much attention, both in the research and application domains. The original idea of the integrated design scheme proposed in Nett et al. (1988) is to manage the interactions between the control and diagnosis in an integrated manner (Jacobson and Nett 1991; Ding 2009).

Robustness is an essential performance for model-based control and diagnostic systems. In the control and diagnosis framework, robustness is often addressed in different context (Ding 2013) and thus calls for special attention in the integrated design of control and diagnostic systems. In their study on fault-tolerant controller architecture, Zhou and Ren (2001) have proposed to deal with the integrated design in the framework of Youla parameterization of stabilization controllers (Zhou et al. 1996), which also builds the basis for achieving high robustness in an integrated control and diagnosis system. Below, we present the basic ideas and some representative schemes and methods for the integrated design of robust diagnosis and control systems.

Plant Model and Factorization Technique

Consider processes modeled by a linear time-invariant (LTI) system given in the state space representation

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + E_d d(t) + E_f f(t) \\ y(t) &= Cx(t) + Du(t) + F_d d(t) + F_f f(t) \\ z(t) &= C_z x(t) + D_z u(t)\end{aligned}$$

where $x \in \mathcal{R}^n$, $y \in \mathcal{R}^m$, and $u \in \mathcal{R}^{k_u}$ stand for the system state, output, and input vectors, respectively. $z \in \mathcal{R}^{k_z}$ is the controlled output vector. $d \in \mathcal{R}^{k_d}$, $f \in \mathcal{R}^{k_f}$ denote disturbance and fault vectors, respectively. $A, B, C, D, C_z, D_z, E_d, E_f, F_d$, and F_f are known matrices of appropriate dimensions.

A transfer matrix $G(s) = D + C(sI - A)^{-1}B$ with the minimal state space realization (A, B, C, D) can be factorized into

$$\begin{aligned}G(s) &= \hat{M}^{-1}(s)\hat{N}(s) \\ \hat{M}(s) &= I - C(sI - A_L)^{-1}L \\ \hat{N}(s) &= D + C(sI - A_L)^{-1}B_L \\ A_L &= A - LC, B_L = B - LD\end{aligned}$$

where L is selected so that A_L is Hurwitz and can be interpreted as an observer gain matrix. This factorization is called left coprime (Zhou et al. 1996).

Parametrization of Stabilizing Controllers

Let

$$u(s) = K(s)y(s)$$

be an LTI feedback controller. By means of the well-known Youla parameterization (Zhou et al. 1996), all stabilizing controllers can be described and parametrized by

$$\begin{aligned}K(s) &= \hat{V}^{-1}(s)\hat{U}(s) \\ \hat{V}(s) &= \hat{X}(s) - Q_c(s)\hat{N}(s) \\ \hat{U}(s) &= \hat{Y}(s) - Q_c(s)\hat{M}(s) \\ \hat{X}(s) &= I - F(sI - A_L)^{-1}B_L \\ \hat{Y}(s) &= F(sI - A_L)^{-1}L\end{aligned}$$

where $Q_c(s)$ is a stable parameter matrix and F is selected so that $A_F = A + BF$ is Hurwitz and can be interpreted as a state feedback gain matrix.

Parametrizations of Residual Generators

Given the system under consideration, an LTI residual generator is a dynamic system with $u(t)$, $y(t)$ as its inputs and $r(t)$ as output which satisfies, for $d(t) = 0$, $f(t) = 0$,

$$\forall x(0), u(t), \lim_{t \rightarrow \infty} r(t) = 0.$$

Residual generation is the first step for a successful fault diagnosis. The generated residual vector is an indicator for the occurrence of a fault. It is well-known that all LTI residual generators can be parameterized by

$$r(s) = R(s) \left(\hat{M}(s)y(s) - \hat{N}(s)u(s) \right)$$

where $R(s)$ is a stable parameter matrix and called post-filter (Ding 2013).

Integration of Controller and Residual Generator into a Control Loop

It is remarkable that both the feedback controllers and residual generators can be parametrized based on the left coprime factorization of the plant model. This is the basis for an integration of diagnosis and control into a feedback control system. In Ding (2014), it is demonstrated that the abovementioned Youla parametrization form is in fact an observer-based feedback controller, which can be expressed by

$$u(s) = F\hat{x}(s) + Q_c(s) \left(\hat{N}(s)u(s) - \hat{M}(s)y(s) \right)$$

where $\hat{x}(s)$ is a state estimate delivered by a full-order state observer (Zhou et al. 1996; Anderson 1998). Moreover, the residual generator can also be written as

$$r(s) = R(s)r_o(s), r_o(s) = y(s) - \hat{y}(s)$$

with $\hat{y}(s)$ being the output estimate delivered by an observer (Ding 2013). As a result, a stabilization feedback controller and residual generator can be integrated into a dynamic system of the following form

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + Lr_o(t)$$

$$= A_L\hat{x}(t) + B_Lu(t) + Ly(t)$$

$$r_o(t) = y(t) - \hat{y}(t), \hat{y}(t) = C\hat{x}(t) + Du(t)$$

$$\begin{bmatrix} u(s) \\ r(s) \end{bmatrix} = \begin{bmatrix} F\hat{x}(s) \\ 0 \end{bmatrix} + \begin{bmatrix} -Q_c(s) \\ R(s) \end{bmatrix} r_o(s).$$

The core of the above control and diagnostic system is a state observer that delivers a state estimation $\hat{x}(t)$ and the primary residual vector $r_o(t)$. The design parameters of this integrated control and diagnosis system are L , F , the observer and state feedback control gain matrices, as well as $Q_c(s)$, $R(s)$.

Robustness of Diagnostic and Control Systems

While in the robust control framework the controller design is typically formulated as minimizing a system norm of the transfer function matrix from the disturbance vector d to the control output z (Zhou et al. 1996), the design objective of a robust fault detection system consists in an optimal trade-off between the robustness against d and the sensitivity to the fault vector f . Considering that

$$\begin{aligned} r(s) &= R(s) (y(s) - \hat{y}(s)) \\ &= R(s) \left(\hat{N}_d(s)d(s) + \hat{N}_f(s)f(s) \right) \end{aligned}$$

$$\hat{N}_d(s) = F_d + C(sI - A_L)^{-1}(E_d - LF_d)$$

$$\hat{N}_f(s) = F_f + C(sI - A_L)^{-1}(E_f - LF_f)$$

(Ding 2013), the design objective can be formulated as

$$\sup_{R(s)} \frac{\|R(s)\hat{N}_f(s)\|_{index}}{\|R(s)\hat{N}_d(s)\|}$$

or in a suboptimum form as finding $R(s)$ so that for some given $\alpha > 0$, $\beta > 0$

$$\|R(s)\hat{N}_d(s)\| \leq \alpha, \|R(s)\hat{N}_f(s)\|_{index} > \beta.$$

Similar to the robust controller design, a (system) norm like $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norm, denoted by $\|\cdot\|$, is applied for the evaluation of the influence of the disturbances. Differently, the evaluation of the sensitivity to the fault vector, expressed by $R(s)\hat{N}_f(s)$, can be realized either using a system norm or the so-called $\mathcal{H}_-$ index, denoted by $\|\cdot\|_-$, which indicates the minimum influence of f on r (Ding 2013).

In order to detect the fault occurrence reliably and successfully, a decision-making procedure is needed. It consists of a further evaluation of the residual signal and a detection logic. Typically, a signal norm of r , e.g., $\mathcal{L}_2$ norm, and a simple detection logic like

$$\begin{cases} \|r\| > J_{th} \implies \text{Alarm for fault} \\ \|r\| \leq J_{th} \implies \text{Fault-free} \end{cases}$$

are adopted for this purpose, where J_{th} is a further design parameter and called threshold (Ding 2013). The threshold setting depends on the dynamics of r , its norm-based evaluation, and has significant influence on the fault detection performance. For the purpose of reducing false alarms, the threshold is often set as

$$\begin{aligned} J_{th} &= \sup_{f=0, \|d\| \leq d_d} \|r\| \\ &= \sup_{f=0, \|d\| \leq d_d} \|R(s)\hat{N}_d(s)d(s)\|. \end{aligned}$$

That is, the threshold is set to be the maximum value of the influence of the disturbances on the residual signal in the fault-free case. Thus, different designs of the residual generator will result in different threshold settings. In this context, an optimal design of a fault diagnosis system is understood as an integrated design of the residual generator, the evaluation function, and the threshold (Ding 2013).

An Integrated Design Scheme for Robust Diagnosis and Control

Assume that the system under consideration satisfies the following conditions:

- $\|d\|_2 \leq \delta_d$
- (A, B) is stabilizable and (C, A) is detectable
- $D = 0$
- $D_z^T D_z > 0$ and $F_d F_d^T > 0$
- $\begin{bmatrix} A - j\omega I & B \\ C_z & D \end{bmatrix}$ has full column rank for all ω
- $\begin{bmatrix} A - j\omega I & E_d \\ C & F_d \end{bmatrix}$ has full row rank for all ω .

Then, the following observer and state feedback gain matrices

$$\begin{aligned} L^* &= (E_d F_d^T + Y C^T) (F_d F_d^T)^{-1} \\ F^* &= - (D_z^T D_z)^{-1} (B^T X + D_z^T C_z) \end{aligned}$$

as well as

$$Q_c^*(s) = 0, R^*(s) = (F_d F_d^T)^{-1/2}$$

result in an optimal integrated design of the robust diagnostic and control system with:

- $\mathcal{H}_2$ optimal control performance and
- maximal fault detectability and the optimal threshold setting

$$J_{th} = \sup_{f=0, \|d\| \leq \delta_d} \|r\|_2 = \delta_d$$

where $Y \geq 0, X \geq 0$ are, respectively, the solution of the following two Riccati equations

$$\begin{aligned} AY + Y A^T + E_d E_d^T - (E_d F_d^T + Y C^T) \cdot \\ (F_d F_d^T)^{-1} (E_d F_d^T + Y C^T)^T &= 0 \\ A^T X + X A + C_z^T C_z - (C_z^T D_z + X B) \cdot \\ (D_z^T D_z)^{-1} (C_z^T D_z + X B)^T &= 0. \end{aligned}$$

That L^*, F^* lead to minimizing the $\mathcal{H}_2$ norm of the transfer matrix from d to z is a well-known result (Zhou et al. 1996). The optimal fault detection performance can be understood from two different viewpoints:

- optimum in the sense of

$$\forall \omega, \sup_{R(s)} \frac{\sigma_i \left(R(j\omega) \hat{N}_f(j\omega) \right)}{\left\| R(s) \hat{N}_d(s) \right\|_{\infty}} \\ = \sigma_i \left(\left(F_d F_d^T \right)^{-1/2} \hat{N}_f^*(j\omega) \right)$$

where $\sigma_i \left(R(j\omega) \hat{N}_f(j\omega) \right)$ is the i -th singular value of matrix $R(j\omega) \hat{N}_f(j\omega)$, $i = 1, \dots, k_f$, $\hat{N}_f^*(s) = \hat{N}_f(s) |_{L=L^*}$ (Ding 2013)

- a fault that can be detected by any LTI detection system will also be detected using the detection system with the above parameter and threshold setting. Thus, this detection system provides the maximal fault detectability (Ding 2013).

It is worth to remark that:

- the assumptions mentioned above are standard in the $\mathcal{H}_2$ optimal control (Zhou et al. 1996)
- the optimization problem

$$\forall \omega, \sup_{R(s)} \frac{\sigma_i \left(R(j\omega) \hat{N}_f(j\omega) \right)}{\left\| R(s) \hat{N}_d(s) \right\|_{\infty}}$$

is a more general form of the so-called $\mathcal{H}_-/\mathcal{H}_{\infty}$ or $\mathcal{H}_{\infty}/\mathcal{H}_{\infty}$ optimization of observer-based fault detection systems, and thus its solution is called unified solution (Ding 2013)

- The solution given above is a state space realization of the robust fault detection problems, which is, e.g., described in Ding (2013)
- this integrated design scheme can also be applied to discrete-time and stochastic systems (Ding 2013).

Summary and Future Directions

Increasing reliability and availability of advanced automatic control systems is of considerable

practical interests. Integration of fault diagnosis into feedback control systems and integrated design of robust control and diagnosis are useful solutions for real time applications. They can also be integrated into a fault-tolerant control system (Blanke et al. 2006; Zhou and Ren 2001). A further potential application field is fault diagnosis in feedback control loops using embedded residual signals.

From the viewpoint of research, integrated design of robust control and diagnosis in nonlinear and time-varying dynamic systems with parameter faults are challenging issues. The $\mathcal{L}_2$ -gain technique for nonlinear control (Van der Schaft 2000) and diagnosis (Yang et al. 2016) and the fault detection schemes proposed in Li and Zhou (2009) and Zhong et al. (2016) are promising results for the future investigations in this area.

Cross-References

- [Fault Detection and Diagnosis](#)
- [Fault-Tolerant Control](#)
- [Robust \$\mathcal{H}_2\$ Performance in Feedback Control](#)

Bibliography

- Anderson BDO (1998) From Youla-Kucera to identification, adaptive and nonlinear control. *Automatica* 34:1485–1506
- Blanke M, Kinnaert M, Lunze J, Staroswiecki M (2006) *Diagnosis and fault-tolerant control*, 2nd edn. Springer, Berlin/Heidelberg
- Ding SX (2009) Integrated design of feedback controllers and fault detectors. *Annu Rev Control* 33:124–135
- Ding SX (2013) *Model-based fault diagnosis techniques – design schemes, algorithms and tools*, 2nd edn. Springer, London
- Ding SX (2014) *Data-driven design of fault diagnosis and fault-tolerant control systems*. Springer, London
- Isermann R (2006) *Fault diagnosis systems*. Springer, Berlin/Heidelberg
- Jacobson CA, Nett CN (1991) An integrated approach to controls and diagnostics using the four parameter controller. *IEEE Control Syst* 11:22–28
- Li X, Zhou K (2009) A time domain approach to robust fault detection of linear time-varying systems. *Automatica* 45:94–102

- Nett CN, Jacobson CA, Miller AT (1988) An integrated approach to controls and diagnostics. In: Proceedings of ACC, pp 824–835
- Patton RJ, Frank PM, Clark RN (eds) (2000) Issues of fault diagnosis for dynamic systems. Springer, London
- Van der Schaft A (2000) L2 – gain and passivity techniques in nonlinear control. Springer, London
- Yang Y, Ding SX, Li L (2016) Parametrization of nonlinear observer-based fault detection systems. *IEEE Trans Autom Control* 61:3687–3692
- Zhong M, Ding SX, Zhou D (2016) A new scheme of fault detection for linear discrete time-varying systems. *IEEE Trans Autom Control* 61:2597–2602
- Zhou K, Ren Z (2001) A new controller architecture for high performance, robust, and fault-tolerant control. *IEEE Trans Autom Control* 46:1613–1618
- Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice-Hall. Upper Saddle River

Robust H_2 Performance in Feedback Control

Fernando Paganini
Universidad ORT Uruguay, Montevideo,
Uruguay

Abstract

This entry discusses an important compromise in feedback design: reconciling the superior performance characteristics of the $\mathcal{H}_2$ optimization criterion, with robustness requirements expressed through induced norms such as $\mathcal{H}_\infty$. The fact that both criteria have frequency-domain characterizations and involve similar state-space machinery motivated many researchers to seek an adequate combination. We review here robust $\mathcal{H}_2$ analysis methods based on convex optimization developed in the 1990s and comment on their implications for controller synthesis.

Keywords

Linear matrix inequalities · Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control · Robustness analysis · Structured uncertainty

Introduction

Can mathematics help us deal with the inevitable theory-practice gap? Should we be optimistic and assume that discrepancies between models and nature are random and neutral towards our actions or be pessimistic and design for the worst such discrepancies? Feedback control theory has struggled with these questions, perhaps more so than other fields.

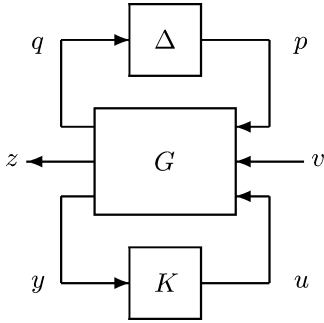
During the surge of optimal control in the 1960s, optimism carried the day. A prominent example is the LQG ($\mathcal{H}_2$) regulator, which minimizes the effect of random disturbances and has an elegant state-space solution; in comparison, the frequency-domain designs of classical control appeared primitive and conservative. But the pessimists struck back in the late 1970s, showing things could go very wrong (unstable) with LQG, when a parameter variation was introduced in the plant model. This ushered in the robust control era of the 1980s, with its worst-case analysis of stability over deterministic sets of plants, leading to other design metrics such as $\mathcal{H}_\infty$ control. In this mentality, exogenous disturbances were also treated as an adversary to be protected against in the worst case, perhaps an excess of pessimism.

The robust $\mathcal{H}_2$ problem incarnates the search for a middle ground, where stability is treated with the conservatism it deserves, but performance is optimized for a more neutral noise. This entry summarizes efforts made around the 1990s to seek this compromise.

$\mathcal{H}_2$ Optimal Control

In the feedback diagram of Fig. 1, signals are vector valued, and we focus on continuous time. G is a linear system with a given state-space representation. Initially omit the upper loop (set $\Delta = 0$). The LQG regulator is the controller K that internally stabilizes the feedback loop and minimizes the variance of the error variable z , assuming the input v is white Gaussian noise.

For an alternative description, denote by $\hat{T}_{zv}(s)$ the closed-loop transfer function from v to z ; we wish to design K such that $\hat{T}_{zv}(s)$ is



Robust H_2 Performance in Feedback Control, Fig. 1 Feedback control and model uncertainty

analytic in $\text{Re}(s) \geq 0$ and has minimum $\mathcal{H}_2$ norm, defined by

$$\|\hat{T}_{zv}\|_{\mathcal{H}_2} = \left(\int_{-\infty}^{\infty} \text{Tr}(\hat{T}_{zv}(j\omega)^* \hat{T}_{zv}(j\omega)) \frac{d\omega}{2\pi} \right)^{\frac{1}{2}}; \quad (1)$$

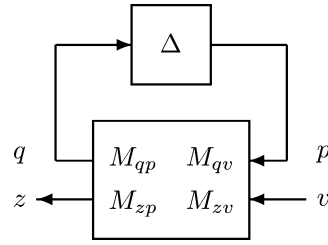
here Tr denotes matrix trace and $*$ denotes conjugate transpose. The equivalence between this $\mathcal{H}_2$ -optimal control and LQG follows from classical filtering, modeling v as uncorrelated components of unit power spectral density over all frequency. By adding a filter in the input of G , noise of known, colored spectrum can be accommodated as well.

A different motivation, in the case of scalar v , is to observe that $\|\hat{T}_{zv}\|_{\mathcal{H}_2}^2$ is the energy ($\mathcal{L}_2$ -norm square) of the system impulse response. Thus it measures the transient error in response to known inputs or initial conditions which may be generated by an impulse.

The $\mathcal{H}_2$ (LQG) optimal feedback has an elegant solution, computable in state-space through two algebraic Riccati equations (AREs). Its quick popularity was, however, hampered due to its lack of *stability margins*: a small error in model parameters can make the closed-loop unstable (Doyle 1978). This motivated methods to explicitly address such modeling errors.

Model Uncertainty and Robustness

Suppose some parameter in the model of G is uncertain, $\alpha = \alpha_0 + \kappa\delta$, $\delta \in [-1, 1]$; often,



Robust H_2 Performance in Feedback Control, Fig. 2 Robustness analysis setup

the normalized variation δ can be “pulled out” into the *uncertainty block* Δ of Fig. 1. The same technique can account for unmodeled linear time-invariant (LTI) dynamics, e.g., high frequency effects: they can be “covered” by a normalized transfer function $\hat{\Delta}(j\omega)$ and frequency weights that connect it to G . Even further, a nonlinear or time-varying (NL,TV) modeling error can be represented through an *operator* Δ in signal space. The references contain details on this modeling technique.

To analyze the effect of such errors, suppose K has been chosen to stabilize G and M is the resulting closed-loop system, with state-space representation

$$\begin{aligned} \dot{x} &= Ax + B_p p + B_v v, \\ q &= C_q x, \\ z &= C_z x. \end{aligned} \quad (2)$$

A is an $n \times n$ stable (Hurwitz) matrix, and for simplicity there are no feed-through terms. Figure 2, represents the interconnection of M with the uncertainty.

To quantify the size of uncertainty, it is convenient to use an *induced norm* (gain) in signal space and constrain Δ to the normalized ball $\{\|\Delta\| \leq 1\}$. If the subsystem M_{qp} in feedback with Δ satisfies itself the induced norm constraint $\|M_{qp}\| < 1$, the small gain theorem implies *robust stability* over the entire ball. Focusing for the rest of this article on the $\mathcal{L}_2$ signal space (square-integrable functions), the latter induced norm is equivalent to the $\mathcal{H}_\infty$ norm of the transfer function:

$$\|\hat{M}_{qp}(s)\|_{\mathcal{H}_\infty} := \operatorname{ess\,sup}_{\omega \in \mathbb{R}} \bar{\sigma}(\hat{M}_{qp}(j\omega)),$$

where $\bar{\sigma}(\cdot)$ denotes matrix maximum singular value.

This motivates $\mathcal{H}_\infty$ -optimal control: design K to minimize the above quantity with internal stability. This problem also admits state-space solutions based on AREs and thus is a valid competing paradigm to $\mathcal{H}_2$.

To accommodate multiple sources of uncertainty within Fig. 1, we can use a *block diagonal* structure:

$$\Delta = \operatorname{diag}[\Delta_1, \dots, \Delta_d]. \quad (3)$$

Here, different uncertainty blocks (parametric, LTI, LTV, or NL) enter in separate “channels”; $\mathbf{B}_\Delta$ denotes the unit ball of operators with the prescribed structure. For stability studies, *causality* of the operator is required.

Robust stability under structured uncertainty is a rich topic: we refer to the article on the *structured singular value* (μ) in this encyclopedia. We invoke here robustness conditions based on the set $\mathbf{A}$ of positive definite matrix *scalings* or *multipliers* of the form:

$$\Lambda = \operatorname{diag}[\lambda_1 I, \dots, \lambda_d I], \quad (4)$$

with submatrices of the same dimensions as the blocks in (3), thus commuting with a matrix Δ of that structure.

Consider the frequency family of matrix inequalities

$$\begin{aligned} \hat{M}_{qp}^*(j\omega)\Lambda(\omega)\hat{M}_{qp}(j\omega) - \Lambda(\omega) &< 0 \quad \forall \omega; \\ \Lambda(\omega) &\in \mathbf{A}. \end{aligned} \quad (5)$$

At each ω , this is a linear matrix inequality (LMI); testing its feasibility is a convex, tractable problem. A solution implies the *scaled-small gain condition*

$$\bar{\sigma}\left(\Lambda(\omega)^{\frac{1}{2}}\hat{M}_{qp}(j\omega)\Lambda^{-\frac{1}{2}}(\omega)\right) < 1;$$

this “ μ upper bound” implies robust stabil-

ity when uncertainty is LTI, through commuting $\Lambda(\omega)$ with $\hat{\Delta}(j\omega)$.

If uncertainty is NLTV, (5) must be strengthened to enforce $\Lambda(\omega) \equiv \Lambda$, constant in frequency. This condition turns out to be both necessary and sufficient for robust stability. Here the LMIs would be coupled in frequency; however, the Kalman-Yakubovich-Popov lemma reduces them to an equivalent LMI in terms of the state-space matrices in (2), with variables $\Lambda \in \mathbf{A}$ and an $n \times n$ matrix $P > 0$:

$$\begin{bmatrix} A^*P + PA + C_q^*\Lambda C_q & PB_p \\ B_p^*P & -\Lambda \end{bmatrix} < 0. \quad (6)$$

What about performance? The mapping $T_{zv}(\Delta)$ between the disturbance v and the error z now depends on the uncertainty. The default procedure in robust control has been to measure performance with the same induced norm, evaluating $\|T_{zv}(\Delta)\|_{\mathcal{L}_2 \rightarrow \mathcal{L}_2}$ in the worst-case over $\Delta \in \mathbf{B}_\Delta$. This can be computed with similar complexity to establishing robust stability. It amounts, however, to treating noise with the same worst-case mentality as stability, a questionable choice. For instance, in LTI systems the worst-case signals are sinusoids at the worst frequency and spatial direction; while one should protect against such signals arising in the Δ -loop due to instability, it is not natural to expect them as external disturbances, which are usually of broad spectrum.

Robust $\mathcal{H}_2$ Performance Analysis

In the absence of uncertainty, the $\mathcal{H}_2$ norm of the nominal mapping $T_{zv}(0) = M_{zv}$ provides a natural performance criterion, measuring the response to flat-spectrum disturbances or the transient response. When uncertainty is present, it motivates a worst-case analysis of stability; a natural combination is to impose *robust $\mathcal{H}_2$ performance*: evaluating the worst-case $\mathcal{H}_2$ norm of $T_{zv}(\Delta)$ over the uncertainty class $\mathbf{B}_\Delta$. We will highlight some methods based on semidefinite programming to perform such evaluations; for further details and comparisons, we refer to Paganini and Feron (1999).

A Frequency Domain Robust Performance Criterion

Consider the following optimization:

$$J_f := \inf \int_{-\infty}^{\infty} \text{Tr}(Y(\omega)) \frac{d\omega}{2\pi}, \text{ subject to}$$

$$\hat{M}(j\omega)^* \begin{bmatrix} \Lambda(\omega) & 0 \\ 0 & I \end{bmatrix} \hat{M}(j\omega) - \begin{bmatrix} \Lambda(\omega) & 0 \\ 0 & Y(\omega) \end{bmatrix} < 0 \quad (7)$$

for each ω , and $\Lambda(\omega) \in \mathbf{\Lambda}$.

Here $\hat{M}$ is the transfer function in Fig. 2; a submatrix of the above includes (5), implying robust stability under structured LTI uncertainty. Furthermore, we have the robust $\mathcal{H}_2$ performance bound (Paganini 1999):

$$\sup_{\Delta \in \mathbf{B}_{\mathbf{\Lambda}}^{LTI}} \|T_{zv}(\Delta)\|_{\mathcal{H}_2}^2 \leq J_f. \quad (8)$$

We sketch the argument based on the Fourier transforms $\hat{p}(j\omega)$, etc., for signals in Fig. 2. Applying the quadratic form in (7) to the joint vector of $\hat{p}$ and $\hat{v}$ gives

$$\sum_{i=1}^d \lambda_i(\omega) |\hat{q}_i|^2 + |\hat{z}|^2 \leq \sum_{i=1}^d \lambda_i(\omega) |\hat{p}_i|^2 + \hat{v}^* Y(\omega) \hat{v}.$$

The subvectors $\hat{p}_i$, $\hat{q}_i$ correspond to uncertainty blocks, $\hat{p}_i = \hat{\Delta}_i(j\omega) \hat{q}_i$; since $\bar{\sigma}(\hat{\Delta}_i(j\omega)) \leq 1$, $|\hat{q}_i| \geq |\hat{p}_i|$. Also $\lambda_i(\omega) > 0$, so these terms can be simplified, leading to

$$|\hat{T}_{zv}(j\omega) \hat{v}|^2 = |\hat{z}|^2 \leq \hat{v}^* Y(\omega) \hat{v}.$$

This means $\hat{T}_{zv}(j\omega)^* \hat{T}_{zv}(j\omega) \leq Y(\omega)$ for every Δ , and therefore the $\mathcal{H}_2$ norm bound

$$\|T_{zv}(\Delta)\|_{\mathcal{H}_2}^2 \leq \int_{-\infty}^{\infty} \text{Tr}(Y(\omega)) \frac{d\omega}{2\pi}$$

holds, from which (8) follows.

The computation involved in (7) at each frequency is a semidefinite program (SDP): minimizing the linear cost $\text{Tr}(Y(\omega))$ subject to an LMI constraint, a tractable problem. Adding a frequency sweep, we have a practical method to bound the desired robust performance.

The inequality (8) is in general strict. Beyond the usual conservatism of convex bounds for μ , when noise is of dimension m , a conservatism of up to this order may appear; an improvement to address this issue with augmented SDPs is given in Sznaier et al. (2002). Finally, causality of the uncertainty is not imposed in the frequency-domain criterion.

As in the study of robust stability, we wish to extend the analysis to NLTV uncertainty blocks. Now the mapping $T_{zv}(\Delta)$ can no longer be represented by a transfer function, so what is the “ $\mathcal{H}_2$ ” cost? We return to our motivation for this performance notion: to measure the effect of disturbances of flat spectrum.

In Paganini (1999), the flat-spectrum property is imposed as a deterministic constraint on the input disturbances. For the scalar v case, define $W_{\eta,B} \subset \mathcal{L}_2$ by the family of *integral quadratic constraints*:

$$\int_{-\beta}^{\beta} |v(j\omega)|^2 \frac{d\omega}{2\pi} \begin{cases} \leq \frac{\beta}{\pi} + \eta & \forall \beta; \\ \geq \frac{\beta}{\pi} - \eta, & \beta \in [0, B]. \end{cases} \quad (9)$$

This imposes that the cumulative spectrum is approximately linear (to a tolerance $\eta > 0$), up to bandwidth B , and has sublinear growth beyond that. Extensions to vector-valued signals are also given. For a stable LTI system T_{zv} , it is not difficult to verify that

$$\|T_{zv}\|_{\mathcal{H}_2}^2 = \lim_{\substack{\eta \rightarrow 0 \\ B \rightarrow \infty}} \sup_{v \in W_{\eta,B}} \|T_{zv} v\|_2^2,$$

but the right-hand side applies to NLTV systems as well. The following result can be established in the latter case:

$$\lim_{\substack{\eta \rightarrow 0 \\ B \rightarrow \infty}} \sup_{v \in W_{\eta,B}, \Delta \in \mathbf{B}_{\mathbf{\Lambda}}^{NLTV}} \|T_{zv}(\Delta) v\|_2^2 = J'_f,$$

where the right-hand side is the variant of (7) with the restriction that $\Lambda(\omega) \equiv \mathbf{\Lambda}$, constant in frequency. In this case the characterization is *exact*, with equality above. This follows from a duality argument in function space, where $Y(\omega)$ appears as the multiplier for the constraint in (9). While coupled in frequency, J'_f is again equivalent to a finite-dimensional SDP in state space.

Let us review, instead, a different state-space method, motivated by alternate definitions of the $\mathcal{H}_2$ cost.

A State-Space Criterion Invoking Causality

Consider the semidefinite program

$$J_s := \inf \operatorname{Tr}(B_v^* P B_v) \text{ subject to } P > 0, \Lambda \in \mathbf{A},$$

$$\begin{bmatrix} A^* P + P A + C_q^* \Lambda C_q + C_z^* C_z & P B_p \\ B_p^* P & -\Lambda \end{bmatrix} < 0. \quad (10)$$

The LMI above is very similar to (6); indeed it provides a robust stability certificate and in addition a bound on a generalized $\mathcal{H}_2$ cost, for arbitrary (NLTV) causal uncertainty blocks. Again, we sketch the argument.

For stability, consider the system of Fig. 2 with $v \equiv 0$, initial condition $x(0) = x_0$. Define the storage function $V(x) = x^* P x$; differentiating it under (2) and applying the LMI (10) to the joint vector of $x(t)$, $p(t)$ yield

$$\dot{V} + |z|^2 \leq -q^* \Lambda q + p^* \Lambda p = \sum_{i=1}^d \lambda_i (|p_i|^2 - |q_i|^2).$$

Integrating the above over $(0, t)$, the sum on the right becomes nonpositive because $\lambda_i > 0$ and the operator $\Delta_i : q_i \rightarrow p_i$ is causal and contractive. This leads to

$$V(x(t)) + \int_0^t |z(\tau)|^2 d\tau \leq V(x_0), \quad (11)$$

which implies Lyapunov stability; the bound can be sharpened to prove asymptotic stability. Also, letting $t \rightarrow \infty$ yields the energy bound $\|z\|_2^2 \leq V(x_0)$.

Suppose now that x_0 is generated by applying to the (causal) system at rest, an impulse $v(t) = \delta(t)$, assumed scalar. The result is $x(0+) = B_v$, so $V(x_0) = B_v^* P B_v$; the impulse response energy of $T_{zv}(\Delta)$ is thus bounded. Minimizing over P, Λ leads to the robust $\mathcal{H}_2$ performance bound

$$\sup_{\Delta \in \mathbf{B}_{\Delta}^{NLTV}} \|T_{zv}(\Delta) \delta(t)\|_2^2 \leq J_s,$$

where the $\mathcal{H}_2$ cost is generalized as the impulse response energy. An extension to multiple impulse channels is available. This kind of result was first obtained by Stoorvogel (1993) for unstructured uncertainty.

An alternate notion of $\mathcal{H}_2$ cost for NLTV systems, also considered in Stoorvogel (1993), is the average output variance when the input is random white noise. This is formalized by replacing (2) with a stochastic differential equation (e.g., Oksendal 1985) and extending the bound (11) using Ito calculus; for details see Paganini and Feron (1999). The following robust $\mathcal{H}_2$ performance bound is obtained:

$$\limsup_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^{\tau} E|z(t)|^2 dt \leq J_s \quad \forall \Delta \in \mathbf{B}_{\Delta}^{NLTV}.$$

What if the uncertainty is time invariant? Incorporating frequency-dependent scalings, with causality, into the state-space approach must be done approximately, generating $\hat{\Lambda}(j\omega)$ through the span of a predefined finite basis of causal, rational transfer functions. Searching over this basis for a bound on the impulse response energy can be pursued with state-space SDPs, now of a size increasing with the basis dimensionality. We refer to Feron (1997) for details.

Robust $\mathcal{H}_2$ Synthesis

Prior sections have focused on the robustness *analysis* of a closed-loop system M , obtained from G after designing a nominally stabilizing controller. Can we *synthesize* K with robust $\mathcal{H}_2$ performance as an objective? We overview some contributions to this question.

Multiojective $\mathcal{H}_2/\mathcal{H}_{\infty}$ Control

Let us discuss first the more modest objective of optimizing *nominal* $\mathcal{H}_2$ performance while guaranteeing robust stability. If the uncertainty block Δ in Fig. 2 is unstructured, the problem is equivalent to

$$\text{Minimize } \|\hat{M}_{zv}\|_{\mathcal{H}_2}, \text{ subject to } \|\hat{M}_{qp}\|_{\mathcal{H}_{\infty}} < 1.$$

Using a Youla parameterization of stabilizing controllers, $\hat{M}(s)$ depends affinely on a stable parameter $\hat{Q}(s)$; this makes the optimization over $\hat{Q}$ convex. However it has been shown to give infinite-dimensional solutions that must be approximated by suitable truncations; see Sznaier et al. (2000) and references therein.

To better exploit the state-space structure common to $\mathcal{H}_2$ and $\mathcal{H}_\infty$ synthesis, Bernstein and Haddad (1989) proposed a simplification: minimize an *auxiliary cost* that upper bounds the $\mathcal{H}_2$ norm while imposing the $\mathcal{H}_\infty$ constraint, through a common storage function. This cost is optimized by controllers of the order of the plant, characterized in terms of coupled AREs; later on Khargonekar and Rotea (1991) recast this problem using convex optimization. Also Zhou et al. (1994) and Doyle et al. (1994) studied the dual (transpose) structure.

The latter version is in fact directly related to the analysis condition (10), with a fixed $\Lambda = \lambda I$. A matrix P satisfying this condition imposes the $\mathcal{H}_\infty$ norm restriction and upper bounds the nominal $\mathcal{H}_2$ cost. This idea of imposing multiple objectives through a common storage function has more general applicability: Scherer et al. (1997) showed that all such problems admit tractable synthesis based on LMIs, with solutions of the same order as the plant.

Synthesis for Robust Performance

We have seen that rather than just an upper bound on nominal performance, (10) ensures the more stringent robust $\mathcal{H}_2$ performance requirement; therefore it becomes the basis of a robust $\mathcal{H}_2$ synthesis technique. In Stoorvogel (1993) this method is laid out for unstructured uncertainty: search linearly over the scalar λ and solve the auxiliary cost synthesis problem for each λ .

What about structured uncertainty? We run here into a general difficulty of such synthesis questions, even for robust stability alone. In that case, seeking simultaneously a controller K and a scaling Λ so that conditions (5) or (6) are satisfied by the resulting M is not a computationally friendly problem. In the absence of a general solution method, iterating between an $\mathcal{H}_\infty$ design of K for fixed Λ and the analysis

conditions to find Λ is commonly used for design.

Things can be no easier for robust $\mathcal{H}_2$ performance, but the iterative procedure does generalize to the conditions in (10): for fixed K , the SDP will return structured Λ 's, which can then be fixed for a multiobjective synthesis step based on the “auxiliary cost” in (10) as discussed above. If constant Λ are used (designing for NLTV uncertainty), all controllers obtained are of the order of the plant.

If uncertainty is LTI, an alternative is to carry out the analysis step in the frequency domain, finding a $\Lambda(\omega)$, $Y(\omega)$ through (7). In the corresponding situation for μ -synthesis, where only $\Lambda(\omega)$ is found, a step of fitting and spectral factorization is needed to approximate such scalings through a rational weights, which are then incorporated into $\mathcal{H}_\infty$ synthesis. A similar frequency weight in the performance channel can approximate the effect of $Y(\omega)$, thus relying on weighted $\mathcal{H}_\infty$ synthesis to pursue the $\mathcal{H}_2$ performance objective. Of course, the order of the resulting controllers is increased.

Summary and Future Directions

The tradeoff between performance and robustness is essential to feedback control. In the case of linear multivariable design, it motivated a compromise between $\mathcal{H}_2$ performance and $\mathcal{H}_\infty$ -type robustness, pursued with the state-space and frequency-domain tools common to these metrics. We have highlighted robust $\mathcal{H}_2$ analysis conditions obtained in the 1990s based on semidefinite programming, which provided the greatest flexibility to integrate the aforementioned tools and different points of view (worst-case, average case) present in this problem. As in other situations, the robust synthesis question has proven more difficult: design cannot be “automated” to the degree that was once envisioned.

The passage of time makes issues that once attracted strong attention look narrow in scope, so it is not natural to indicate directions that directly follow on this work. Perhaps the best

legacy that the robust $\mathcal{H}_2$ generation can take to other problems is the willingness to integrate various disciplines (dynamics, operator theory, stochastics, optimization) to face the demands of applied mathematical research.

Cross-References

- [H-Infinity Control](#)
- [KYP Lemma and Generalizations/Applications](#)
- [Linear Quadratic Optimal Control](#)
- [LMI Approach to Robust Control](#)
- [Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems](#)

Recommended Reading

LQG control is covered in many textbooks, e.g., Anderson and Moore (1990). A standard text for robust control with an $\mathcal{H}_\infty$ perspective, including structured singular values, the Youla parameterization, and the Riccati equation solution for $\mathcal{H}_\infty$ synthesis, is Zhou et al. (1996); see also Sánchez-Peña and Szaier (1998) with application examples. The textbook of Dullerud and Paganini (2000) incorporates the more recent developments based on LMIs; see Boyd and Vandenberghe (2004) for background on semidefinite programming.

Bibliography

- Anderson B, Moore JB (1990) Optimal control: linear quadratic methods. Prentice Hall, Englewood Cliffs
- Bernstein DS, Haddad WH (1989) LQG control with an $\mathcal{H}_\infty$ performance bound: a Riccati equation approach. *IEEE Trans Autom Control* 34(3):293–305
- Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge
- Doyle J (1978) Guaranteed margins for LQG regulators. *IEEE Trans Autom Control* 23(4):756–757
- Doyle J, Zhou K, Glover K, Bodenheimer B (1994) Mixed $\mathcal{H}_2$ and $\mathcal{H}_\infty$ performance objectives II: optimal control. *IEEE Trans Autom Control* 39(8):1575–1587
- Dullerud GE, Paganini F (2000) A course in robust control theory: a convex approach. Texts in applied mathematics, vol 36. Springer, New York
- Feron E (1997) Analysis of robust $\mathcal{H}_2$ performance using multiplier theory. *SIAM J Control Optim* 35(1):160–177
- Khargonekar P, Rotea M (1991) Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control: a convex optimization approach. *IEEE Trans Autom Control* 36(7):824–837
- Oksendal B (1985) Stochastic differential equations. Springer, New York
- Paganini F (1999) Convex methods for robust $\mathcal{H}_2$ analysis of continuous time systems. *IEEE Trans Autom Control* 44(2):239–252
- Paganini F, Feron E (1999) LMI methods for robust $\mathcal{H}_2$ analysis: a survey with comparisons. In: El Ghaoui L, Niculescu S (Eds) Recent advances on LMI methods in control. SIAM, Philadelphia
- Sánchez-Peña R, Szaier M (1998) Robust systems theory and applications. Wiley, New York
- Scherer C, Gahinet P, Chilali M (1997) Multiobjective output-feedback control via LMI-optimization. *IEEE Trans Autom Control* 42:896–911
- Stoorvogel AA (1993) The robust $\mathcal{H}_2$ control problem: a worst-case design. *IEEE Trans Autom Control* 38(9):1358–1370
- Szaier M, Rotstein H, Bu J, Sideris A (2000) An exact solution to continuous-time mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control problems. *IEEE Trans Autom Control* 45(11):2095–2101
- Szaier M, Amishima T, Parrilo PA, Tierno J (2002) A convex approach to robust $\mathcal{H}_2$ performance analysis. *Automatica* 38:957–966
- Zhou K, Glover K, Bodenheimer B, Doyle J (1994) Mixed $\mathcal{H}_2$ and $\mathcal{H}_\infty$ performance objectives I: robust performance analysis. *IEEE Trans Autom Control* 39(8):1564–1574
- Zhou K, Doyle J, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River

Robust Model Predictive Control

R

Saša V. Raković

School of Automation, Beijing Institute of Technology, Beijing, China

Abstract

Model predictive control (MPC) is indisputably one of the rare modern control techniques that has significantly affected control engineering practice due to its natural ability to systematically handle constraints and optimize performance. Robust MPC is an improved form of MPC that is intrinsically

robust in the face of uncertainty. The main goal of robust MPC is to devise an optimization-based control synthesis method that accounts for the intricate interactions of the uncertainty with the system, constraints, and performance criteria in a theoretically rigorous and computationally tractable way.

Keywords

Model predictive control · Robust model predictive control · Robust optimal control · Robust stability

Introduction

Robust MPC synthesis (Raković 1233) is best seen as a repetitive decision-making process, in which the underlying decision-making process is a robust optimal control (ROC) problem. The ROC problem is formulated in such a way so as to guarantee that all possible predictions of the controlled state and control actions sequences satisfy constraints and that the “worst-case” cost is minimized. The decision variable in the underlying ROC problem is a control policy, which is a sequence of control laws and which enables different control actions at different predicted states. Robust MPC makes use of the solution to the associated ROC problem in a recursive manner in order to implement the feedback control law, which is, in fact, equal to the first control law of an optimal control policy. A theoretically exact robust MPC can be obtained either by employing, in a repetitive fashion, the dynamic programming solution of the ROC problem or by solving online, in a recursive manner, corresponding infinite-dimensional minimaximization. The associated computational complexity has considerably hampered practical applicability of exact robust MPC, and it has made robust MPC an active research field with the prime challenge to devise design methods that adequately handle the effects of the uncertainty and yet are computationally plausible.

Recent research in MPC has witnessed advances across areas of robust, stochastic, and adaptive MPC. Research in these areas

has led to a plethora of MPC algorithms that address specific forms of uncertainty affecting the system under consideration. It is somewhat an unfortunate fact that a number of recent proposals blur the boundaries between robust, stochastic, and adaptive MPC syntheses, which strictly speaking belong to distinctly different mathematical classes. Failing to recognize inevitable heterogeneity of MPC under uncertainty syntheses exhibits potential misunderstanding of appropriate approach to handling different types of uncertainty. The information available about the uncertainty dictates directly the theoretically correct class of MPC under uncertainty.

Understanding MPC Under Uncertainty

The uncertainty types underpinning robust, stochastic, and adaptive MPC are illustrated by a scalar discrete time system

$$x^+ = x + u + w \text{ with } w \in \mathbb{W} := \{-1, 0, 1\},$$

where x is the state, u is the control, w is the uncertainty, and x^+ is the successor state.

Robust MPC is concerned with the set-membership uncertainty, for which the underlying available information on the uncertainty is that at any time instance $w_k \in \mathbb{W} = \{-1, 0, 1\}$. When at any time instance only state x_k is known, state feedback control laws $u_k(\cdot)$ with values $u_k(x_k)$ are structurally permissible, and this information pattern leads to a *minimax* MPC problem. Likewise, when at any time instance both state x_k and uncertainty w_k are known, state-uncertainty feedback control laws $u_k(\cdot, \cdot)$ with values $u_k(x_k, w_k)$ are structurally permissible, and this information pattern results in a *maximin* MPC problem. To illustrate the difference between the minimax and maximin cases, consider the distance function of x^+ from the origin. The optimal minimax distance is attained by control law $u(x) = -x$, and it is equal to 1 for all x , while the optimal maximin distance

is attained by control law $u(x, w) = -x - w$ and it is equal to 0 for all x .

Stochastic MPC considers the case in which a probabilistic information about uncertainty is provided. Typically, uncertainty process is assumed to be stationary and uncorrelated, and stochastic characteristics of the underlying uncertainty are known. For our example, it could be given that $p(w = -1) = 10^{-9}$, $p(w = 1) = 10^{-9}$ and $p(w = 0) = 1 - 2 \cdot 10^{-9}$. In this case, the underlying controlled state process is itself a stochastic process whose essential characteristics can be propagated compatibly with the system equation, employed control functions that should utilize informationally consistent stochastic characteristics of the controlled state process, and prior probabilistic postulates on the initial state. It would be very odd to neglect available stochastic information about uncertainty and implement robust MPC instead of stochastic MPC as well as to compare robust and stochastic MPC, as these belong to mathematically incomparable categories.

In *adaptive MPC*, for a simplified but frequently considered setting, the model of the system, i.e., uncertainty, is unknown and constant. In turn, an implied information about uncertainty is that it actually takes on one of the three possible values $w = -1$, or $w = 0$, or $w = 1$. Any control feedback $u(x)$ allows for learning the value of uncertainty, which is easily deduced from the value of the successor state $w = x^+ - (x + u(x))$. Thus, model uncertainty, in this simplified example, vanishes after one step, and the system becomes purely deterministic after one step. It does not seem appropriate to employ either robust or stochastic MPC as control strategies for adaptive MPC.

Uncertainty Effect on Robust MPC Synthesis

We now dissect the uncertainty effect on robust MPC in a verbose mode and then formally introduce exact robust MPC.

The uncertainty type: The considered uncertainty type is the set-membership uncertainty.

Traditionally, robust MPC treats the minimax case in which, at any time instance k , the state x_k is known, while the values of the current and future uncertainty w_{k+i} are not known but are guaranteed to belong to a known uncertainty constraint set.

The control policy: Under minimax information pattern, state feedback control functions $u_k(\cdot)$, with values $u_k(x_k)$ at states x_k , are structurally permissible and theoretically correct. Thus, within the context of robust MPC, the use of a control policy, which is a sequence of state feedback control functions, is structurally permissible and theoretically flawless.

The generalized state and control predictions: When x and $u(x)$ are the current state and control, the successor state x^+ can take any value in the set of possible successor states, where each possible successor state arises due to a particular uncertainty realization. This set-valued nature of possible successor states propagates through predictions. Consequently, it is necessary to employ generalized predictions in terms of sets of possible states and sets of related control actions.

The controlled dynamics under uncertainty: The controlled dynamics under uncertainty is the dynamics of sets of possible states, which is the controlled set-to-set-dynamics.

The robust constraint satisfaction: The robust constraint satisfaction refers to the requirement that the actual constraints are satisfied for all possible states and associated control actions arising due to all possible uncertainty realizations.

The “worst-case” performance: It is natural to minimize the “worst-case” performance over all uncertainty realizations. The minimization of the “worst-case” performance is done by a selection of a suitable control policy, and it can be also attained by minimization of an adequate, generalized cost function.

The generalized state and control origins: In minimax case of our example, only the stabilization of the state set $\{-1, 0, 1\}$ is possible, and keeping states therein requires control actions from the control set $\{-1, 0, 1\}$. In robust MPC, the state and control origins need to be considered

in a generalized sense as suitably defined state and control sets.

Exact Robust MPC: Contemporary Setting

The system: The most common setting in robust MPC considers the control systems modeled, in discrete time, by:

$$x^+ = f(x, u, w), \quad (1)$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $w \in \mathbb{R}^p$, and $x^+ \in \mathbb{R}^n$ are, respectively, the current state, control, uncertainty, and the successor state, while $f(\cdot, \cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ is the state transition map assumed to be continuous.

The constraints: The system variables x , u , and w are subject to hard constraints:

$$(x, u, w) \in \mathbb{X} \times \mathbb{U} \times \mathbb{W}. \quad (2)$$

The state and control constraint sets $\mathbb{X} \subseteq \mathbb{R}^n$ and $\mathbb{U} \subseteq \mathbb{R}^m$ are assumed to be closed, while the uncertainty constraint set $\mathbb{W} \subset \mathbb{R}^p$ is assumed to be compact.

The control policy: The use of a control policy

$$\Pi_{N-1} := \{\pi_0(\cdot), \pi_1(\cdot), \dots, \pi_{N-1}(\cdot)\}, \quad (3)$$

where N is the prediction horizon and each $\pi_k(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a control law, is structurally permissible and desirable.

The generalized state and control predictions: The interaction of the uncertainty with the system is captured by invoking the maps $F(\cdot, \cdot)$ and $G(\cdot, \cdot)$ specified, for any subset X of $\mathbb{R}^n$ and any control function $\kappa(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$, by:

$$\begin{aligned} F(X, \kappa) &:= \{f(x, \kappa(x), w) : x \in X, w \in \mathbb{W}\} \\ \text{and } G(X, \kappa) &:= \{\kappa(x) : x \in X\}. \end{aligned} \quad (4)$$

The state and control predictions are set-valued and, for each relevant k , obey the relations:

$$\begin{aligned} X_{k+1} &= F(X_k, \pi_k) \text{ and } U_k = G(X_k, \pi_k), \text{ with} \\ X_0 &:= \{x\}. \end{aligned} \quad (5)$$

The set sequences $X_N := \{X_0, X_1, \dots, X_{N-1}, X_N\}$ and $U_{N-1} := \{U_0, U_1, \dots, U_{N-1}\}$ represent the predicted sets of possible states and related control actions, and these set sequences are commonly known as the state and control tubes.

The robust constraint satisfaction: The robust constraint satisfaction reduces to the conditions that, for all $k = 0, 1, \dots, N-1$, the set inclusions

$$X_k \subseteq \mathbb{X} \text{ and } U_k \subseteq \mathbb{U} \quad (6)$$

hold true and that the set of possible states X_N at the prediction time instance N satisfies the set inclusion:

$$X_N \subseteq \mathbb{X}_f, \quad (7)$$

where $\mathbb{X}_f \subseteq \mathbb{X}$ is a suitable terminal constraint set.

The terminal constraint set: The terminal constraint set is obtained by considering the uncertain dynamics:

$$x^+ = f(x, \kappa_f(x), w) \quad (8)$$

controlled by a local control function $\kappa_f(\cdot)$. The design of a control law $\kappa_f(\cdot)$ is usually performed offline, and the terminal constraint set $\mathbb{X}_f$ accounts locally for the state and control constraints. The terminal constraint set $\mathbb{X}_f$ is assumed to be closed and robust positively invariant for the dynamics (8) and constraint sets (2). Thus, the set $\mathbb{X}_f$ and a local control function $\kappa_f(\cdot)$ satisfy:

$$\begin{aligned} F(\mathbb{X}_f, \kappa_f) &\subseteq \mathbb{X}_f \subseteq \mathbb{X} \text{ and} \\ \mathbb{U}_f &:= G(\mathbb{X}_f, \kappa_f) \subseteq \mathbb{U}. \end{aligned} \quad (9)$$

The best choice for $\mathbb{X}_f$ is the maximal robust positively invariant set for the dynamics (8) and constraint sets (2).

The generalized origin: The most natural candidate for the generalized origin is a minimal

robust positively invariant set for the dynamics (8) and constraint sets (2). This set is entirely determined by the associated state set–dynamics

$$X^+ = F(X, \kappa_f). \quad (10)$$

The minimal robust positively invariant set is compact and well defined in the case when the local control function $\kappa_f(\cdot)$ ensures that the corresponding map $F(\cdot, \kappa_f)$ is a contraction over the space of compact subsets of $\mathbb{X}_f$ (Artstein and Raković 2008), in which case it is the unique solution to the fixed point set equation:

$$X = F(X, \kappa_f), \quad (11)$$

and it is an exponentially stable attractor for the state set–dynamics (10) with the basin of attraction being the space of compact subsets of $\mathbb{X}_f$. Thus, the usual $(0, 0)$ fixed-point pair is replaced by the fixed-point pair of sets $(\mathbb{X}_O, \mathbb{U}_O)$ satisfying:

$$\begin{aligned} \mathbb{X}_O &= F(\mathbb{X}_O, \kappa_f) \subseteq \mathbb{X}_f \text{ and} \\ \mathbb{U}_O &:= G(\mathbb{X}_O, \kappa_f) \subseteq \mathbb{U}_f. \end{aligned} \quad (12)$$

The generalized cost function: The stage cost function $\ell(\cdot, \cdot) : \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{R}_+$ is continuous and adequately lower bounded with respect to the generalized origin $\mathbb{X}_O$ so that, for all $x \in \mathbb{X}$ and all $u \in \mathbb{U}$,

$$\alpha_1(\text{dist}(\mathbb{X}_O, x)) \leq \ell(x, u), \quad (13)$$

where $\alpha_1(\cdot)$ is a $\mathcal{K}_\infty$ –class function and $\text{dist}(\mathbb{X}_O, \cdot)$ is the distance function from the set $\mathbb{X}_O$. The consideration of the generalized origin requires the condition that for all $x \in \mathbb{X}_O$ the use of local control function $\kappa_f(\cdot)$ is “free of charge” with respect to $\ell(\cdot, \cdot)$, i.e., that for all $x \in \mathbb{X}_O$ we have:

$$\ell(x, \kappa_f(x)) = 0. \quad (14)$$

The terminal cost function $V_f(\cdot)$ is assumed to be continuous and adequately upper bounded with respect to the generalized origin $\mathbb{X}_O$ so that, for all $x \in \mathbb{X}_f$,

$$V_f(x) \leq \alpha_2(\text{dist}(\mathbb{X}_O, x)), \quad (15)$$

where, as above, $\alpha_2(\cdot)$ is a $\mathcal{K}_\infty$ –class function. In addition, the terminal cost function $V_f(\cdot)$ satisfies locally a usual condition for robust stabilization, so that, for all $x \in \mathbb{X}_f$ and all $w \in \mathbb{W}$,

$$V_f(f(x, \kappa_f(x), w)) - V_f(x) \leq -\ell(x, \kappa_f(x)). \quad (16)$$

The cost function $V_N(\cdot, \cdot, \cdot)$ is defined, for all $x \in \mathbb{X}$, all Π_{N-1} and all $\mathbf{w}_{N-1} := \{w_0, w_1, \dots, w_{N-1}\}$, by

$$V_N(x, \Pi_{N-1}, \mathbf{w}_{N-1}) := \sum_{k=0}^{N-1} \ell(x_k, u_k) + V_f(x_N), \quad (17)$$

where $u_k := \pi_k(x_k)$ and $x_k := x_k(x, \Pi_{N-1}, \mathbf{w}_{N-1})$ denote the solution of (1) when the initial state is x , control policy is Π_{N-1} and uncertainty realization is $\mathbf{w}_{N-1}$.

The exact ROC: The exact ROC problem $\mathbb{P}_N(x)$, $x \in \mathbb{X}$, takes the form of an infinite-dimensional minimaximization:

$$\begin{aligned} J_N(x, \Pi_{N-1}) &:= \max_{\mathbf{w}_{N-1} \in \mathbb{W}^N} V_N(x, \Pi_{N-1}, \mathbf{w}_{N-1}), \\ V_N^0(x) &:= \min_{\Pi_{N-1} \in \mathbf{\Pi}_{N-1}(x)} J_N(x, \Pi_{N-1}), \\ \Pi_{N-1}^0(x) &\in \arg \min_{\Pi_{N-1} \in \mathbf{\Pi}_{N-1}(x)} J_N(x, \Pi_{N-1}), \end{aligned} \quad (18)$$

where $\mathbf{\Pi}_{N-1}(x)$ denotes the set of the constraint admissible control policies defined, for all $x \in \mathbb{X}$, by:

$$\mathbf{\Pi}_{N-1}(x) := \{\Pi_{N-1} : \text{conditions (5)–(7) hold}\}. \quad (19)$$

With the intended scope of this article in mind, it is assumed that the infinite-dimensional minimaximization is well posed and that the value function $V_N^0(\cdot)$ admits an upper bound in terms of a suitable $\mathcal{K}_\infty$ –class function as specified in (23).

The value function $V_N^0(\cdot)$ might admit nonunique optimal control policies, so that $\Pi_{N-1}^0(\cdot)$ represents a selection from the set of optimal control policies. The effective domain

$\mathcal{X}_N$ of the value function $V_N^0(\cdot)$ and optimal control policy $\Pi_{N-1}^0(\cdot)$,

$$\mathcal{X}_N := \{x \in \mathbb{R}^n : \Pi_{N-1}(x) \neq \emptyset\} \quad (20)$$

is known in the literature as the N -step minimax controllable set to a target set $\mathbb{X}_f$, and it is such that $\mathbb{X}_f \subseteq \mathcal{X}_N \subseteq \mathbb{X}$.

The exact robust MPC: The exact robust MPC implements values of the control law $\pi_0^0(\cdot)$. The control law $\pi_0^0(\cdot)$ is well defined for all $x \in \mathcal{X}_N$, and it induces the controlled uncertain dynamics specified, for all $x \in \mathcal{X}_N$, by:

$$x^+ \in \mathcal{F}(x), \quad \mathcal{F}(x) := \{f(x, \pi_0^0(x), w) : w \in \mathbb{W}\}. \quad (21)$$

The robust MPC law $\pi_0^0(\cdot)$ renders the N -step minimax controllable set $\mathcal{X}_N$ robust positively invariant so that, for all $x \in \mathcal{X}_N$,

$$\mathcal{F}(x) \subseteq \mathcal{X}_N \subseteq \mathbb{X} \text{ and } \pi_0^0(x) \in \mathbb{U}. \quad (22)$$

Furthermore, the associated value function $V_N^0(\cdot) : \mathcal{X}_N \rightarrow \mathbb{R}_+$ is, by construction, a Lyapunov certificate verifying the robust asymptotic stability of the generalized origin $\mathbb{X}_\mathcal{O}$ for the controlled uncertain dynamics (21) with the basin of attraction being equal to the N -step minimax controllable set $\mathcal{X}_N$. More precisely, for all $x \in \mathcal{X}_N$, it holds that:

$$\alpha_1(\text{dist}(\mathbb{X}_\mathcal{O}, x)) \leq V_N^0(x) \leq \alpha_3(\text{dist}(\mathbb{X}_\mathcal{O}, x)), \quad (23)$$

where $\alpha_3(\cdot)$ is postulated $\mathcal{K}_\infty$ -class function, while for all $x \in \mathcal{X}_N$ and all $x^+ \in \mathcal{F}(x)$, it holds that:

$$V_N^0(x^+) - V_N^0(x) \leq -\alpha_1(\text{dist}(\mathbb{X}_\mathcal{O}, x)). \quad (24)$$

Under reasonable conditions, exact robust MPC induces strong structural properties, but its computational complexity is overwhelming. However, in the preceding formulation, uncertainty effects have been “dissected,” and the “basic building blocks” employed for exact robust MPC have been identified.

Alternative Approaches to Robust MPC

Inherent Robustness of MPC

An approach to robust MPC can be derived by attempting to utilize inherent robustness of conventional MPC, see, e.g., Limon et al. (2009) and related references. Such approach is an indirect approach to robust MPC, and it is computationally feasible. This approach considers a nominal MPC possibly applied to modified model, constraints, and cost, and when possible it makes use of robustness of its stability properties. The approach might be well suited for problems with neither state nor terminal constraints. However, the robustness properties can be very limited in terms of the magnitude of tolerable uncertainty. In addition, the related robustness analysis is unnecessarily conservative and geometrically insensitive, i.e., it is of very little value for optimal state trajectories on, or near, the boundaries of the state constraints. Major drawbacks of such an approach, however, stem from the facts that the stability property of the conventional MPC might fail to be robust (Grimm et al. 2004) and that the optimal control of constrained discrete time systems can be a fragile process itself (Raković 2009).

Scenario-Based Approaches for Robustness

The approach based on utilization of scenarios of uncertainty realizations is particularly very well suited for the case of linear systems with convex cost functions subject to convex constraints and when the uncertainty set is specified as the convex hull of finitely many points (Scokaert and Mayne 1998). In this approach, with each of finitely many extreme uncertainty realizations, a pair of “extreme” state and control sequences is associated. The collection of all pairs of such “extreme” state and control sequences is optimized subject to dynamical constraints (which are deterministic since uncertainty realizations are postulated), causality constraints, and state, control, and terminal constraints. This approach converts the infinite-dimensional minimaximization

tion into an equivalent finite-dimensional minimaximization, but it remains computationally intractable due to an exponential increase of the number of extreme uncertainty realizations with respect to the prediction horizon.

Some recent proposals consider a limited number of uncertainty realizations, and related scenarios are constructed by allowing uncertainty to vary over a shorter horizon, while uncertainty is considered constant over the remainder of the prediction horizon. Such approaches enhance computational aspects and can be used in more general settings, see, e.g., Lucia et al. (2012) and related references. Theoretically, it is difficult to justify the postulated structural restriction on the scenarios of uncertainty realizations. Furthermore, it is highly unlikely for such approaches to guarantee a priori any of the desirable structural properties, as provided by alternative robust MPC methods.

A very promising approach is to deploy scenario-based optimization for robust MPC (Calafiore and Campi 2006; Calafiore and Fagiano 2013), in which uncertainty is sampled in a traditional manner. These approaches provide probabilistic guarantees of structural properties that are strictly weaker than the absolute ones guaranteed by robust MPC.

Parameterized Predictions and Control Policy

The parameterization of state and control predictions as well as control policy has led to major advances in robust MPC. The core simplification in these approaches is the use of finite-dimensional parameterization of control policy. The control policy is parameterized so as to allow for the utilization of both the least conservative parameterized state and control predictions and a range of simpler, but sensible, cost functions. The computational complexity of the exact state and control tubes is tackled by employing implicit representations of the predicted sets of possible states and control actions. Alternatively, simpler outer-bounding approximations of the exact state and control tubes are deployed, and the exact set-dynamics of the state and control tubes (5) is relaxed to

$$\{x_0\} \subseteq X_0, \text{ and } F(X_k, \pi_k) \subseteq X_{k+1}$$

$$\text{and } G(X_k, \pi_k) \subseteq U_k.$$

The terminal constraint set $\mathbb{X}_f$ is deployed as computationally feasible and “relaxed form” of the generalized state origin $\mathbb{X}_O$. The performance requirements are carefully prioritized, and modified when necessary, and expressed in terms of the cost functions that do not require intractable minimax optimization but still ensure that the associated value function verifies the robust stability and attractivity of the generalized origin $\mathbb{X}_O$ or the terminal constraint set $\mathbb{X}_f$.

Tube MPC

A new paradigm in robust MPC that has effectively defined modern robust MPC is tube MPC. Tube MPC has been pioneered and advanced primarily through joint and individual research activities of David Q. Mayne and Saša V. Raković. For completeness, we provide a brief overview of tube MPC for linear systems subject to bounded additive uncertainty

$$x^+ = Ax + Bu + w. \quad (25)$$

In what follows, $\oplus$ denotes the Minkowski set addition, while $\ominus$ denotes the Pontryagin set difference.

Rigid Tube MPC

In rigid tube MPC, introduced and discussed in detail in Mayne et al. (2005), state and control predictions are parameterized as

$$x_k = z_k + s_k \text{ and } u_k = v_k + K s_k, \quad (26)$$

and state feedback control laws of related control policy are

$$\pi_k(x_k, z_k, v_k) = v_k + K(x_k - z_k). \quad (27)$$

The sets forming state and control tubes are parameterized as

$$X_k = z_k \oplus S \text{ and } U_k = v_k \oplus KS. \quad (28)$$

The state tube cross-sectional shape set S is assumed to be robust positively invariant set for $s^+ = (A + BK)s + w$, i.e.,

$$(A + BK)S \oplus \mathbb{W} \subseteq S. \quad (29)$$

Rigid tube MPC is computationally efficient since it reduces to a conventional MPC for a virtual deterministic linear system

$$z_{k+1} = Az_k + Bv_k \quad (30)$$

subject to modified constraints

$$z_k \in \mathbb{Z} := \mathbb{X} \ominus S \text{ and } v_k \in \mathbb{V} := \mathbb{U} \ominus KS, \quad (31)$$

and suitable terminal constraints $z_N \in \mathbb{Z}_f$ as well as uncertainty free stage and terminal cost functions $\ell(\cdot, \cdot)$ and $V_f(\cdot)$ with values $\ell(z_k, v_k)$ and $V_f(z_N)$. Rigid tube MPC enables a flexible selection of free initial condition of the virtual system

$$x - z_0 \in S \quad (32)$$

at any state x . This condition proves to be a key ingredient in establishing robust exponential stability of the set S for uncertain dynamics controlled by rigid tube MPC $u_0^0(x) = v_0^0(x) + K(x - z_0^0(x))$, based on the exponential stability of the origin for the virtual system $z^+(x) = Az_0^0(x) + Bv_0^0(x)$.

Homothetic Tube MPC

Homothetic tube MPC (Raković et al. 2012, 2013) relaxes rigidity of state and control tubes parameterization in (28) by deploying state and control tubes whose terms are parameterized as

$$X_k = z_k \oplus \alpha_k S \text{ and } U_k = v_k \oplus \alpha_k KS, \quad (33)$$

where nonnegative scalars $\alpha_k \in \mathbb{R}_{\geq 0}$ are referred to as the state and control tubes scaling factors. The exact set-dynamics of the state tubes is utilized, and it is expressed as

$$Az_k + Bv_k + \alpha_k(A + BK)S \oplus \mathbb{W}$$

$$\subseteq z_{k+1} \oplus \alpha_{k+1} S, \quad (34)$$

while state and control constraints take form

$$z_k \oplus \alpha_k S \subseteq \mathbb{X} \text{ and } v_k \oplus \alpha_k KS \subseteq \mathbb{U}, \quad (35)$$

and adequate terminal constraints $(z_N, \alpha_N) \in \Omega_f$ as well as uncertainty free stage and terminal cost functions $\ell(\cdot, \cdot, \cdot)$ and $V_f(\cdot, \cdot)$ with values $\ell(z_k, v_k, \alpha_k)$ and $V_f(z_N, \alpha_N)$. The selection of z_0 and α_0 is governed by

$$x - z_0 \in \alpha_0 S. \quad (36)$$

Homothetic tube MPC, a detailed summary of which can be found in Raković et al. (2012, 2013), enlarges domain of attraction and improves attractivity properties compared to rigid tube MPC.

Elastic Tube MPC

Elastic tube MPC (Raković et al. 2016) further relaxes rigidity and homotheticity of tubes parameterizations in (28) and (33). In elastic tube MPC, the state tube cross-sectional shape set is, in fact, a value of a set-valued function $S(\cdot)$ given, for all $a \in \mathbb{R}_{\geq 0}^q$, by

$$S(a) := \{x : Cx \leq a\}, \quad (37)$$

where a suitable matrix $C \in \mathbb{R}^{q \times n}$ is constructed offline. The terms of state and control tubes are parameterized as

$$X_k = z_k \oplus S(a_k) \text{ and } U_k = v_k \oplus KS(a_k), \quad (38)$$

where nonnegative variables a_k are referred to as the state and control tubes elasticity factors. The exact set-dynamics of the state tubes is given by

$$\begin{aligned} & Az_k + Bv_k + (A + BK)S(a_k) \oplus \mathbb{W} \\ & \subseteq z_{k+1} \oplus S(a_{k+1}), \end{aligned} \quad (39)$$

while state and control constraints take form

$$z_k \oplus S(a_k) \subseteq \mathbb{X} \text{ and } v_k \oplus KS(a_k) \subseteq \mathbb{U}, \quad (40)$$

and suitable terminal constraints $(z_N, a_N) \in \Omega_f$ as well as uncertainty free stage and terminal cost functions $\ell(\cdot, \cdot, \cdot)$ and $V_f(\cdot, \cdot)$ with values $\ell(z_k, v_k, a_k)$ and $V_f(z_N, a_N)$. To ensure computational practicability, guaranteed and convex reformulations of set-dynamics (39) and state and control constraints (40) are utilized, while z_0 and a_0 satisfy

$$x - z_0 \in S(a_0). \quad (41)$$

Elastic tube MPC enlarges domain of attraction and improves attractivity properties relative to both the homothetic and rigid tube MPC. See Raković et al. (2016) for more details about elastic tube MPC.

Tube MPC with Time-Varying Cross Sections

Robust MPC based on constraint tightening proposed in Chisci et al. (2001) can be seen as a tube MPC with time-varying cross sections, since related sets of possible state and controls take the form

$$X_k = z_k \oplus S_k \text{ and } U_k = v_k \oplus K S_k, \quad (42)$$

where

$$S_{k+1} = (A + BK)S_k \oplus \mathbb{W}. \quad (43)$$

As proposed in Chisci et al. (2001), this form of tube MPC utilizes virtual system (30) and time-varying state and control constraints

$$z_k \in \mathbb{Z}_k := \mathbb{X} \ominus S_k \text{ and } v_k \in \mathbb{V}_k := \mathbb{U} \ominus K S_k, \quad (44)$$

and terminal constraints $z_N \oplus S_N \subseteq \mathbb{X}_f$ where $\mathbb{X}_f$ is the maximal robust positively invariant set for $s^+ = (A + BK)s + w$ and constraints $(s, Ks) \in \mathbb{X} \times \mathbb{U}$ and $w \in \mathbb{W}$. The proposal of Chisci et al. (2001) minimizes the cost in terms of “control perturbations” $c_k = v_k + K z_k$, and it enforces a restrictive constraint on the initial states $z_0 = x$, i.e., $S_0 = \{0\}$, which prevents robust exponential stability to be established. However, this proposal can be modified to include selection of initial condition of the virtual system via $x - z_0 \in \mathbb{X}_f$

in analogy to (32), which would enable robust exponential stability to be verified.

Tube MPC with Optimized Time-Varying Cross Sections

A further contribution to robust MPC is made in Löfberg (2003) and Goulart et al. (2006), where the so-called affine in the past disturbances control policy is utilized. Within the context of tube MPC, the related parameterizations of predicted states and controls are

$$\begin{aligned} x_k &= z_k + \sum_{j=1}^k T_{(j,k)} w_j \text{ and} \\ u_k &= v_k + \sum_{j=1}^k M_{(j,k)} w_j, \end{aligned} \quad (45)$$

where matrices $T_{(j,k)}$ and $M_{(j,k)}$ are decision variables and the related sets of possible states and controls take the form

$$\begin{aligned} X_k &= z_k \oplus \bigoplus_{j=1}^k T_{(j,k)} \mathbb{W} \text{ and} \\ U_k &= v_k \oplus \bigoplus_{j=1}^k M_{(j,k)} \mathbb{W}. \end{aligned} \quad (46)$$

The state tube dynamics is governed by deterministic z -dynamics (30) and matrix-valued $T_{(j,k)}$ -dynamics

$$T_{(j,k+1)} = AT_{(j,k)} + BM_{(j,k)} \text{ with } T_{(j,j)} = I. \quad (47)$$

The state and control constraints are given by

$$\begin{aligned} z_k \oplus \bigoplus_{j=1}^k T_{(j,k)} \mathbb{W} &\subseteq \mathbb{X} \text{ and} \\ v_k \oplus \bigoplus_{j=1}^k M_{(j,k)} \mathbb{W} &\subseteq \mathbb{U}, \end{aligned} \quad (48)$$

and terminal constraints are $z_N \oplus \bigoplus_{j=1}^N T_{(j,N)} \mathbb{W} \subseteq \mathbb{X}_f$. The polyhedral state, control, and terminal constraints can be efficiently

handled by using support functions and duality. A range of uncertainty free cost functions can be utilized to design a robust exponentially stable tube MPC via efficient online optimization of the uncertainty free states z_k and controls v_k and collections of matrices $T_{(j,k)}$ and $M_{(j,k)}$.

Parameterized Tube MPC

The current state of the art in robust MPC is parameterized tube MPC (Raković 2012; Raković et al. 2012), which employs superposition and convex decomposition for predictions of states and controls

$$x_k = \sum_{j=0}^k x_{(j,k)} \text{ and } u_k = \sum_{j=0}^k u_{(j,k)}, \text{ with } (49)$$

$$x_{(j,k)} = \sum_{i=0}^q \lambda_i x_{(i,j,k)} \text{ and } u_{(j,k)} = \sum_{i=0}^q \lambda_i u_{(i,j,k)}, \quad (50)$$

via partial extreme states and controls $x_{(0,k)}$, $x_{(i,j,k)}$, $u_{(0,k)}$, and $u_{(i,j,k)}$, and where $\lambda_i = \lambda_{(i,j,j)}$ are convex multipliers. The sets of possible states and controls take the form

$$X_k = \bigoplus_{j=0}^k X_{(j,k)} \text{ and } U_k = \bigoplus_{j=0}^k U_{(j,k)}, \quad (51)$$

where partial state and control tubes cross sections are given as $X_{(0,k)} = \{x_{(0,k)}\}$, $X_{(j,k)} = \text{convh}(\{x_{(1,j,k)}, x_{(2,j,k)}, \dots, x_{(q,j,k)}\})$, $U_{(0,k)} = \{u_{(0,k)}\}$, and $U_{(j,k)} = \text{convh}(\{u_{(1,j,k)}, u_{(2,j,k)}, \dots, u_{(q,j,k)}\})$. The exact set-dynamics of the state tubes is governed by deterministic extreme partial states dynamics specified by

$$\begin{aligned} x_{(0,k+1)} &= Ax_{(0,k)} + Bu_{(0,k)} \text{ with } x_{(0,0)} = x \text{ and} \\ x_{(i,j,k+1)} &= Ax_{(i,j,k)} + Bu_{(i,j,k)} \text{ with } x_{(i,j,j)} = \bar{w}_i, \end{aligned} \quad (52)$$

where $\mathbb{W} = \text{convh}(\{\bar{w}_1, \bar{w}_2, \dots, \bar{w}_q\})$. The state and control constraints are given by

$$\bigoplus_{j=0}^k X_{(j,k)} \subseteq \mathbb{X} \text{ and } \bigoplus_{j=0}^k U_{(j,k)} \subseteq \mathbb{U}, \quad (53)$$

and terminal constraints are $\bigoplus_{j=0}^N X_{(j,N)} \subseteq \mathbb{X}_f$. The polyhedral state, control, and terminal constraints can be efficiently handled by using support functions without need to perform any of Minkowski set additions and convex hull operations utilized for parameterization and analysis. A range of uncertainty free cost functions can be utilized to design a robust exponentially stable tube MPC via efficient online optimization of the sequences of extreme partial states $x_{(0,k)}$ and $x_{(i,j,k)}$ and extreme partial controls $u_{(0,k)}$ and $u_{(i,j,k)}$. Parameterized tube MPC outperforms proposals of Löfberg (2003) and Goulart et al. (2006), as it uses separable nonlinear state feedback control functions.

Closing Remarks and Recommended Reading

The exact robust MPC synthesis has reached a remarkable degree of theoretical maturity in the general setting. Furthermore, a variety of rather sophisticated robust MPC synthesis methods, which are both computationally efficient and theoretically sound, has been developed for the frequently encountered linear-polyhedral case. The further advances in the robust MPC field should be driven by the utilization of more structured types and models of the uncertainty. Given a tremendous progress in robust MPC and its successful utilization across a number of control engineering areas, a comprehensive survey of this field is definitely due.

The comprehensive monograph (Rawlings and Mayne 2009) provides an in-depth systematic exposure to MPC and robust MPC, and it is also a rich source of relevant references. The invaluable overview of the theory and computations of the maximal and minimal robust positively invariant sets can be found in Kolmanovsky and Gilbert (1998) and Artstein and Raković (2008). The important work Scokaert and Mayne (1998) points out the theoretical benefits of the use of the control policy, but it also indicates indirectly the computational impracticability of the associated feedback minimax robust

MPC. The early tube MPC synthesis (Langson et al. 2004; Mayne et al. 2005) represent key steps forward in robust MPC for the linear–polyhedral setting. Homothetic and elastic tube MPC syntheses (Raković et al. 2012, 2013, 2016) provide a number of improvements relative to the first generation of tube MPC (Mayne et al. 2005), and these methods have a high potential to effectively handle the parametric uncertainty of the matrix pair (A, B) . The current state of the art in the linear–polytopic setting is reached by parameterized tube MPC (Raković 2012; Raković et al. 2012). The output feedback robust MPC synthesis in the linear–polyhedral setting can be handled with direct extensions of the tube MPC syntheses (Mayne et al. 2006, 2009).

Cross-References

- [Model Predictive Control in Practice](#)
- [Nominal Model-Predictive Control](#)
- [Stochastic Model Predictive Control](#)
- [Tracking Model Predictive Control](#)

Bibliography

- Artstein Z, Raković SV (2008) Feedback and invariance under uncertainty via set iterates. *Automatica* 44(2):520–525
- Calafiore GC, Campi MC (2006) The scenario approach to robust control design. *IEEE Trans Autom Control* 51:742–753
- Calafiore GC, Fagiano L (2013) Robust model predictive control via scenario optimization. *IEEE Trans Autom Control* 56:219–224
- Chisci L, Rossiter JA, Zappa G (2001) Systems with persistent disturbances: predictive control with restricted constraints. *Automatica* 37:1019–1028
- Goulart PJ, Kerrigan EC, Maciejowski JM (2006) Optimization over state feedback policies for robust control with constraints. *Automatica* 42(4):523–533
- Grimm G, Messina MJ, Tuna SE, Teel AR (2004) Examples when nonlinear model predictive control is nonrobust. *Automatica* 40:1729–1738
- Kolmanovsky IV, Gilbert EG (1998) Theory and computation of disturbance invariant sets for discrete time linear systems. *Math Probl Eng: Theory Methods Appl* 4:317–367
- Langson W, Chrysoschoos I, Raković SV, Mayne DQ (2004) Robust model predictive control using tubes. *Automatica* 40:125–133
- Limon D, Alamo T, Raimondo DM, noz de la Peña DM, Bravo JM, Ferramosca A, Camacho EF (2009) Input-to-state stability: a unifying framework for robust model predictive control. In: *Lecture notes in control and information sciences – nonlinear model predictive control: towards new challenging applications*, vol 384.
- Löfberg J (2003) Minimax approaches to robust model predictive control. Ph.D. dissertation, Department of Electrical Engineering, Linköping University, Linköping, Sweden
- Lucia S, Finkler T, Basak D, Engell S (2012) Robust model predictive control by scenario-based multi-stage optimization. In: *Proceedings of the 5th international conference on high performance scientific computing*, Hanoi, Vietnam.
- Mayne DQ, Seron M, Raković SV (2005) Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica* 41:219–224
- Mayne DQ, Raković SV, Findeisen R, Allgöwer F (2006) Robust output feedback model predictive control of constrained linear systems. *Automatica* 42:1217–122
- Mayne DQ, Raković SV, Findeisen R, Allgöwer F (2009) Robust output feedback model predictive control of constrained linear systems: time varying case. *Automatica* 45:2082–2087
- Raković SV (2015) Robust model–predictive control. In: Baillieul J, Samad T (eds) *Encyclopedia of systems and control*. Springer, London, pp 1225–1233
- Raković SV (2009) Set theoretic methods in model predictive control. In: *Lecture notes in control and information sciences – nonlinear model predictive control: towards new challenging applications*, vol 384, pp 41–54
- Raković SV (2012) Invention of prediction structures and categorization of robust mpc syntheses. In: *Proceedings of the IFAC conference on nonlinear model predictive control NMPC 2012*, Noordwijkerhout, the Netherlands, , Plenary Paper
- Raković SV, Kouvaritakis B, Findeisen R, Cannon M (2012) Homothetic tube model predictive control. *Automatica* 48:1631–1638
- Raković SV, Kouvaritakis B, Cannon M, Panos C, Findeisen R (2012) Parameterized tube model predictive control. *IEEE Trans Autom Control* 57:2746–2761
- Raković SV, Kouvaritakis B, Cannon M (2013) Equinormalization and exact scaling dynamics in homothetic tube model predictive control. *Syst Control Lett* 62(2):209–217
- Raković SV, Levine WS, Açıkmese B (2016) Elastic tube model predictive control. In: *Proceedings of the 2016 American control conference (ACC)*, Boston, MA, USA
- Rawlings JB, Mayne DQ (2009) *Model predictive control: theory and design*. Nob Hill Publishing, Madison
- Scokaert POM, Mayne DQ (1998) Min–max feedback model predictive control for constrained linear systems. *IEEE Trans Autom Control* 43:1136–1142

Robust Synthesis and Robustness Analysis Techniques and Tools

Gary Balas¹, Andrew Packard², and Peter Seiler¹

¹Aerospace Engineering and Mechanics
Department, University of Minnesota,
Minneapolis, MN, USA

²Mechanical Engineering Department,
University of California, Berkeley, CA, USA

Abstract

This entry provides a brief summary of the synthesis and analysis tools that have been developed by the robust control community. Many software tools have been developed to implement the major theoretical techniques in robust control. These software tools have enabled robust synthesis and analysis techniques to be successfully applied to numerous industrial applications.

Keywords

Integral quadratic constraint · Linear matrix inequality · Model uncertainty · Robust control toolbox · Structured singular values

Introduction

Robust control is a methodology to address the effect of uncertainty on feedback systems. This approach includes techniques and tools to model system uncertainty, assess stability and/or performance characteristics of the uncertain system, and synthesize controllers for uncertain systems. The theory was developed over a number of years. The foundational results can be found in classical papers Packard and Doyle (1993a), Desoer et al. (1980), Doyle (1978, 1982), Doyle et al. (1989), Doyle and Stein (1981), Megretski and Rantzer (1997), Safonov (1982), Willems (1971), and Zames (1981)

and more recent textbooks Boyd et al. (1994), Desoer and Vidyasagar (2008), Dullerud and Paganini (2000), Francis (1987), Skogestad and Postlethwaite (2005), Vidyasagar (1985), and Zhou et al. (1996). It should be emphasized that this entry is not meant to be a survey and more complete references to the literature can be found in the cited textbooks. The remainder of this entry discusses the main theoretical and computational tools for robust synthesis and robustness analysis.

Notation

$\mathbf{R}$ and $\mathbf{C}$ denote the set of real and complex numbers, respectively. $\mathbf{R}^{m \times n}$ and $\mathbf{C}^{m \times n}$ denote the sets of $m \times n$ matrices whose elements are in $\mathbf{R}$ and $\mathbf{C}$, respectively. A single superscript index is used for vectors, e.g., $\mathbf{R}^n$ denotes the set of $n \times 1$ vectors whose elements are in $\mathbf{R}$. For a matrix $M \in \mathbf{C}^{m \times n}$, M^T denotes the transpose and M^* denotes the complex conjugate transpose. A matrix M is Hermitian (Skew-Hermitian) if $M = M^*$ ($M = -M^*$). The maximum singular value of a matrix M is denoted by $\bar{\sigma}(M)$. The trace of a matrix M , denoted $\text{tr}[M]$, is the sum of the diagonal elements. $M = M^*$ is a positive semidefinite matrix, denoted $M \succeq 0$, if all eigenvalues are nonnegative. $M = M^*$ is negative semidefinite, denoted $M \preceq 0$, if $-M \succeq 0$. $\mathcal{L}_2^n[0, \infty)$ is the space of functions $u : [0, \infty) \rightarrow \mathbf{R}^n$ satisfying $\|u\| < \infty$ where $\|u\| := \left[\int_0^\infty u(t)^T u(t) dt \right]^{0.5}$. For $u \in \mathcal{L}_2^n[0, \infty)$, u_T denotes the truncated function $u_T(t) = u(t)$ for $t \leq T$ and $u(t) = 0$ otherwise. The extended space, denoted $\mathcal{L}_{2e}$, is the set of functions u such that $u_T \in \mathcal{L}_2$ for all $T \geq 0$. The Fourier transform $\hat{v} := \mathcal{F}(v)$ maps the time domain signal $v \in \mathcal{L}_2^n[0, \infty)$ to the frequency domain by

$$\hat{v}(j\omega) := \int_0^\infty e^{-j\omega t} v(t) dt \quad (1)$$

Capital letters are used to represent dynamical systems. For linear systems, the same letter is used to represent the system, its convolution kernel, as well as its frequency-response function. Lowercase letters denote time-signals, and

ω represents the continuous-time frequency variable. For an $m \times n$ system G , define the H_∞ and H_2 norms as $\|G\|_\infty = \sup_\omega \bar{\sigma}(G(j\omega))$ and $\|G\|_2 = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{tr}[G(j\omega)^* G(j\omega)] d\omega}$. The $\mathcal{L}_1$ norm of G is defined as $\|G\|_1 = \max_{1 \leq i \leq m} \sum_{j=1}^n \int_0^\infty |g_{ij}(t)| dt$ where $g_{ij}(t)$ is the response of the i th output due to a unit impulse in the j th input. The entry describes continuous-time systems. Most results carry over, in a similar form, to discrete-time systems.

Theoretical Tools

Uncertainty Modeling

In order to analyze and/or design for the degrading effects of uncertainty, it is imperative that explicit models of uncertainty be characterized. Two distinct forms of uncertainty are considered: signal uncertainty and model uncertainty. Signal uncertainty represents external signals (plant disturbances, sensor noise, reference signals) as sets of time functions, with explicit descriptions. For example, a particular reference input might be characterized as belonging to the set $\left\{ \frac{4}{2s+1} d : d \in \mathcal{L}_2, \|d\|_2 \leq 1 \right\}$. This set is often referred to as a *weighted ball in $\mathcal{L}_2$* . The transfer function $\frac{4}{2s+1}$ is called a *weighting function* and it shapes the normalized signals d , in a manner that its output represents the actual traits of the reference inputs that occur in practice.

Model uncertainty represents unknown or partially specified gains (more generally, operators) that relate pairs of signals in the model. For example, z and w are signals within a model and are related by an operator $\mathcal{N}$ as $w = \mathcal{N}(z)$. Typical partial specifications either constrain $\mathcal{N}$ to be drawn from a specified set or describe the set of signals (w, z) that $\mathcal{N}$ allows. An *uncertain parameter* δ is modeled as time-invariant (i.e., constant), belonging to the interval $[a, b]$ and relating z and w as $w(t) = \delta z(t)$. An *uncertain linear dynamic element*, Δ is modeled as linear, time-invariant, causal system, described by a convolution kernel δ whose frequency-response function (i.e., Fourier transform) satisfies $\max_\omega |\hat{\delta}(j\omega)| \leq 1$, and relating z and

w as $w = \delta \star z$. More generally, consider an $\mathcal{L}_2$ bounded, causal operator, mapping $\mathcal{L}_{2,e} \rightarrow \mathcal{L}_{2,e}$ relating the signals as $w = \Delta(z)$. The behavior of Δ is unknown but constrained by a family of multipliers, $\{\Pi_\alpha\}_{\alpha \in \mathcal{A}}$. Specifically, each Π_α is a Hermitian, matrix-valued function of frequency, and for any $z \in \mathcal{L}_2$, the mapping Δ is known to satisfy

$$\int_{-\infty}^{\infty} \begin{bmatrix} \hat{z}(j\omega) \\ \hat{w}(j\omega) \end{bmatrix}^* \Pi_\alpha(\omega) \begin{bmatrix} \hat{z}(j\omega) \\ \hat{w}(j\omega) \end{bmatrix} d\omega \geq 0$$

This is called an *integral quadratic constraint* (IQC) description of Δ , as the input/output pairs of Δ satisfy a family of quadratic, integral constraints. These different descriptions of model uncertainty are related. For example, if $w(t) = \delta z(t)$, with $w(t) \in \mathbf{R}^n$ and $z(t) \in \mathbf{R}^n$, and $\delta \in \mathbf{R}$, $|\delta| \leq 1$, then for any Hermitian-valued $X : \mathbf{R} \rightarrow \mathbf{C}^{n \times n}$ with $X(\omega) \geq 0$ for all $\omega \in \mathbf{R}$ and Skew-Hermitian $Y : \mathbf{R} \rightarrow \mathbf{C}^{n \times n}$,

$$\begin{aligned} \int_{-\infty}^{\infty} \begin{bmatrix} \hat{z}(j\omega) \\ \hat{w}(j\omega) \end{bmatrix}^* \begin{bmatrix} X(\omega) & Y(\omega) \\ Y^*(\omega) & -X(\omega) \end{bmatrix} \begin{bmatrix} \hat{z}(j\omega) \\ \hat{w}(j\omega) \end{bmatrix} d\omega \\ = \int_{-\infty}^{\infty} \hat{z}^*(j\omega) \left[(1 - \delta^2) X(\omega) \right] \hat{z}(j\omega) d\omega \end{aligned}$$

which is always ≥ 0 . Hence, the *uncertain parameter* can be recast as an operator satisfying an infinite family of IQCs. Nonlinear operators may also satisfy IQCs and it is common to “model” known nonlinear elements (e.g., saturation) by enumerating IQCs that they satisfy (Megretski and Rantzer 1997). An *uncertain dynamic model* is made up of an interconnection of these uncertain elements with a known (usually) linear system G .

Performance Metric

The main goal of robustness analysis is to assess the degrading effects of uncertainty. For this, a concrete notion of performance is needed, resulting in a mathematical/computational exercise to quantify the average or worst-case effects of the two types of uncertainty, signal and model, described earlier. In the robust control framework, adequate performance is

characterized in terms of the variability of possible behavior of particular signals. For instance, in the presence of reference inputs and disturbance inputs, as well as parameter uncertainty, it is required that tracking errors (e) and control inputs (u) remain small. A common measure of smallness is the $\mathcal{L}_2$ norm of signals. Typically, frequency-dependent weighting functions are used to preferentially weight one frequency range over another and/or to weight one signal relative to another. In this way, adequate performance can be defined as $\|W \begin{bmatrix} e \\ u \end{bmatrix}\|_2 \leq 1$, where W is a stable, linear system, called the “output” weighting function. Weighting functions are often used to transform a collection of performance objectives into a single norm bound objective in the robust control framework.

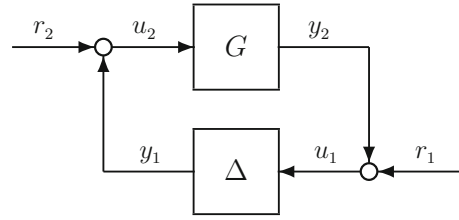
Robustness Analysis

Robustness analysis refers to the task of ascertaining the stability and/or performance characteristics of the uncertain system, given the limited knowledge about the uncertain information. The main result from Megretski and Rantzer (1997) concerns the stability of the interconnection shown in Fig. 1, where G is a known, stable, linear system and Δ is an operator that satisfies the IQC defined by Π . Under some important technical conditions, the theorem states “if there exists an $\epsilon > 0$ such that

$$\begin{bmatrix} G(j\omega) \\ I \end{bmatrix}^* \Pi(\omega) \begin{bmatrix} G(j\omega) \\ I \end{bmatrix} \leq -\epsilon I \quad (2)$$

for all $\omega \in \mathbf{R}$, then the interconnection is stable.” Stability here refers to finite $\mathcal{L}_2$ gain from inputs (r_1, r_2) to loop signals (u_1, u_2) .

Multiple IQCs satisfied by Δ can be incorporated into the analysis. In particular, assume that Δ satisfies the IQCs defined by the multipliers $\{\Pi_k\}_{k=1}^N$. Then Δ satisfies the IQC defined by any multiplier of the form $\Pi^\alpha := \sum_{k=1}^N \alpha_k \Pi_k$ where $\alpha_k \geq 0$. The stability test amounts to a semi-infinite, semidefinite feasibility problem: find nonnegative scalars $\{\alpha_k\}_{k=1}^N$ such that for some $\epsilon > 0$,



Robust Synthesis and Robustness Analysis Techniques and Tools, Fig. 1 Feedback interconnection for IQC stability test

$$\begin{bmatrix} G(j\omega) \\ I \end{bmatrix}^* \Pi^\alpha(\omega) \begin{bmatrix} G(j\omega) \\ I \end{bmatrix} \leq -\epsilon I \quad (3)$$

for all $\omega \in \mathbf{R}$. This infinite family of matrix inequalities (one for each frequency) can be equivalently expressed as a finite-dimensional linear matrix inequality (LMI) under some additional restrictions.

The structured singular value (μ) approach provides an alternative robust stability test in the case of only linear, time-invariant uncertainty (parametric or dynamic). Suppose Δ is drawn from a set of matrices, $\mathbf{\Delta} \subseteq \mathbf{C}^{m \times n}$ of the form

$$\mathbf{\Delta} = \left\{ \text{diag} \left[\delta_1^r I_{I_1}, \dots, \delta_V^r I_{I_V}, \delta_1^c I_{I_1}, \dots, \delta_S^c I_{I_S}, \Delta_1, \dots, \Delta_F \right] : \begin{array}{l} \delta_k^r \in \mathbf{R}, \delta_i^c \in \mathbf{C}, \Delta_j \in \mathbf{C}^{m_j \times n_j} \end{array} \right\}$$

The inclusion of complex-valued, uncertain matrices within $\mathbf{\Delta}$ may seem unusual and hard to motivate. However, in terms of their effect on stability, these are equivalent to the *uncertain linear dynamic element* introduced earlier in the *Uncertainty Modeling* section. This is discussed in more detail in the entry [► Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems](#).

Using the Nyquist stability criterion, the (G, Δ) interconnection is stable for all $\Delta \in \mathbf{\Delta}$, with $\bar{\sigma}(\Delta) < \beta$ if and only if G is stable, and

$$\det(I - G(j\omega)\Delta) \neq 0$$

for all $\Delta \in \mathbf{\Delta}$ with $\bar{\sigma}(\Delta) < \beta$ and all $\omega \in \mathbf{R}$ including $\omega = \infty$. The importance of the nonva-

nishing determinant condition warrants a definition of its own, the *structured singular value*. For a matrix $M \in \mathbb{C}^{n \times m}$, and Δ as given, define

$$\mu_{\Delta}(M) := \frac{1}{\min \{ \bar{\sigma}(\Delta) : \Delta \in \mathbf{\Delta}, \det(I - M\Delta) = 0 \}}$$

unless no $\Delta \in \mathbf{\Delta}$ makes $(I - M\Delta)$ singular, then $\mu_{\Delta}(M) := 0$. In this parlance, the (G, Δ) interconnection is stable for all $\Delta \in \mathbf{\Delta}$, with $\bar{\sigma}(\Delta) < \beta$ if and only if

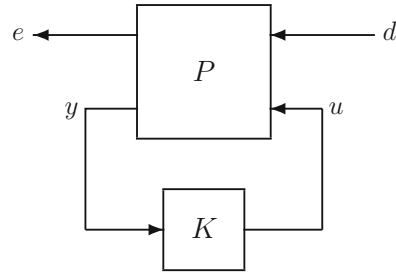
$$\mu_{\Delta}(G(j\omega)) \leq \frac{1}{\beta}$$

for all $\omega \in \mathbf{R}$ including $\omega = \infty$.

In summary, the structured singular value approach employs a Nyquist-based argument, resulting in a nonvanishing determinant condition, which must hold over all frequency and all possible frequency-response values of the uncertain elements. However, checking the nonvanishing determinant is difficult, and sufficient conditions, in the form of semidefinite programs (Doyle 1982; Fan et al. 1991) to ensure this are derived. This results in semidefinite feasibility problems which must hold at all frequencies. It is common to verify these only on a finite grid of frequencies, which is equivalent to ensuring that the closed-loop poles cannot migrate across the stability boundary at these frequencies. Semidefinite programs can be defined which carve out intervals around these fixed frequencies to completely guarantee stability.

Robust Synthesis

Synthesis refers to the mathematical design of the control law. The nominal synthesis problem (with no uncertainty) is formulated using the generic feedback structure shown in Fig. 2. The various signals in the diagram are the control inputs u , measurements y , exogenous disturbances d , and regulated variables e . P is a generalized plant that contains all information required to specify the synthesis problem. This includes the dynamics of the actual plant being controlled as well as any frequency domain weights that are used to



Robust Synthesis and Robustness Analysis Techniques and Tools, Fig. 2 Feedback interconnection for H_2 , H_∞ , and $\mathcal{L}_1$ optimal control

specify the performance objective. The objective of an optimal control problem is to synthesize a controller K that minimizes the closed-loop (e.g., H_2 , H_∞ , $\mathcal{L}_1$) norm from disturbances (d) to regulated variables (e), i.e., solve

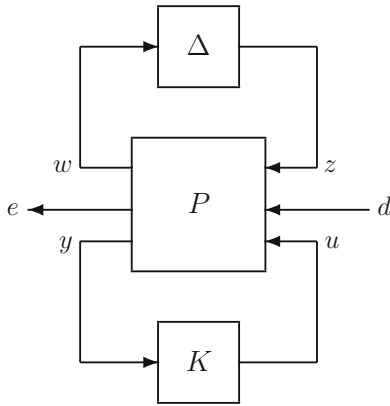
$$\min_{\text{allowable } K} \|F_L(P, K)\|$$

where $F_L(P, K)$ denotes the system obtained by closing the controller K around the lower loop of P . The H_2 , H_∞ , and $\mathcal{L}_1$ optimal control problems refer to the choice of the specific norm $\|F_L(P, K)\|$ used to specify the performance. A generalization of the H_∞ performance objective is simply to require that the closed-loop map from $d \rightarrow e$ satisfy an IQC defined by a given multiplier Π , called the performance multiplier, Apkarian and Noll (2006). The H_2 , H_∞ and $\mathcal{L}_1$ optimal control problems formulated as in Fig. 2 only involve signal uncertainty. In other words, these design problems do not explicitly account for the effects of model uncertainty.

Robust synthesis refers to control design that explicitly accounts for model uncertainty. It is usually formulated as a worst-case optimization, where the controller is chosen to minimize the worst-case effect of the signal and model uncertainty, loosely

$$\min_{\text{allowable } K} \max_{\text{allowable } d, \Delta} \|T(d, \Delta, K)\|$$

where d is a set of exogenous disturbances and Δ corresponds to the model uncertainty set. T represents the closed-loop relationship between



Robust Synthesis and Robustness Analysis Techniques and Tools, Fig. 3 Feedback interconnection for μ synthesis

d , Δ and the controller K . μ -synthesis is a specific technique developed to synthesize control algorithms which achieve robust performance, i.e., performance in the presence of signal and model uncertainty. The objective of μ -synthesis is to minimize over all stabilizing controllers K , the peak value of $\mu_{\Delta}(F_L(P, K))$ of the closed-loop transfer function defined by the interconnection in Fig. 3. P is the generalized plant model. The Δ block is the uncertain element from the set $\mathbf{\Delta}$, which parameterizes all of the assumed model uncertainty in the problem. The μ -synthesis optimization has high computational complexity (so-called NP-hard problem), though practical algorithms and software have been developed to design controllers using this control technique (Balas et al. 2013). Alternative robust synthesis approaches exist and often involve nonlinear optimization algorithms (Apkarian and Noll 2006). Drastic simplification regarding the models and uncertainty can be made resulting in problems that can be solved using LMI and semidefinite programming techniques (Boyd and Barrat 1991; Boyd et al. 1994).

Computational Tools

The MATLAB Robust Control Toolbox is a commercially available software product that

is part of the Mathworks control product line. It is tightly integrated with Control System Toolbox and Simulink products (Balas et al. 2013). The Robust Control Toolbox includes tools to analyze and design multi-input, multi-output control systems with uncertain elements. The primary building blocks, called uncertain elements or atoms, are uncertain real parameters and uncertain linear, time-invariant objects. These can be used to create coarse and simple or detailed and complex descriptions of model uncertainty. The uncertain object data structure eliminates the need to generate models of uncertainty and control analysis and design problem formulations, thereby allowing the practicing engineer to apply advanced robust control theory to their applications. Functions are available to analyze the robust stability, robust performance, and worst-case performance of uncertain multivariable system models using the structured singular value, μ . The Robust Control Toolbox also includes multivariable control synthesis tools to compute controllers that optimize worst-case performance and identify worst-case parameter values.

The IQC-Beta Toolbox is a publicly available robust analysis toolbox based on the IQC framework (Jönsson et al. 2004). A wide range of robust stability and performance analysis tests are available for uncertain, nonlinear, and time-varying systems. IQC-Beta is written in MATLAB and works seamlessly with the Control System Toolbox objects and basic interconnection functions. The Users manual nicely complements the literature on IQCs. The Computer Aided Control System Design package in Scilab, an open source numerical computation software, includes functionality for robustness analysis and the synthesis of robust control algorithms for multivariable systems (<http://www.scilab.org/>).

Conclusions

Robust control analysis and synthesis software tools are widely available and have been extensively used by industry since the late 1980s. The availability of software tools for robustness

analysis and synthesis played a major role in their wide and ubiquitous adoption in industry. They have been successfully applied to a variety of applications including aircraft flight control, launch vehicles, satellites, compact disk players, disk drives, backhoe excavators, nuclear power plants, helicopters, thin film extrusion, gas- and diesel-powered engines, missile autopilots, heating and ventilation systems, process control, and active suspension systems.

Cross-References

- [LMI Approach to Robust Control](#)
- [Optimization-Based Robust Control](#)
- [Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems](#)

Bibliography

- Apkarian P, Noll D (2006) IQC analysis and synthesis via nonsmooth optimization. *Syst Control Lett* 55:971–981
- Balas GJ, Packard AK, Chiang RC, Safonov M (2013) MATLAB robust control toolbox. The Mathworks Inc.
- Boyd S, Barrat C (1991) Linear controller design: limits of performance. Prentice Hall
- Boyd S, El Ghaoui L, Feron E, Balakrishnan V (1994) Linear matrix inequalities in system and control theory. SIAM, Philadelphia
- Desoer C, Vidyasagar M (2008) Feedback systems: input–output properties. *Classics in applied mathematics*. SIAM, Philadelphia
- Desoer C, Liu R, Murray J, Saeks R (1980) Feedback system design: a fractional representation approach to analysis and synthesis. *IEEE Trans Autom Control* 25(6):399–412
- Doyle J (1978) Guaranteed margins for LQG regulators. *IEEE Trans Autom Control* 23(4):756–757
- Doyle J (1982) Analysis of feedback systems with structured uncertainties. *IEE Proc Part D* 129(6):242–251
- Doyle J, Stein G (1981) Multivariable feedback design: concepts for a classical/modern synthesis. *IEEE Trans Autom Control* 26(1):4–16
- Doyle J, Glover K, Khargonekar P, Francis B (1989) State-space solutions to standard H_2 and H_∞ control problems. *IEEE Trans Autom Control* 34(8):831–847
- Dullerud G, Paganini F (2000) A course in robust control theory: a convex approach. Springer, New York
- Fan MKH, Tits AL, Doyle JC (1991) Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. *IEEE Trans Autom Control* 36(1):25–38
- Francis B (1987) A course in $\mathcal{H}_\infty$ control theory. Lecture notes in control and information sciences, vol 88. Springer, Berlin/New York
- Jönsson U, Kao CY, Megretski A, Rantzer A (2004) A guide to IQC β : a MATLAB toolbox for robust stability and performance analysis
- Megretski A, Rantzer A (1997) System analysis via integral quadratic constraints. *IEEE Trans Autom Control* 42(6):819–830
- Packard A, Doyle J (1993) The complex structured singular value. *Automatica* 29(1):71–109
- Safonov M (1982) Stability margins of diagonally perturbed multivariable feedback systems. *IEE Proc Part D* 129(6):251–256
- Skogestad S, Postlethwaite I (2005) Multivariable feedback control: analysis and design. Wiley, Hoboken
- Vidyasagar M (1985) Control system synthesis: a factorization approach. MIT, Cambridge
- Willems JC (1971) Least squares stationary optimal control and the algebraic Riccati equation. *IEEE Trans Autom Control* 16:621–634
- Zames G (1981) Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Trans Autom Control* 26(1):301–320
- Zhou K, Doyle J, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River

Robustness Analysis of Biological Models

Steffen Waldherr¹ and Frank Allgöwer²

¹Institute for Automation Engineering,
Otto-von-Guericke-Universität Magdeburg,
Magdeburg, Germany

²Institute for Systems Theory and Automatic
Control, University of Stuttgart, Stuttgart,
Germany

Abstract

Robustness analysis is the process of checking whether a system's function is maintained despite perturbations. Robustness analysis of biological models is typically applied to differential equation models of biochemical reaction networks. While robustness is primarily a yes-or-no question, for many applications in biological models, it is also desired to

compute a quantitative robustness measure. Such a measure is usually defined to be the maximum size of perturbations that the system can still tolerate. In addition, it is often of interest to specifically compute fragile perturbations, i.e., perturbations for which the system loses its function.

Keywords

Biochemical reaction networks · Fragile perturbations · Parametric uncertainty · Robustness measure · Structural uncertainty

Introduction

In biological systems analysis, robustness is the property that a system maintains its function in the face of internal or external perturbations (Kitano 2007). For a robustness analysis, one therefore needs to specify the system to be analyzed, the function that should be maintained, and the perturbation class.

The models to which robustness analysis is applied are mostly differential equation models of biochemical reaction networks. They are generally written as

$$\dot{x} = Sv(x), \quad (1)$$

where $x \in \mathbb{R}^n$ is the vector of intracellular concentrations; $S \in \mathbb{R}^{n \times m}$ is the stoichiometric matrix, containing the information how the individual network components participate in the reactions; and $v(x) \in \mathbb{R}^m$ is the reaction rate vector, in most cases a nonlinear function of the concentrations x .

The biological functions that are being studied by robustness analysis are very broad, pertaining to the wide range of biological functions implemented by biochemical reaction networks. Specific problems being considered are:

1. The occurrence of qualitative dynamical patterns such as sustained oscillations or multistability, where the system converges to one of multiple stable steady states depending on

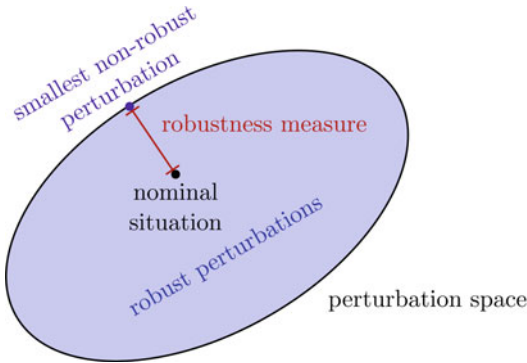
initial conditions or external stimuli (Ma and Iglesias 2002; Eissing et al. 2005).

2. The steady-state concentration value for a subset of the biochemical network's components (Shinar and Feinberg 2010; Steuer et al. 2011).
3. Quantitative measures derived from the network's dynamics, for example, the period of sustained oscillations (Stelling et al. 2004).

For the perturbation classes, two approaches can be distinguished. In parametric robustness analysis, a parametrized biological model is given, and the perturbation consists in varying the values of the parameters away from their nominal value. In structural robustness analysis, perturbations to the interaction structure of the network or the functional form of the reaction rate functions $v(x)$ are considered. Robustness analysis with these perturbation classes is presented in more detail below.

The perturbation class is also relevant for two applications of robustness analysis which go beyond simply deciding whether a system is robust or not. First, it is often of interest to get a better quantification of robustness than a binary decision. Then, it is common to define a robustness measure, which usually quantifies how large perturbations can be without affecting the system's function (Morohashi et al. 2002; Ma and Iglesias 2002). Such a measure requires an appropriate definition of the perturbation size. With parametric perturbations, norms in parameter space are often useful (Ma and Iglesias 2002; Waldherr and Allgöwer 2011). With structural perturbations, the proximity of interaction functions in function space (Breindl et al. 2011) or the number of changes in the interaction structure can be evaluated.

Second, one often desires to compute specific non-robust perturbations, i.e., perturbations within the given class for which the system loses the considered functionality. There is a close relation between non-robust perturbations and the robustness measure, in that the norm of the smallest non-robust perturbation is equal to the robustness measure. Yet, it is often easier to compute a robustness measure than a non-robust perturbation. Especially algorithms that give a



Robustness Analysis of Biological Models, Fig. 1
Illustration of key characteristics in robustness analysis

lower bound on the robustness measure will usually not provide a non-robust perturbation.

An illustration of the key characteristics in robustness analysis is shown in Fig. 1.

This also illustrates that any norm-based robustness measure depends on the nominal situation, where no perturbation is present.

When performing robustness analysis on a mathematical model of the considered system, the potential mismatch between model and system has to be kept in mind. By comparing the mathematical analysis results to experimental observations, robustness analysis methods are also useful for the validation or invalidation of biological network models (Bates and Cosentino 2011).

Robustness Analysis with Parametric Perturbations

Robustness analysis with parametric perturbations is applied to parametrized differential equation models of biochemical reaction networks, which are described by an equation of the form

$$\dot{x} = Sv(x, \mu), \quad (2)$$

where $\mu \in \mathbb{R}^p$ is a vector of parameters. Such parameters may, for example, represent the total expression level of proteins involved in the reaction network, where usually a large variability

due to the stochastic process of gene expression occurs.

This entry focuses on two specific system functionalities for robustness with respect to parametric perturbations, the qualitative dynamical behavior, and the steady-state level of a subset of the network's components. These are particularly relevant for biological models: the dynamical behavior often represents qualitative biological regulatory mechanisms, whereas the steady-state level of network components with a downstream regulatory effect is important for the stimulus-response relation of a biological network.

The Qualitative Dynamical Behavior

Considering the qualitative dynamical behavior, it is of interest to distinguish situations of a globally stable equilibrium point, multiple locally stable equilibrium points, or sustained oscillations due to a limit cycle or more complex attractors. Since changes in these dynamical patterns correspond to the occurrence of bifurcations in the model dynamics (2), this type of robustness analysis is closely related to bifurcation analysis. In the case of scalar, positive parameters μ , a corresponding robustness measure DOR has been defined by Ma and Iglesias (2002) as

$$DOR = 1 - \max \left\{ \frac{\check{\mu}}{\mu_0}, \frac{\mu_0}{\hat{\mu}}, 0 \right\}, \quad (3)$$

where $\check{\mu}$ and $\hat{\mu}$ are the closest bifurcation points smaller and larger than μ , respectively. The robustness measure DOR is between 0 and 1 and indicates how much the parameter can be varied before reaching a bifurcation: for any multiplicative perturbation of less than $(1 - DOR)^{-1}$, no bifurcation will occur. A generalization to multiparametric models has been proposed in Waldherr and Allgöwer (2011): their robustness measure ϱ is defined as

$$\varrho = \sup \{ \varrho \geq 1 \mid \text{no bifurcation occurs in the hyperrectangle } [\varrho^{-1}\mu_0, \varrho\mu_0] \}. \quad (4)$$

The measure ϱ directly gives the multiplicative parameter variation up to which no bifurcation occurs.

In general, the information required for a bifurcation-based robustness measure will only be available from a complete bifurcation analysis of the model. When restricting the types of bifurcations that are considered to bifurcations of equilibrium point, one can however check robustness by studying linear approximations at the system's equilibrium points. Since the reaction rates $v(x, \mu)$ are usually modeled as polynomial or rational functions, polynomial programming methods can be applied to compute a robustness measure (Waldherr and Allgöwer 2011) in this case.

The Steady-State Output Concentration

In biochemical network analysis, mostly linear outputs of the form

$$y = Cx, \quad (5)$$

with $C \in \mathbb{R}^{q \times n}$ are considered. A common special case is that the rows of C are a subset of the rows of the identity matrix in $\mathbb{R}^n$, i.e.,

$$C = (e_i^T)_{i \in \mathcal{I}_y}, \quad (6)$$

and $\mathcal{I}_y \subset \{1, 2, \dots, n\}$ is the index set defining the output concentrations.

A biochemical network has a robust steady-state output concentration, if the steady-state output $\bar{y}$ is independent of the parameters μ (Steuer et al. 2011). For a steady-state map $\bar{y} = h(\mu)$, this corresponds to the condition

$$h'(\mu) = 0. \quad (7)$$

For the special case of an output given by (6), a sufficient and necessary condition for steady-state output robustness has been discovered by Steuer et al. (2011). The condition amounts to checking that a vector P , which describes the perturbation of the reaction rates under parameter variations, is in a subspace $\mathcal{I} = \text{im } M + \ker S \text{diag}(\alpha)$ for any α in the kernel of S , where M is a matrix composed

of the normalized derivatives of the reaction rates with respect to the concentrations which do not appear in the output. A notable underlying assumption here is that the network's steady state does not undergo any local bifurcations within the considered parameter region, which directly relates back to the robustness analysis discussed in the previous section.

For the special case where parameters are the concentrations of conserved chemical species, a sufficient condition for steady-state output robustness has also been discovered by Shinar and Feinberg (2010). They propose the term *absolute concentration robustness* for this property. Here, the assumption that no local bifurcations occur within the considered parameter region is not required a priori but rather is also a consequence of the proposed condition.

Robustness Analysis with Structural Perturbations

Robustness analysis with parametric perturbations is based on the assumption that the reaction rate expressions are exact and that all perturbations are captured by parameter variations. This assumption can hardly be justified for many practical models, and an analysis with structural perturbations becomes necessary. Such analyses have discovered models which are very robust against parametric perturbations but non-robust against structural perturbations (Jacobsen and Cedersund 2008).

The biological functions for which rigorous results on structural robustness are available are again related to the nonoccurrence of bifurcations in the model. For the restriction to local bifurcations of equilibria, linear systems theory offers efficient analysis tools for structural robustness.

In a first step, a structural perturbation of the network's interaction graph was suggested (Jacobsen and Cedersund 2008). This approach considers the network's Jacobian

$$A = S \frac{\partial v}{\partial x}(\bar{x}) \quad (8)$$

evaluated at a steady state $\bar{x}$. The Jacobian is then perturbed to

$$\tilde{A} = \text{diag } A + (A - \text{diag } A)(I + \Delta), \quad (9)$$

where $\text{diag } A$ is the diagonal of A and Δ is a perturbation matrix, containing uncertain time-invariant linear systems as elements.

As an alternative approach, Waldherr et al. (2009) have suggested a structural perturbation of the reaction rate expressions. Thereby, the network's Jacobian is perturbed to

$$\tilde{A} = S \left(\frac{\partial v}{\partial x}(\bar{x}) + \Delta \right). \quad (10)$$

In the case of real Δ , this perturbation simply corresponds to a change in the reaction rate slopes at steady state.

With both approaches, robustness analysis with structured singular values can be applied to test for changes in the local dynamics at the considered equilibrium point. This allows to evaluate a model's robustness against this type of structural perturbations and also yields non-robust perturbations.

Summary and Future Directions

Robustness analysis of biological models is well established in biological network theory. Mathematical methods rooted in systems and control are particularly beneficial for approaching this task.

While this entry focuses on models of biochemical reaction networks given by differential equations, the robustness analysis problem has also been studied in other model frameworks, for example, discrete dynamical models (Chaves et al. 2006). Yet, beyond simulation-based studies, robustness analysis is still an open problem in many practically relevant biological model classes. This concerns, for example, stochastic models or models on the cell population level.

In a similar manner, it will be important to extend the perturbation classes that are being considered and to include, for example, time-varying or other perturbations that are relevant for biological models. Concerning the biological function, most robustness analysis methods focus on the steady-state behavior. In the future, it

will be of interest to also take, for example, the transient dynamics into account.

In linear systems theory, the concept of robust performance is well established. While efforts have been made to transfer that concept to biochemical networks (Doyle and Stelling 2005), it remains difficult to quantify performance of such networks, thus impeding the development of stringent robustness analysis tools. One of the reasons for this difficulty is certainly that biological performance is more naturally defined in the time domain than in the frequency domain, which narrows the conclusions that could be drawn from a direct application of classical robust performance analysis methods.

Cross-References

- [Computational Complexity in Robustness Analysis and Design](#)
- [Deterministic Description of Biochemical Networks](#)
- [Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems](#)

Bibliography

- Bates D, Cosentino C (2011) Validation and invalidation of systems biology models using robustness analysis. *IET Syst Biol* 5(4):229–244
- Breindl C, Waldherr S, Wittmann DM, Theis FJ, Allgöwer F (2011) Steady-state robustness of qualitative gene regulation networks. *Int J Robust Nonlinear Control* 21(15):1742–1758. doi:10.1002/rnc.1786, <http://dx.doi.org/10.1002/rnc.1786>
- Chaves M, Sontag ED, Albert R (2006) Methods of robustness analysis for Boolean models of gene control networks. *IEE Proc Syst Biol* 153(4):154–167. doi:10.1049/ip-syb:20050079, <http://dx.doi.org/10.1049/ip-syb:20050079>
- Doyle FJ, Stelling J (2005) Robust performance in biochemical networks. In: *Proceedings of the 16th IFAC World Congress, Prague*
- Eissing T, Allgöwer F, Bullinger E (2005) Robustness properties of apoptosis models with respect to parameter variations and intrinsic noise. *IEE Proc Syst Biol* 152(4):221–228. doi:10.1049/ip-syb:20050046
- Jacobsen EW, Cedersund G (2008) Structural robustness of biochemical network models—with application to the oscillatory metabolism of activated neu-

- trophils. *IET Syst Biol* 2(1):39–47. <http://link.aip.org/link/?SYB/2/39/1>
- Kitano H (2007) Towards a theory of biological robustness. *Mol Syst Biol* 3:137. doi:10.1038/msb4100179, <http://dx.doi.org/10.1038/msb4100179>
- Ma L, Iglesias PA (2002) Quantifying robustness of biochemical network models. *BMC Bioinform* 3:38
- Morohashi M, Winn AE, Borisuk MT, Bolouri H, Doyle J, Kitano H (2002) Robustness as a measure of plausibility in models of biochemical networks. *J Theor Biol* 216(1):19–30. doi:10.1006/jtbi.2002.2537, <http://dx.doi.org/10.1006/jtbi.2002.2537>
- Shinar G, Feinberg M (2010) Structural sources of robustness in biochemical reaction networks. *Science* 327(5971):1389–1391. doi:10.1126/science.1183372, <http://dx.doi.org/10.1126/science.1183372>
- Stelling J, Gilles ED, Doyle III FJ (2004) Robustness properties of circadian clock architectures. *Proc Natl Acad Sci* 101(36):13210–13215. doi:10.1073/pnas.0401463101, <http://dx.doi.org/10.1073/pnas.0401463101>
- Steuer R, Waldherr S, Sourjik V, Kollmann M (2011) Robust signal processing in living cells. *PLoS Comput Biol* 7(11):e1002218. <http://dx.doi.org/10.1371/journal.pcbi.1002218>
- Waldherr S, Allgöwer F (2011) Robust stability and instability of biochemical networks with parametric uncertainty. *Automatica* 47:1139–1146. doi:10.1016/j.automatica.2011.01.012, <http://dx.doi.org/10.1016/j.automatica.2011.01.012>
- Waldherr S, Allgöwer F, Jacobsen EW (2009) Kinetic perturbations as robustness analysis tool for biochemical reaction networks. In: *Proceedings of the 48th IEEE Conference on Decision and Control, Shanghai*, pp 4572–4577. doi:10.1109/CDC.2009.5400939, <http://dx.doi.org/10.1109/CDC.2009.5400939>

Robustness Issues in Quantum Control

Ian R. Petersen

Research School of Electrical, Energy and Materials Engineering, Australian National University, Canberra, ACT, Australia

Abstract

Robust quantum control theory is concerned with the design of controllers for quantum

systems taking into account uncertainty in the model of the system. The robust open-loop control of quantum systems is discussed in this article. Also discussed is the robust stability analysis problem for quantum systems and two forms of quantum small gain theorem are presented. In addition, the article discusses the design of robust quantum feedback control systems.

Keywords

Quantum control · Robustness · Robust stability · Minimax control · Ensemble controllability · H^∞ control

Introduction

The control of systems whose dynamics are governed by the laws of quantum mechanics is the subject of quantum control theory. As in the case of classical control theory, the models used in quantum control are often subject to uncertainties. This motivates the study of robust quantum control, in which the quantum systems to be controlled are modelled as uncertain quantum systems; e.g., see Mabuchi and Khaneja (2005). A related problem, is the problem of robust estimation and filtering for uncertain quantum systems; e.g., see Yamamoto and Bouten (2009), Petersen (2017), and Xiang et al. (2017a, b). The issue of robust stability is particularly important in the case of quantum feedback control since, in this case, there is always the possibility of instability. An important area of quantum control theory is open-loop quantum control. Since uncertainties arise in the quantum system models being considered, the robustness of open-loop quantum control systems is also important; e.g., see Li and Khaneja (2009), Rabitz (2002), Owrutsky and Khaneja (2012).

This article surveys some of the important research results on robust quantum control which have arisen in various application areas. These include some recent results on robust open-loop control of quantum systems; see Zhang and Rabitz (1994). Also considered are some recent

robust stability analysis results for uncertain quantum systems, which amount to quantum versions of the classical small gain theorem; see Petersen et al. (2012). Finally, the article looks at robust quantum feedback controller design; see James et al. (2008) and Dong et al. (2009).

Robust Open-Loop Control of Quantum Systems

In the robust open-loop control of quantum systems, the quantum system is modeled in the Schrödinger picture. The models can be given either in terms of the Schrödinger equation for the system state $|\psi(t)\rangle$:

$$i \frac{\partial}{\partial t} |\psi(t)\rangle = \left[H_0 + \sum_{k=1}^m u_k(t) H_k \right] |\psi(t)\rangle \quad (1)$$

or the master equation for the system density operator ρ :

$$\dot{\rho}(t) = -i \left[\left(H_0 + \sum_{k=1}^m u_k(t) H_k \right), \rho(t) \right] \quad (2)$$

e.g., see Nurdin and Yamamoto (2017). In these equations, H_0 is the free Hamiltonian of the system and H_k are corresponding control Hamiltonians. Also, $i = \sqrt{-1}$ and m is the number of control inputs. In the robust open-loop control of quantum systems, these quantities are assumed to be uncertain, and the control law $u_k(t)$ is to be designed to guarantee an adequate level of performance for all possible values of the uncertainties. Here, performance is measured in terms of the fidelity between the actual final state or density matrix of the system and the desired final state or density matrix; e.g., see Nielsen and Chuang (2000).

In the minimax optimal control approach to robust open-loop control of quantum systems, the uncertainties in the Hamiltonian are represented in terms of a vector quantity w which is subject to constraints. Then, the robust control problem

is the minimax optimal control problem

$$\min_u \max_w J(u, w)$$

where $J(u, w)$ is a suitable cost function and the problem is subject to the constraints defined by the system dynamics (1) and the constraints on the uncertainty w ; see Zhang and Rabitz (1994). Some standard numerical procedures have been proposed to solve this minimax optimal control problem with applications in chemical physics; see Zhang and Rabitz (1994).

Related to the robust open-loop control of quantum systems is the control of inhomogeneous quantum ensembles. In this problem, the same control signal $u_k(t)$ is applied to a large number of quantum particles in an ensemble. Also, the Hamiltonians corresponding to individual particles may have different parameter values, and so this problem is equivalent to a robust open-loop quantum control problem; e.g., see Li and Khaneja (2006). In studying this problem, the issue of controllability has been considered (see e.g., Li and Khaneja 2009) as in the standard open-loop quantum control problem. Also, numerical methods have been proposed for constructing an optimal control law for inhomogeneous ensembles; e.g., see Ruths and Li (2012); Owrutsky and Khaneja (2012). This approach has arisen in applications to chemical physics.

Robustness Analysis for Uncertain Quantum Systems

The problem of robust stability analysis for uncertain quantum systems was considered in the paper D'Helon and James (2006) which was concerned with the feedback interconnection of two quantum optical systems as shown in Fig. 1. In this interconnection, each of the quantum systems is a linear quantum optical system described in the Heisenberg picture by linear quantum stochastic differential equations (QSDEs) of the form

$$dx(t) = Ax(t)dt + Bdu(t);$$

$$dy(t) = Cx(t)dt + Ddu(t); \quad (3)$$

see James et al. (2008) for more details on this class of quantum system models. Here, $x(t)$ is a vector system variables which are operators on the underlying Hilbert space of the system. Also, the input and output fields are decomposed as $du(t) = \beta_u(t)dt + d\tilde{u}(t)$ and $dy(t) = \beta_y(t)dt + d\tilde{y}(t)$ where $\beta_u(t)$, $\beta_y(t)$ denote the signal parts of the quantities $du(t)$, $dy(t)$ respectively. Furthermore, $d\tilde{u}(t)$, $d\tilde{y}(t)$ denote the noise parts of the quantities $du(t)$, $dy(t)$ respectively; e.g., see James et al. (2008). Such a system is stable and has a finite gain $g > 0$ if there exist constants $\mu > 0$ and $\lambda > 0$ such that

$$\int_0^t \langle \|\beta_y(\tau)\|^2 \rangle d\tau \leq \mu + \lambda t \\ + \int_0^t \langle \|\beta_u(\tau)\|^2 \rangle d\tau \quad \forall t > 0;$$

e.g., see D'Helon and James (2006) and James and Gough (2010). Here $\langle \cdot \rangle$ denotes quantum expectation.

The two quantum optical systems shown in Fig. 1 are interconnected via beam splitters which are described by equations

$$\begin{aligned} u_1 &= \epsilon_1 w_1 - \sqrt{1 - \epsilon_1^2} y_2; \\ z_1 &= \sqrt{1 - \epsilon_1^2} w_1 + \epsilon_1 y_2; \\ u_2 &= \epsilon_2 w_2 - \sqrt{1 - \epsilon_2^2} y_1; \\ z_2 &= \sqrt{1 - \epsilon_2^2} w_2 + \epsilon_2 y_1 \end{aligned}$$

where $\epsilon_1 \in (0, 1)$ and $\epsilon_2 \in (0, 1)$ are given constants. The quantum small gain theorem established in D'Helon and James (2006) shows that if both of the quantum systems in Fig. 1 are stable and have finite gains $g_1 > 0$ and $g_2 > 0$, respectively, such that $\sqrt{1 - \epsilon_1^2} \sqrt{1 - \epsilon_2^2} g_1 g_2 < 1$, then the feedback interconnected system will also be stable and have a finite gain. This result can be thought of as a stability robustness result if the first quantum system is regarded as the nominal quantum system and the second quantum system

is regarded as being the uncertain part of the system subject to the given finite gain constraint.

An alternative approach to the robust stability analysis of uncertain quantum systems considers an uncertain quantum system described using the (S, L, H) description (see Gough and James 2009 for more details on this class of quantum systems). Here, the system Hamiltonian is described in terms of vectors of annihilation and creation operators a and $a^\#$, respectively, as

$$H = \frac{1}{2} \begin{bmatrix} a^\dagger & a^T \end{bmatrix} M \begin{bmatrix} a \\ a^\# \end{bmatrix} + \frac{1}{2} \tilde{\zeta}^\dagger \Delta \tilde{\zeta}$$

where $a^\dagger = (a^\#)^T$, M is a known complex Hermitian matrix describing the nominal Hamiltonian, and Δ is a complex Hermitian uncertainty matrix subject to the norm bound $\|\Delta\| \leq \frac{2}{\gamma}$,

and $\tilde{\zeta} = E \begin{bmatrix} a \\ a^\# \end{bmatrix}$, where E is a known complex matrix describing the uncertainty structure. Here $\gamma > 0$ is a given constant. Furthermore, it is assumed that the scattering matrix $S = I$ and the coupling operator vector L is such that $\begin{bmatrix} L \\ L^\# \end{bmatrix} = N \begin{bmatrix} a \\ a^\# \end{bmatrix}$ where N is a known complex matrix. This uncertain quantum system is robustly mean square stable if the H^∞ norm bound condition

$$\left\| E \left(sI + iJM + \frac{1}{2} JN^\dagger JN \right)^{-1} J E^\dagger \right\|_\infty < \frac{\gamma}{2}$$

is satisfied where $J = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$; see Petersen et al. (2012).

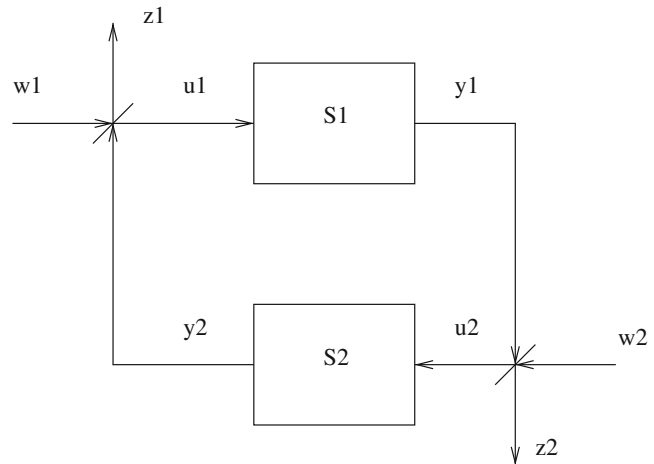
Robust Feedback Control of Quantum Systems

Schrödinger Picture Approaches to Robust Measurement-Based Quantum Feedback Control

A number of results have appeared which use Schrödinger picture models in robust measurement based quantum feedback control. These

Robustness Issues in Quantum Control, Fig. 1

Feedback interconnection of two quantum optical systems



results are based on uncertain quantum system models of the form (1) or (2) and extend the results mentioned above by allowing for measurements of the quantum system in order to achieve improved robustness against uncertainties in the system Hamiltonian. For example, consider a quantum system of the form (1) with uncertainties in the system Hamiltonian. Then a measurement feedback robust control scheme can be constructed which involves periodic projective measurements on the system. In a projective measurement of the quantum system (1), the state $|\psi(t)\rangle$ collapses to an eigenstate of H_0 corresponding to the measurement outcome obtained. The sliding mode control algorithm uses open-loop time optimal control to steer the state of the system back to a specified eigenstate of the system whenever a measurement is obtained which does not correspond to this desired eigenstate; see Dong and Petersen (2009). This desired eigenstate is referred to as the sliding mode domain and the state of the system is guaranteed to stay within the sliding mode domain with a specified probability provided that the measurement sampling period in the proposed feedback control algorithm is chosen to be sufficiently fast; see Dong and Petersen (2009). In the case of two-level quantum systems, this sliding mode control approach is implemented using a Lyapunov method for open-loop quantum control to steer the system back to the sliding mode domain; see Dong and Petersen (2012). In all of these cases,

robustness is ensured by including uncertainty in the underlying quantum system models and then taking this into account in the design of the control laws and sampling period.

Another approach to the measurement-based robust quantum feedback control problem involves an extension of the robust open-loop control results considered in the section “[Robust Open-Loop Control of Quantum Systems](#).” In this approach, robust open-loop control results are extended to solve the problem of stabilization of an ensemble of quantum particles; see Beauchard et al. (2012).

Heisenberg Picture Approaches to Robust Quantum Feedback Control

Consider a quantum linear system modeled in the Heisenberg picture by quantum stochastic differential equations (QSDEs) as follows

$$\begin{aligned} dx(t) &= Ax(t)dt + Bd w(t); \\ dy(t) &= Cx(t)dt + Dd w(t); \end{aligned} \quad (4)$$

see Nurdin and Yamamoto (2017) for details on this class of quantum system models which arises in the area of quantum optics. In the robust quantum feedback control problem, the matrices A , B , C , D may be uncertain, and a feedback controller can be designed using the quantum H^∞ control approach to ensure that the resulting closed loop system is robustly stable; see James

et al. (2008). In the case of measurement-based feedback control, the controller is a classical system described by linear stochastic differential equations of the form

$$\begin{aligned} dx_K(t) &= A_K x_k(t)dt + B_K dy(t) \\ \beta_u(t)dt &= C_K x_k(t)dt; \end{aligned} \quad (5)$$

see Nurdin and Yamamoto (2017). In the case of coherent feedback control, the controller is another quantum linear system described by QSDEs of the form

$$\begin{aligned} dx_K(t) &= A_K x_k(t)dt + B_K dy(t) + \bar{B}_K d\bar{w}_K(t) \\ dy_K(t) &= C_K x_k(t)dt + \bar{D}_K d\bar{w}_K(t); \end{aligned} \quad (6)$$

see Nurdin and Yamamoto (2017).

In this approach to robust quantum feedback control, the uncertainty in the quantum system being controlled is represented by uncertainty in the matrix A as $A = \hat{A} + \bar{B}\Delta\bar{C}$ where Δ is a constant but unknown uncertain matrix satisfying the bound $\Delta^T \Delta \leq I$. The controller, which may be either a classical controller or a coherent controller, is designed using the quantum H^∞ approach. Then the resulting closed loop system will be robustly stable; see James et al. (2008). Similarly, in the case of uncertainty in the plant Hamiltonian matrix such as considered in the section “[Robustness Analysis for Uncertain Quantum Systems](#)” or uncertainty in the form of an uncertain subsystem connected optically to the plant in feedback, also as considered in the section “[Robustness Analysis for Uncertain Quantum Systems](#),” then the quantum H^∞ approach combined with the robust stability analysis results of the section “[Robustness Analysis for Uncertain Quantum Systems](#)” shows that the quantum H^∞ method can also be used to design robustly stabilizing controllers in these cases.

Summary and Future Directions

To date there have been only a few papers published in the general area of robust quantum control. The results which were considered in

this article covered open-loop and feedback quantum control problems along with stability robustness analysis problems. A common theme in the results which were considered is that they were based on uncertain quantum mechanical models. It is expected that future research in this area will intensify as the use of feedback control becomes more prevalent in areas of experimental quantum technology.

Cross-References

- [Application of Systems and Control Theory to Quantum Engineering](#)
- [Control of Quantum Systems](#)
- [Quantum Networks](#)

Bibliography

- Beauchard K, da Silva PSP, Rouchon P (2012) Stabilization for an ensemble of half-spin systems. *Automatica* 48(1):68–76
- D’Helon C, James M (2006) Stability, gain, and robustness in quantum feedback networks. *Phys Rev A* 73:053803
- Dong D, Petersen IR (2009) Sliding mode control of quantum systems. *New J Phys* 11:105033. arXiv:0911.0062
- Dong D, Petersen IR (2012) Sliding mode control of two-level quantum systems. *Automatica* 48(5):725–735. arXiv:1009.0558
- Dong D, Lam J, Petersen IR (2009) Robust incoherent control of qubit systems via switching and optimization. *Int J Control* 83(1):206–217
- Gough J, James MR (2009) The series product and its application to quantum feedforward and feedback networks. *IEEE Trans Autom Control* 54(11):2530–2544
- James M, Gough J (2010) Quantum dissipative systems and feedback control design by interconnection. *IEEE Trans Autom Control* 55(8):1806–1821
- James MR, Nurdin HI, Petersen IR (2008) H^∞ control of linear quantum stochastic systems. *IEEE Trans Autom Control* 53(8):1787–1803
- Li JS, Khaneja N (2006) Control of inhomogeneous quantum ensembles. *Phys Rev A* 73:030302
- Li JS, Khaneja N (2009) Ensemble control of Bloch equations. *IEEE Trans Autom Control* 54(3):528–536
- Mabuchi H, Khaneja N (2005) Principles and applications of control in quantum systems. *Int J Robust Nonlinear Control* 15:647–667
- Nielsen M, Chuang I (2000) Quantum computation and quantum information. Cambridge University Press, Cambridge, UK

- Nurdin HI, Yamamoto N (2017) Linear dynamical quantum systems: analysis, synthesis, and control. Springer, Berlin
- Owrutsky P, Khaneja N (2012) Control of inhomogeneous ensembles on the Bloch sphere. *Phys Rev A* 86:022315
- Petersen IR (2017) Quantum Popov robust stability analysis of an optical cavity containing a saturated Kerr medium. *Quantum Science and Technology* 2(3):034,009
- Petersen IR, Ugrinovskii V, James MR (2012) Robust stability of uncertain linear quantum systems. *Philos Trans R Soc A* 370(1799):5354–5363
- Rabitz H (2002) Optimal control of quantum systems: Origins of inherent robustness to control field fluctuations. *Phys Rev A* 66:063405
- Ruths J, Li JS (2012) Optimal control of inhomogeneous ensembles. *IEEE Trans Autom Control* 57(8):2021–2032
- Xiang C, Petersen IR, Dong D (2017a) Coherent robust H-infinity control of linear quantum systems with uncertainties in the Hamiltonian and coupling operators. *Automatica* 81:8–21
- Xiang C, Petersen IR, Dong D (2017b) Performance analysis and coherent guaranteed cost control for uncertain quantum systems using small gain and Popov methods. *IEEE Trans Autom Control* 62(3):1524–1529
- Yamamoto N, Bouten L (2009) Quantum risk-sensitive estimation and robustness. *IEEE Trans Autom Control* 54(1):92–107
- Zhang H, Rabitz H (1994) Robust optimal control of quantum molecular systems in the presence of disturbances and uncertainties. *Phys Rev A* 49:2241–2254

Routing in Transportation Networks

Ketan Savla
Sonny Astani Department of Civil and
Environmental Engineering, University of
Southern California, Los Angeles, CA, USA

Abstract

Routing is an effective way to control traffic to optimize network-level objective. The effectiveness depends in part on complex traffic flow dynamics and driver preferences. We

provide an overview of the basics of routing design under such considerations and point to promising research directions.

Keywords

Traffic flow dynamics over network · Robustness · Routing under information constraint · Social optimality · User equilibrium

Introduction

We start with a simple setting, through which we also describe preliminaries. The directed graph structure of a transportation network is described by the tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, c)$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of directed edges, and c is the set of edge-wise flow capacities. Two specific nodes $o \in \mathcal{V}$ and $d \in \mathcal{V}$ are, respectively, designated to be the origin and destination nodes. Without loss of generality, we let o (resp., d) have no incoming (resp., outgoing) edge and only one outgoing (resp., incoming) edge, say i_{in} (resp., i_{out}), with unbounded capacity $c_{i_{\text{in}}} = \infty$ (resp., $c_{i_{\text{out}}} = \infty$); see Fig. 1a for an illustration. $\mathcal{E}_v^+$ (resp., $\mathcal{E}_v^-$) denotes the set of edges outgoing from (resp., incoming to) node v . The head node of $i \in \mathcal{E}$ is denoted as τ_i . Each edge $i \in \mathcal{E}$ is associated with a flow variable $f_i \geq 0$. At times, it will be convenient to refer to the set $\hat{\mathcal{E}} := \mathcal{E} \setminus \{i_{\text{in}}, i_{\text{out}}\}$ of edges other than the ones associated with the origin and destination nodes.

Routing is an assignment of flow $\{f_i\}_{i \in \mathcal{E}}$ to edges of the transportation network subject to certain constraints. A simple form of such an assignment is solution to the following linear program:

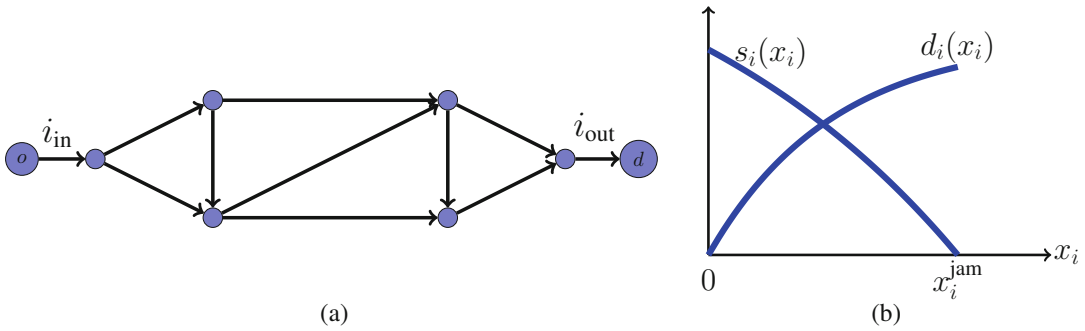
$$\min_{f_i, i \in \mathcal{E}} \sum_{i \in \mathcal{E}} g_i f_i \quad (1a)$$

$$\sum_{i \in \mathcal{E}_v^-} f_i = \sum_{i \in \mathcal{E}_v^+} f_i, \quad v \in \mathcal{V} \setminus \{o, d\} \quad (1b)$$

$$f_i = \lambda, \quad i \in \{i_{\text{in}}, i_{\text{out}}\} \quad (1c)$$

$$0 \leq f_i \leq c_i, \quad i \in \mathcal{E} \quad (1d)$$

This work was supported in part by NSF CAREER ECCS # 1454729. K. Savla has financial interest in Xtelligent, Inc.



Routing in Transportation Networks, Fig. 1 Illustration of (a) transportation network topology; (b) demand and supply functions on a sample edge i

where $\{g_i\}_{i \in \mathcal{E}}$ are cost coefficients in (1a); (1b) represents flow conservation at nodes; (1c) captures the requirement that a constant rate of traffic $\lambda > 0$ needs to be routed from o to d ; and (1d) captures non-negativity and edge-wise capacity constraints on flow. A necessary and sufficient condition for (1) to admit a feasible solution is that λ is less than or equal to the minimum among capacities of all $o - d$ cuts in the networks. An $o - d$ cut is a subset of edges $\tilde{\mathcal{E}} \subset \mathcal{E}$, such that $o \in \tilde{\mathcal{E}}$ and $d \notin \tilde{\mathcal{E}}$. The capacity of cut $\tilde{\mathcal{E}}$ is the sum of capacities of all edges outgoing from $\tilde{\mathcal{E}}$.

The notion of *residual* capacity plays a key role in tight characterization of robustness in terms of loss in capacities under which routing is feasible. For instance, if the loss in capacities from c to $\tilde{c} \leq c$ is such that $\|c - \tilde{c}\|_1$ is no greater than the *network residual capacity*, then routing under (1) with new capacities $\tilde{c}$ is feasible. Network residual capacity is the difference between its min-cut capacity and λ .

The framework in (1) is for socially optimal routing in static setting (Cascetta 2009). Extensions to traffic flow dynamics are discussed in section “Routing Dynamical Traffic Flow over Network”, and extensions to settings which take into account driver preferences are discussed in section “Routing Under Driver Preferences”. The setup outlined in this section is for single-commodity flow, i.e., when there is no constraint on which destination node the flow needs to be routed to. The single origin-destination setting in (1) is the simplest such setting. Extension to multiple origins and destinations for single-commodity flow is straightforward. The

discussion in section “Fixed Routing” applies also to multi-commodity flow, but the discussion in section “Dynamic Routing” does not. Future directions and further reading are suggested in sections “Summary and Future Directions” and “Recommended Reading” respectively. Finally, unless stated otherwise, the network topology is assumed to be directed acyclic such that, for every $v \in \mathcal{V} \setminus \{o, d\}$, there exists a directed path from o to v and from v to d .

Routing Dynamical Traffic Flow over Network

The set of f satisfying (1b)–(1c) can be parameterized in terms of, not necessarily unique, routing matrix $R \in [0, 1]^{\mathcal{E} \times \mathcal{E}}$, i.e., a row stochastic matrix whose (i, j) -th entry R_{ij} specifies what fraction of flow from i gets routed to j . Naturally, $R_{ij} = 0$ if i is not incident to j . Formally, for every f satisfying (1b)–(1c), there exists R such that

$$R^T f + \lambda \mathbf{1}_{i_{in}} = f \quad (2)$$

and vice-versa, where $\mathbf{1}_{i_{in}}$ is a vector of size $|\mathcal{E}|$ whose i_{in} -th entry is one, and other entries are zero.

Traffic Flow Dynamics

Equation (2) can be regarded as corresponding to equilibrium of the following mass balance equations:

$$\dot{x}_i = \sum_{j \in \mathcal{E}} R_{ji} f_j(x) - f_i(x), \quad i \in \hat{\mathcal{E}} \quad (3)$$

where x_i denotes traffic density (The density is assumed to be uniform over an edge. We also assume, without loss of generality, that all the edges have unit length.) on link i and $f_i(x)$ (For brevity in notation, we use the same notation f to denote a specific flow vector as well as the function which maps density x to outflow. The specific usage should be clear from the context.) is interpreted as the *outflow* from link i with $f_{i_{in}}(x) \equiv \lambda$. We now describe a few settings for dependence of f on x . A simple model is where, for $i \in \hat{\mathcal{E}}$, $f_i(x) \equiv d_i(x_i)$, where the d_i 's are continuously differentiable, strictly increasing and strictly concave such that $d_i(x_i) = 0$ and upper bounded by c_i . The latter models the capacity constraint in (1d). An example is:

$$d_i(x_i) = c_i (1 - e^{-\beta x_i}), \quad \beta > 0, \quad i \in \hat{\mathcal{E}} \quad (4)$$

A generalization is where

$$f_i(x) = \gamma_i(x) d_i(x_i), \quad \gamma_i(x) \in [0, 1] \quad (5)$$

In the *cell transmission model* (CTM) (Daganzo 1995) and its variants, $\gamma_i(x)$ captures the dependence of outflow on immediately downstream links, in the context of freeway networks, as follows. Equation (3) is augmented with:

$$\dot{x}_{i_{in}} = \lambda - f_{i_{in}}(x) \quad (6)$$

where, for CTM, $f_{i_{in}}(x)$ is as in (5) with finite $c_{i_{in}}$. The d_i 's, $i \in \hat{\mathcal{E}} \cup \{i_{in}\}$, are interpreted as *demand* functions, and each $i \in \hat{\mathcal{E}} \cup \{i_{out}\}$ is associated with a *supply* function $s_i(x_i)$. The s_i 's, $i \in \hat{\mathcal{E}}$, are continuously differentiable, strictly decreasing, and strictly concave, such that $s_i(0) > 0$ with $s_i(x_i^{\text{jam}}) = 0$ for some $x_i^{\text{jam}} > 0$ known as the *jam density* of i . See Fig. 1b for an illustration. By convention, $s_{i_{out}}(x) \equiv \infty$. Under a variant of CTM:

$$\gamma_i(x) = \sup \left\{ \theta \in [0, 1] : \max_{j \in \mathcal{E}} \left(\theta \cdot \sum_{k \in \mathcal{E}} R_{kj} d_k(x_k) - s_j(x_j) \right) \leq 0 \right\} \quad (7)$$

An alternative to (5) commonly used to study the average traffic flow dynamics over signalized networks is:

$$f_i(x_i) = \begin{cases} \gamma_i(x) c_i, & x_i > 0 \\ \min\{\gamma_i(x) c_i, \sum_j R_{ji} f_j(x)\}, & x_i = 0 \end{cases}, \quad i \in \hat{\mathcal{E}}, \quad \sum_{i \in \mathcal{E}_v^-} \gamma_i(x) \leq 1, \quad v \in \mathcal{V} \quad (8)$$

where γ_i 's are given by a specific traffic signal control policy.

A generalization of (3) is to dynamic routing which can be either open-loop as $R(t)$ or in feedback form:

$$\dot{x}_i = \sum_{j \in \mathcal{E}} R_{ji}(x) f_j(x) - f_i(x), \quad i \in \hat{\mathcal{E}} \quad (9)$$

Routing and outflow can be combined into inter-edge flow $z_{ij}(x) := f_i(x) R_{ij}(x)$ to further gener-

alize (9) as:

$$\dot{x}_i = \sum_{j \in \mathcal{E}} z_{ji}(x) - \sum_{j \in \mathcal{E}} z_{ij}(x), \quad i \in \hat{\mathcal{E}} \quad (10)$$

where $\sum_{j \in \mathcal{E}} z_{ij}(x) = f_i(x) \sum_{j \in \mathcal{E}} R_{ij}(x) = f_i(x)$. Equation (10) is augmented either with $\sum_{j \in \mathcal{E}} z_{i_{in}j}(x) = f_{i_{in}}(x) \equiv \lambda$ or with (6).

Fixed Routing

A necessary condition for the existence of an equilibrium x^* for (3) with $f(x)$ in (5) and

$\gamma_i(x) = 1$ for all $i \in \hat{\mathcal{E}}$ is that there exists $f^* \in [0, c]$ such that $f_i^* = \sum_{j \in \mathcal{E}} R_{ji} f_j^*$ for all $i \in \hat{\mathcal{E}}$ and $f_i^* = \lambda$ for $i \in \{i_{\text{in}}, i_{\text{out}}\}$. Under the same condition, existence of a global asymptotically stable x^* can be established. Additionally, robustness to loss in capacity, within the same formalism as in section “Introduction”, is equal to the minimum edge residual capacity, i.e., $\min_{i \in \hat{\mathcal{E}}} (c_i - f_i^*)$.

A necessary condition for the existence of equilibrium for (3), (8) is the existence of f^* satisfying the necessary condition above for (3), (5) with $\gamma_i(x) = 1$ for all $i \in \hat{\mathcal{E}}$, and additionally, $\sum_{i \in \mathcal{E}_v^-} \frac{f_i^*}{c_i} \leq 1$ for all $v \in \mathcal{V}$. If the necessary condition is satisfied, then under the centralized open-loop controller $\gamma_i(x) \equiv f_i^*/c_i$, $i \in \mathcal{E}$, every $x > \mathbf{0}$ is an equilibrium. It is shown in Savla et al. (2013) that, if the necessary condition is satisfied, then, under the decentralized controller $\gamma_i(x) = \frac{x_i}{\sum_{j \in \mathcal{E}_i^+} x_j + \kappa}$, $\kappa > 0$, there exists a unique $x^* > \mathbf{0}$ whose basin of attraction is $\mathbf{R}_{>0}^{\mathcal{E}}$. Extension of this result to cyclic graph topology, as would be relevant, e.g., for multi-commodity flow, is possible by establishing $\sum_{i \in \hat{\mathcal{E}}} x_i \log(\gamma_i(x) \frac{c_i}{f_i^*})$ as Lyapunov function, the inspiration for which comes from the literature on stochastic networks, e.g., see Massoulié (2007). Note that the decentralized controller gives the same stability guarantee as the centralized open-loop controller without requiring information about (even local) λ , R or c required to compute f^* . The literature on back pressure controllers, e.g., see Tassiulas and Ephremides (1992), suggests that one can get the same stability guarantee by establishing $\sum_{i \in \mathcal{E}} x_i^2$ as Lyapunov function. Such tight stability guarantees translate into tight guarantees on robustness to loss in capacities. That is, for a loss in capacity, if there exists a centralized controller under which $x(t)$ remains bounded, then it does so also under the above decentralized controllers, hence no loss in robustness under decentralized control.

A more accurate model to capture the alternating red and green states of traffic signal for a given road in signalized networks is (3) with $f_i(x) = c_i(t)$ if $x_i > 0$ and equal to $\min\{c_i(t), \sum_j R_{ji} f_j(x)\}$ if $x_i = 0$, where $c_i(t)$

is an exogenous periodic capacity function. In this setting, Muralidharan et al. (2015) provides tight conditions for the existence of a unique periodic $x^*(t)$ to which every trajectory converges to, and (Hosseini and Savla 2017) provides a procedure to compute $x^*(t)$. Extension to feedback-based periodic $c_i(t)$ is outstanding. Finally, comprehensive stability and robustness analysis of (9), (5) under (7) is still ongoing, with partial results reported in Lovisari et al. (2014).

Dynamic Routing

A necessary condition for the existence of an equilibrium x^* for (9) with $f(x)$ in (5) and $\gamma_i(x) = 1$ for all $i \in \hat{\mathcal{E}}$ is that λ is not greater than the min-cut capacity. Under this condition, existence and global asymptotic stability of x^* is shown in Como et al. (2013a, b) under a class of decentralized routing $R_{ij}(x, x^*) \equiv R_{ij}(\{x_\ell, x_\ell^* : \ell \in \mathcal{E}_{t_i}^+\})$, referred to as *monotone routing*, parameterized by x^* . This class is defined by two properties: (I) $R_{ij}(\{x_\ell^*, x_\ell^* : \ell \in \mathcal{E}_{t_i}^+\}) = \frac{f_j(x_j^*)}{\sum_{k \in \mathcal{E}_{t_i}^+} f_k(x_k^*)}$ for all $j \in \mathcal{E}_{t_i}^+$, $i \in \mathcal{E}$ and (II) $\frac{\partial}{\partial x_k} R_{ij}(\{x_\ell, x_\ell^* : \ell \in \mathcal{E}_{t_i}^+\}) \geq 0$ for all x , $\{j, k\} \in \mathcal{E}_{t_i}^+$, $j \neq k$, $i \in \mathcal{E}$. An example of monotone routing is:

$$R_{ij}(\{x_\ell, x_\ell^* : \ell \in \mathcal{E}_{t_i}^+\}) = \frac{f_j(x_j^*) e^{-\eta(x_j - x_j^*)}}{\sum_{k \in \mathcal{E}_{t_i}^+} f_k(x_k^*) e^{-\eta(x_k - x_k^*)}},$$

$$\eta > 0, \quad \forall i \in \mathcal{E} \setminus \{i_{\text{out}}\} \quad (11)$$

Monotone routing also possesses the following sharp robustness property. For every loss in capacity from c to $\tilde{c} \leq c$ such that $\|c - \tilde{c}\|_1$ is no greater than the minimum, over $v \in \mathcal{V} \setminus \{d\}$, *node residual capacity* w.r.t. $f(x^*)$, there exists a new globally asymptotically stable equilibrium under monotone routing. Residual capacity of v w.r.t. $f(x^*)$ is $\sum_{i \in \mathcal{E}_v^+} (c_i - f_i(x_i^*))$. On the other hand, if the loss in capacity on edges in $\mathcal{E}_v^+$ corresponding to v with minimum node residual capacity exceeds their respective residual capacities w.r.t. $f(x^*)$, then $\|x(t)\|$ grows unbounded under every decentralized, i.e., not necessarily monotone, routing. The minimum node residual capacity is less than or equal to the network residual

capacity, which, from section “[Introduction](#)”, is the analogous robustness measure in the static and centralized setting. The gap could be substantial for large networks, suggesting loss in robustness due to loss of information when routing. This gap can be overcome either by providing more information (Como et al. 2013c) or by augmenting routing with an extra control action (Como et al. 2015).

One way to provide more information is by letting the parameter x^* adapt to global information. In Como et al. (2013c), x^* is interpreted as *anticipated* traffic density by drivers who use it as a proxy for global state of the network when making local routing decisions according to monotone routing. x^* is updated by drivers using global information according to a best response dynamics over a slower time scale. If the time scale separation is sufficiently large and if λ is not greater than the min-cut capacity, then it is shown in Como et al. (2013c) that the Wardrop equilibrium (see section “[Routing Under Driver Preferences](#)”) is globally stable. An implication of this is that multiscale routing has the same robustness performance as centralized routing has for the static setting, as described in section “[Introduction](#)”.

In Como et al. (2015), it is shown how loss in robustness can also be recovered by augmenting decentralized routing with decentralized flow control action, as in (10). Specifically, it is shown that, under a monotone decentralized $z_{ij}(x) \equiv z_{ij}(\{x_\ell : \ell \in \{i\} \cup \mathcal{E}_i^+\})$, if λ is no greater than the min-cut capacity, then (10) admits a globally asymptotically stable equilibrium, which then also implies the same robustness performance as centralized routing for the static setting in section “[Introduction](#)”. This result is true even if the graph topology is cyclic, to allow multiple origins and destinations within single commodity flow setting. The monotonicity conditions are, for all $i \in \mathcal{E}$: $\frac{\partial}{\partial x_k} z_{ij}(\{x_\ell : \ell \in \{i\} \cup \mathcal{E}_i^+\}) \geq 0$ for all $k \in \{i\} \cup \mathcal{E}_i^+ \setminus \{j\}$ and $\frac{\partial}{\partial x_k} \sum_{j \in \mathcal{E}_i^+} z_{ij}(\{x_\ell : \ell \in \{i\} \cup \mathcal{E}_i^+\}) \leq 0$ for all $k \in \mathcal{E}_i^+$. An example is $z_{ij}(x) = \gamma_i(x) d_i(x_i) R_{ij}(x)$ with $\gamma_i(x) = \frac{\sum_{k \in \mathcal{E}_i^+} e^{-\eta x_k}}{\sum_{k \in \{i\} \cup \mathcal{E}_i^+} e^{-\eta x_k}}$, $\eta > 0$, $d_i(x_i)$ as in (4) and $R_{ij}(x) = \frac{e^{-\eta x_j}}{\sum_{k \in \mathcal{E}_i^+} e^{-\eta x_k}}$.

Rigorous stability and robustness analysis of (9), (6), (5) under (7) is outstanding. However, optimal solution to the following dynamic version of (1), also known as *dynamic traffic assignment*, has been derived in Como et al. (2016); Jafari and Savla (2019): $\min_{(\alpha(t), R(t)): t \in [0, T]} \int_0^T \psi(x(s), f(s)) ds$ s.t. (9), (6), (5), (7) with $d_i(x_i)$ replaced by $\alpha_i d_i(x_i)$, convex ψ , given initial condition $x(0)$ and control horizon T . $\alpha_i \in [0, 1]$ is interpreted as variable speed limit control on $i \in \hat{\mathcal{E}}$ and as ramp metering control on i_{in} . The solution is synthesized in two steps. First, an optimal solution is obtained for the following convex relaxation of (7): $\min_{(x(t), z(t)): t \in [0, T]} \int_0^T \psi(x(s), f(s)) ds$ s.t. (10), (6), initial condition $x(0)$, $\sum_j z_{ij}(t) \leq d_i(x_i(t))$ for all $i \in \hat{\mathcal{E}} \cup \{i_{in}\}$ and $t \in [0, T]$, and $\sum_j z_{ji}(t) \leq s_i(x_i(t))$ for all $i \in \hat{\mathcal{E}} \cup \{i_{out}\}$ and $t \in [0, T]$. In the second step, setting $R_{ij}^*(t) = \frac{z_{ij}^*(t)}{\sum_{k \in \mathcal{E}_i^+} z_{ik}^*(t)}$ and $\alpha_i^*(t) = \frac{\sum_{j \in \mathcal{E}_i^+} z_{ij}^*(t)}{d_i(x_i^*(t))}$ implies that $\{(x_i^*(t), f_i^*(t) = \alpha_i^*(t) d_i(x_i^*(t)) : i \in \hat{\mathcal{E}} \cup \{i_{in}\})\}$ satisfies (7) with $\gamma_i(x^*(t)) = \alpha_i^*(t)$. This proves optimality of $(\alpha^*(t), R^*(t))$ for the original unrelaxed problem. If ψ is separable over the edges, then the first step can be executed efficiently using distributed iterative algorithms, e.g., as in Ba and Savla (2016). If ψ is linear, then, in discrete time setting, using multi-parametric programming technique, the optimal solution in the first step can be equivalently expressed in feedback form $z^*(x(t))$ which is piecewise affine. Additionally, Jafari and Savla (2019) provides the most general known sufficient condition on the network topology and coefficients in linear ψ under which the feedback structure admits a decentralized form. These then translate into corresponding feedback and decentralization forms for (α^*, R^*) in the second step.

Routing Under Driver Preferences

The routing policies presented in section “[Routing Dynamical Traffic Flow over Network](#)” can be regarded as minimizing social cost, whether that be the negative of robustness or others modeled by ψ . For instance, the monotone routing in

(11) can be regarded as the probability of a single driver choosing edge j among $\mathcal{E}_{\tau_i}^+$ under the multinomial logit model, if the cost, or disutility, associated with $j \in \mathcal{E}_{\tau_i}^+$ is $x_j - x_j^* - \log f_j(x_j^*)$. However, the cost actually assigned by drivers might be different. The routing outcome under such user-centric cost functions in the static setting is given by *user equilibrium*. A flow assignment is called user equilibrium if the cost of each path from o to d actually used, i.e., on which flow is positive, is equal and is less than or equal to the cost of a path not used. A natural candidate for user-centric cost function is travel time; the user equilibrium in this special case is also often referred to as *Wardrop equilibrium*.

User equilibrium flow, in general, does not minimize social cost. The disparity is often formalized through *price of anarchy*, e.g., see Nisan et al. (2007), defined as the ratio between the social cost of user equilibrium flow and the minimum possible under socially optimal routing. In case of multiple user equilibria, it is defined to be the largest ratio over all user equilibria. Edge-wise tolls are among the most widely studied mechanisms to reduce the price of anarchy, as follows. Since tolls affect the cost that a driver associates with a path, and therefore also the resulting user equilibrium flow, they can potentially be used to align user equilibrium flow with socially optimal flow. Indeed, let f^{opt} minimize $\sum_i f_i \cdot \text{TT}_i(f_i)$ subject to constraints in (1), where $\text{TT}_i(f_i)$ is flow-dependent travel time on edge $i \in \hat{\mathcal{E}}$. Let the user cost of a path be the sum of costs of edges on that path, and let the user cost for an edge i be the sum of $\text{TT}_i(f_i)$ and fixed toll τ_i . If, for every $i \in \hat{\mathcal{E}}$, $\tau_i = f_i^{\text{opt}} \cdot \text{TT}'_i(f_i^{\text{opt}})$, TT_i is continuous differentiable and strictly increasing, and $f_i \cdot \text{TT}_i(f_i)$ is convex, then f^{opt} is also a user equilibrium (Nisan et al. 2007). Here, TT'_i denotes the derivative of TT w.r.t. f_i . Optimal toll design in dynamical settings is outstanding in general.

Summary and Future Directions

Routing design for transportation networks has traditionally focused on centralized, static, or open-loop settings. Distributed feedback routing

under realistic traffic flow dynamics, preferably under output feedback that captures common sensing modes in traffic, is a fruitful direction to pursue. In this context, multi-commodity flow particularly remains largely unexplored and therefore presents great opportunities. While monetary mechanisms, such as tolls, have been studied extensively as means to align user- and system-level objectives, information design schemes are gaining interest in this context. However, it remains to integrate them properly with the transportation settings.

Cross-References

- [Multi-vehicle Routing](#)
- [Network Games](#)

Recommended Reading

Extensive discussion on traffic assignment models and solution algorithms can be found in Patriksson (1994); Cascetta (2009). The majority of the stability analysis reported in section “[Routing Dynamical Traffic Flow over Network](#)” utilizes contraction theory, e.g., see Sontag (2010), by exploiting monotonicity of underlying dynamical systems (Hirsch and Smith 2006). Distributed routing under cascading failure dynamics is relatively unexplored; one specific setting is studied in Savla et al. (2014).

Bibliography

- Ba Q, Savla K (2016) On distributed computation of optimal control of traffic flow over networks. In: Allerton conference on communication, control and computing, pp 1102–1109
- Cascetta E (2009) Transportation systems analysis. Springer
- Como G, Savla K, Acemoglu D, Dahleh MA, Frazzoli E (2013a) Robust distributed routing in dynamical networks – part I: locally responsive policies and weak resilience. IEEE Trans Autom Control 58(2):317–332
- Como G, Savla K, Acemoglu D, Dahleh MA, Frazzoli E (2013b) Robust distributed routing in dynamical networks – part II: strong resilience, equilibrium selection and cascaded failures. IEEE Trans Autom Control 58(2):333–348
- Como G, Savla K, Acemoglu D, Dahleh MA, Frazzoli E (2013c) Stability analysis of transportation networks

- with multiscale driver decisions. *SIAM J Control Optim* 51(1):230–252
- Como G, Lovisari E, Savla K (2015) Throughput optimality and overload behavior of dynamical flow networks under monotone distributed routing. *IEEE Trans Control Netw Syst* 2(1):57–67
- Como G, Lovisari E, Savla K (2016) Convexity and robustness of dynamic traffic assignment for control of freeway networks. *Transp Res B Methodol* 91:446–465
- Daganzo CF (1995) The cell transmission model, part II: network traffic. *Transp Res B Methodol* 29(2):79–93
- Hirsch M, Smith H (2006) Monotone dynamical systems, vol. 2 of *Handbook of differential equations: ordinary differential equations*, chapter 4, pp 239–357. North-Holland
- Hosseini P, Savla K (2017) Queue length simulation for signalized arterial networks and steady state computation under fixed time control. Available at <https://arxiv.org/abs/1705.07493>
- Jafari S, Savla K (2019) On structural properties of optimal feedback control for traffic networks. In: *American control conference*, Philadelphia
- Lovisari E, Como G, Rantzer A, Savla K (2014) Stability analysis and control synthesis for dynamical transportation networks. Available at <http://arxiv.org/abs/1410.5956>
- Massoulié L (2007) Structural properties of proportional fairness: stability and insensitivity. *Ann Appl Probab* 17:809–839
- Muralidharan A, Pedarsani R, Varaiya P (2015) Analysis of fixed-time control. *Transp Res B Methodol* 73: 81–90
- Nisan N, Roughgarden T, Tardos E, Vazirani V (eds) (2007) *Algorithmic game theory*. Cambridge University Press
- Patriksson M (1994) *The traffic assignment problem: models and methods*. V.S.P. Intl Science
- Savla K, Lovisari E, Como G (2013) On maximally stabilizing adaptive traffic signal control. In: *Allerton conference on communication, control and computing*, Monticello, pp 464–471
- Savla K, Como G, Dahleh MA (2014) Robust network routing under cascading failures. *IEEE Trans Netw Sci Eng* 1(1):53–66
- Sontag ED (2010) Contractive systems with inputs. In: *Perspectives in mathematical system theory, control, and signal processing*, pp 217–228. Springer
- Tassiulas L, Ephremides A (1992) Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans Autom Control* 37(12):1936–1948

Run-to-Run Control in Semiconductor Manufacturing

James Moyne

Mechanical Engineering Department, University of Michigan, Ann Arbor, MI, USA

Abstract

Run-to-run (R2R) control is a form of adaptive model-based process control that can be tailored to environments where the process is discrete, dynamic, and highly unobservable; this is characteristic of processes in the semiconductor manufacturing industry. It generally has, at its roots, a rather straightforward approach to adaptive model-based control such as a first-order linear plant model with moving average weighting applied to adapt the (zeroth-order) constant term in the model. Most of the complexity of R2R control science lies and will continue to lie in extensions to support practical application of R2R control in semiconductor manufacturing facilities of the future; these extensions include support for weighting and bounding of parameters, run-time modeling of a large number of disturbance types, and incorporating prediction information such as virtual metrology and yield prediction into the control solution. As we move forward in the era of smart manufacturing and Industrie 4.0, R2R control will play a key role in the microelectronics manufacturing ecosystem as a process digital twin.

Keywords

Adaptive control · Advanced process control (APC) · EWMA control · Feed-forward and feedback control · Model-based control · R2R control · Run-to-run control · Single-threaded control · Virtual metrology · Wafer-to-wafer control · Yield prediction · Digital twin (DT)

RTO

► [Real-Time Optimization of Industrial Processes](#)

Introduction

The semiconductor manufacturing industry involves the processing of semiconductor “wafers” using a variety of physical and chemical processes to produce dies or “chips” that contain a number of nanometer size features organized in layers. As feature sizes shrink, the industry must innovate to maintain acceptable product yield and throughput. One effective dimension of innovation that has been utilized since the early 1990s is model-based process control. The use of this technology in semiconductor manufacturing has been largely industry specific due to unique industry requirements and been given the name “run-to-run (R2R) control.”

R2R control is defined as “...a form of discrete process and machine control in which the product recipe with respect to a particular machine process is modified *ex situ*, i.e., between machine ‘runs,’ so as to minimize process drift, shift, and variability” (Moyné et al. 2000). (The “recipe” is the group of process settings for a process or process step, e.g., temperature, flow, and pressure.) The term “R2R control” was coined in the early 1990s in the semiconductor industry as the industry struggled to come up with mechanisms to keep critical semiconductor manufacturing processes such as chemical vapor deposition (CVD), chemical mechanical polishing (CMP), and reactive ion etching (RIE) under control. The processes are highly unobservable and are subjected to a number of disturbances. However, many of these disturbances can be modeled or tracked as they create measurable shifts in the process (e.g., after a maintenance operation) or gradual drifts in the process (e.g., chamber wall “seasoning” of an etch process over time, resulting in polymer buildup on chamber walls, causes changes to the operational effectiveness of the tool). Process and product quality is generally assessed through metrology measurements made *ex situ*, i.e., after the process is complete; examples of post-process metrology parameters are wafer average deposited or removed film thickness and film uniformity. R2R control generally uses statistically developed models of tool process

operation updated or “tuned” with process metrology feedback information on a “run-to-run” basis to keep the process under control and process quality high, in the face of these process drifts and shifts, as shown in Fig. 1. Note that the granularity of control could be wafer-to-wafer, batch-to-batch (“lot-to-lot”), etc.

Run-to-Run Control Approach

Because the processes are highly unobservable and dynamic, rather simple model forms are usually employed with filtering techniques used to track process shift and drift. The most commonly utilized R2R controller in the industry is the exponentially weighted moving average (EWMA) controller. The algorithm uses a linear model with an additional constant term. (Equations will use the following notation: arrays of vectors will be capitals, vectors will be lower case, and indexing within a vector or matrix will be lower case with subscripts. In addition, the special subscript “t” will be reserved for time or run number information.)

$$Y = Ax + c \quad (1)$$

where:

y = System output,

x = Input (Recipe),

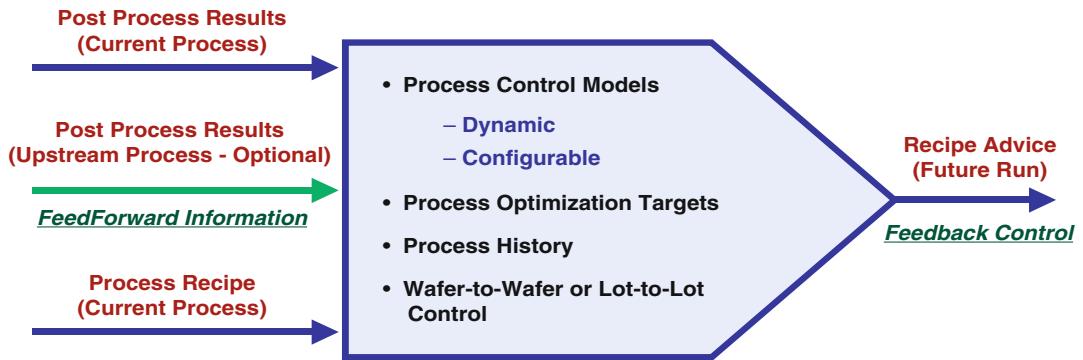
A = Slope coefficients for equation,

c = Constant term for linear model.

Each output represents a target of control (usually measured by pre- and post-process metrology tools), and each input represents an adjustable parameter in the recipe.

$$\begin{aligned} y_1 &= a_{11}x_1 + a_{12}x_2 + \dots a_{1m}x_m + c_1 \\ &\dots \\ y_n &= a_{n1}x_1 + a_{n2}x_2 + \dots a_{nm}x_m + c_n \end{aligned} \quad (2)$$

The models are generally developed by executing a design of experiments (DOE), where the process area is explored with respect to the allowed variation of the process inputs by



Run-to-Run Control in Semiconductor Manufacturing, Fig. 1 Input/output structure of a typical R2R control solution

processing wafers with various input settings (see, e.g., Box and Draper 1987). Statistical packages are then used to determine the base model of the form described in (1) at the normal process operating point. As the processes are dynamic, the base model is updated on a “run-to-run” basis to compensate for model error. The algorithm operates under the assumption that the underlying process is locally approximated by a first-order linear polynomial model in the form of equation (1) and that this polynomial model can be maintained near a local optimal point solely by updating the constant term “c.”

The control process involves updating the model and then using that model to compute a recipe update. The model is updated by first comparing the actual process output, Y_t , to the model-predicted process output, AX_t . Using an EWMA filtering technique as an example, the constant term, c_t , can be updated as follows:

$$c_t = \alpha(y_t - Ax_t) + (I - \alpha)c_{t-1} \quad (3)$$

where α is a weighting factor between 0 and 1, often called a “forgetting factor.” Note that because of the additive nature of the EWMA series, the C_t calculation only requires knowledge (and storage) of the previous run measurements; this, combined with its relative simplicity, led to the widespread adoption of EWMA as the R2R controller filter of choice in this industry during the 1990s and early 2000s.

Once the model is updated, the process recipe is calculated. Since there are generally more inputs that can be tuned than outputs measured, the process is underdetermined, and there is an infinite solution space. Approaches such as Lagrange multipliers are used to determine the solution that is closest to the previous solution (Moyné et al. 2000).

Many extensions and alternatives to this basic approach have been developed and deployed over the past 10 years. These include (1) the replacement of EWMA filtering with other approaches such as the more general Kalman filtering, (2) explicitly modeling drift (termed “predictor corrector”), (3) modeling updates to first-order terms (in the “A” matrix), and (4) leveraging phenomenological models that capture process knowledge in equation forms, customized and tuned with statistical data. Perhaps the most important extensions to the basic approach involve addressing the practical issues associated with control systems application in this area. For example, providing capabilities for addressing bounding, weighting, and granularity (e.g., integer) of input and output settings often requires much more programming effort than supporting the core algorithm (Moyné et al. 2000).

Current Status and Future Extensions

Over the past 20 years, R2R control has evolved from a value-added capability applied to a few

processes to a required component to achieve cost and productivity competitiveness in most processes in the semiconductor manufacturing industries (IRDS 2018). As part of this evolution, a number of common trends in the R2R control space have emerged:

Support for fab-wide reusable and reconfigurable solutions for R2R control: As the benefits of R2R control were proven across multiple processes in semiconductor fabrication facilities, the focus turned to reusable and reconfigurable integrated “fab-wide” solutions for R2R control. The event-based capabilities described in Chapter 9 of Moyne et al. (2000) were leveraged to provide these solutions as they allow for integration and configuration of R2R control solutions to the particular application environment. This event-based approach has also been used to integrate R2R control with other capabilities such as fault detection and classification (FDC), work scheduling, and “virtual metrology” (see below), to provide another level of benefits toward improved product yield and throughput (Moyne 2004, 2009; Khan et al. 2007).

Movement to more granular control: The ever more stringent requirements on product quality are being addressed in large part by a movement from batch-level control (often called “lot-based control” in this domain), to wafer-level control (usually called “wafer-to-wafer” (W2W) control), to within-wafer (WIW) control. Although the granularity has changed, the basic approach to control has not. It is important to note that the improvement in quality associated with this trend results mostly from the use of pre-(process) metrology to reject incoming product disturbances, rather than post metrology to address the dynamics of the plant model (Moyne et al. 2000; IRDS 2018).

Support for control across multiple recipes using “single-threaded control”: Semiconductor manufacturing process control systems are characterized by a number of disturbance types that usually can be modeled as independent from the base process model and from each other. Perhaps the most common type of disturbance that is addressed is recipe or product change. When there is a change in product and related product

recipe, a single-process model must be adjusted to capture this disturbance while maintaining knowledge of process drift and/or shift. Oftentimes this process disturbance can be modeled as a shift to the overall process. Thus, the process model of Eq. (1) can be adjusted to the following:

$$Y = Ax + c_1 + c_2 + c_3 + \dots + c_n \quad (4)$$

where:

$c_1, c_2, \dots, c_{n-1}$ = constant terms associated with modeled disturbances such as product

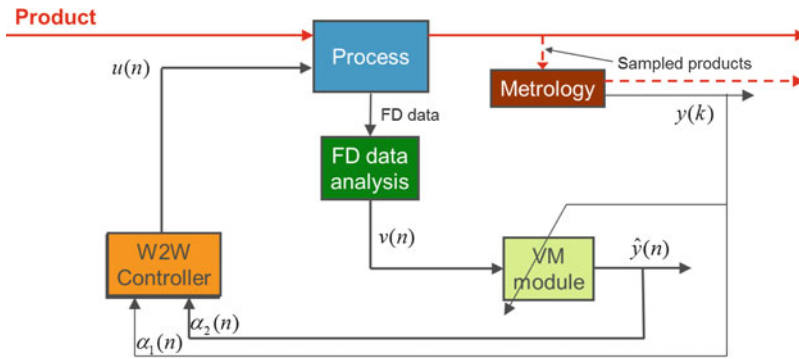
c_n = constant term associated with process dynamics (drift and shift)

$c_1 + c_2 + \dots + c_n = c$ in Eq. (1)

Approaches have been devised for the assessment of c_i associated with a particular disturbance type (Edgar et al. 2004; Zou 2013); the result is that a single control model can be used across multiple product recipes and other disturbance types.

Enhancing R2R control with “virtual metrology”: Ex situ metrology plays a crucial role in semiconductor manufacturing as it is often the only source of product quality data before and after a process. However, given its high capital equipment cost and cycle time impact on critical processes, optimizing metrology by minimizing wasteful use and optimizing measurement value is important. Virtual metrology (VM) is a technology rapidly gaining acceptance in the marketplace as an efficient and cost-effective way to optimize and augment metrology value. VM is a modeling and metrology prediction solution, whereby process and product data, such as in situ fault detection (FD) information and upstream metrology information, is correlated to post-process metrology data. This same data can then be used to predict metrology information when conventional metrology data is not available (Khan et al. 2007; Cheng et al. 2011).

One of the uses of VM that is expected to become prominent over the next decade is in support of enhanced R2R control. As shown in Fig. 2, fault detection (FD) summary information is used along with adaptive VM modeling to



Run-to-Run Control in Semiconductor Manufacturing, Fig. 2 Virtual metrology enhanced R2R control

predict metrology information. The VM predictions are then used to fill in the measurement gaps in feed-forward and feedback control, thus enabling wafer-to-wafer or even within-wafer control. One of the research challenges is to optimally tune the control to best utilize both the real and predicted metrology information. This requires that VM data contain information on predicted measurement data quality (Khan et al. 2007).

$u(n)$ Tunable process inputs

$v(n)$ FD summary information

$y(k)$ Metrology measurement data for measured wafers

$\hat{y}(n)$ Predicted metrology measurements for all wafers

$\alpha_1(n)$ Feedback filter coefficient for feedback of measured data

$\alpha_2(n)$ Feedback filter coefficient for feedback of predicted data

Movement toward interprocess and eventually fab-wide control: The generally accepted vision of the future of advanced process control (APC) in general is a fabrication-wide fully integrated solution that incorporates all of the APC capabilities (R2R control, FDC, fault prediction, and statistical process control) as well as predictive capabilities such as predictive scheduling, predictive maintenance, virtual metrology, and predictive yield (IRDS 2018). Opportunities for research and development exist with the integration of these technologies, especially as the powers of the predictive domain are tapped. For

example, it is expected that R2R control will eventually incorporate predicted yield as a target with feedback to multiple coordinated process controllers (Moyné and Schulze 2010). Thus, the future of research in R2R control, while evolving, should remain strong in the coming years.

Summary and Future Directions

R2R control is a form of adaptive model-based process control that is tailored to environments where the process is discrete, dynamic, and highly unobservable; this is characteristic of processes in the semiconductor manufacturing industry. R2R control has evolved from a strictly research effort in the early 1990s to a required facility-wide capability in all of semiconductor manufacturing. It generally has, at its roots, a rather straightforward approach to adaptive model-based control. Most of the complexity of R2R control science lies and will continue to lie in extensions to support practical application of R2R control in semiconductor manufacturing facilities of the future.

The science of R2R control will continue to expand as the academic and industry communities look to incorporating capabilities that will allow R2R control to continue to be an integral part of the fabrication facility of the future. One key research direction over the next decade is the development of approaches for incorporating virtual metrology and yield prediction into control solutions. Other focus areas will likely include

hybrids of R2R control and continuous process control, learning mechanisms for single-threaded control in “high-mix” environments where there are a large number of disturbances that should be modeled, phenomenological R2R control models, and model libraries that combine stochastic information with process physics and chemistry knowledge, control solutions that are more directly optimized to financial parameters such as yield and throughput, and R2R control solutions that incorporate other analysis capabilities, such as FDC, either algorithmically or via event-based control rule approaches.

R2R control technology will also play an important role in the efforts associated with the move to smart manufacturing and Industrie 4.0 (Thoben et al. 2017; Moyne and Iskandar 2017). Specifically, R2R control is considered a process digital twin (DT), with R2R controllers becoming an integral part of a Smart Manufacturing DT framework (Qamsane et al. 2019). This integration will help facilitate the exploration of R2R control opportunities in other domains such as process commissioning and maintenance recovery, supply chain control and optimization, and control to fabrication facility yield objectives, but also to end-user yield (often termed latent yield) objectives. Additionally, the DT framework will allow R2R controllers to be integrated with other DT types; for example, combining R2R control and scheduling/dispatch DTs can allow for the inclusion of process quality metrics in scheduling/dispatch optimization decisions (Lopez et al. 2017). Each of these topics provides significant opportunity for research as well as benefit in application to the semiconductor manufacturing ecosystem.

Cross-References

- [Adaptive Control: Overview](#)
- [Controllability and Observability](#)
- [Event-Triggered and Self-Triggered Control](#)

- [Experiment Design and Identification for Control](#)
- [Fault Detection and Diagnosis](#)
- [Kalman Filters](#)
- [Moving Horizon Estimation](#)
- [Nominal Model-Predictive Control](#)
- [Robust Model Predictive Control](#)
- [Stochastic Model Predictive Control](#)

Bibliography

- Advanced Process Control (APC) Conference Proceedings (2000–2014) Titles available at <http://www.apconference.com>
- Box GEP, Draper NR (1987) Empirical model-building and response surfaces. Wiley, New York
- Cheng F-T et al (2011) Benefit model of virtual metrology and integrating AVM into MES. *IEEE Trans Semicond Manuf* 24(2):261–272
- Edgar TF, Firth SK, Bode C (2004) Multi-product run-to-run control for high-mix fabs (session keynote) *AEC/APC Asia*, Hsinchu. Available at http://140.113.156.45/files/reference/2nd%20AEC-APC%20Symposium%20Asia/APCAEC/presentations/session_KeynoteInvited/Edgar.Thomas.pdf
- International Roadmap for Devices and Systems (IRDS) (2018) Edition, IEEE. Available at <http://IRDS.ieee.org>. (See especially “Factory Integration” chapter)
- Khan A, Moyne J, Tilbury D (2007) Fab-wide control utilizing virtual metrology (invited). *IEEE Trans Semicond Manuf-Spec Issue Adv Process Control* 20(7):364–375
- Lopez F et al (2017) Process-capability-aware scheduling/dispatching in wafer fabs. In: Advanced process control conference XXIX proceedings, Austin. Available via <http://www.apconference.com>
- Moyne J (2004) The evolution of APC: the move to total factory control (invited). *Solid State Technol* 47(9): 47–52
- Moyne J (2009) A blueprint for enterprise-wide deployment of advanced process control. *Solid State Technol* 52(7):35–37
- Moyne J, Iskandar J (2017) Big data analytics for smart manufacturing: case studies in semiconductor manufacturing (invited). *Proc J* 5(7). Available at <http://www.mdpi.com/2227-9717/5/3/39/html>
- Moyne J, Schulze B (2010) Yield management enhanced advanced process control system (YMeAPC): part I, description and case study of feedback for optimized multi-process control. *IEEE Trans Semicond Manuf Spec Issue Adv Process Control* 23(2):221–235
- Moyne J, Del Castillo E, Hurwitz A (2000) Run-to-run control in semiconductor manufacturing. CRC, Boca Raton

- Qamsane Y et al (2019) A unified digital twin framework for real-time monitoring and evaluation of smart manufacturing systems. In: 2019 IEEE 15th international conference on automation science and engineering (CASE), Vancouver
- Thoben K-D, Wiesner S, Wuest T (2017) “Industrie 4.0” and smart manufacturing a review of research issues and application examples. *Int J Autom Technol* 11: 4–16
- Zou J (2013) Method and system for estimating context offsets for run-to-run control in a semiconductor fabrication facility. United States Patent, Patent Number US 8,355,810 B2 (Filed, Jan 2010; Issued, Jan 2013)

Safety Guarantees for Hybrid Systems

Raphael M. Jungers¹ and Nikolaos Athanasopoulos²

¹UCLouvain, ICTEAM Institute, Louvain-la-Neuve, Belgium

²School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, UK

Abstract

Hybrid systems describe processes that typically need to satisfy a set of strict physical, computation, and communication constraints. Mission-critical and time-critical cyber-physical systems are a prime example where these constraints play a key role in analysis, controller synthesis, and implementation. On top of classical notions such as stability, safety plays a major role in the control design of hybrid systems. There is a long history of methods related to the safety analysis and safety enforcement for dynamical systems, with the ones concerning linear systems being more mature than the others. Due to the importance and complexity of the underlying problem, several different techniques have

been developed for hybrid systems. This entry summarizes the most important approaches and tools, together with references for further reading.

Keywords

Hybrid systems · Safety · Verification · Reachability · Abstraction · Hybrid automata · Switching · Lyapunov methods · Formal methods

Introduction

More and more applications use advanced, data-driven, decentralized, nonlinear control solutions for uncertain or complex models. This complexification leads to unavailability of classical guarantees on the system's behavior. Safety-critical control is a leading challenge in hybrid and cyber-physical systems. Examples can be found in (air) traffic control, medicine, drug administration, logistics and supply chain networks, robotics, planning, the smart grid, autonomous vehicles and autonomous navigation, digital manufacturing and Industry 4.0, control over networks, and edge and cloud computing. Safety analysis of such systems is extremely difficult, with negative complexity results holding even for simple hybrid dynamics, for example, discrete-time switching systems (Jungers 2009) and rectangular hybrid automata (Doyen et al. 2018). Many different approaches have been established to address

R.J. and N.A. are supported by the CHIST-ERA 2018 project DRUID-NET "Edge Computing Resource Allocation for Dynamic Networks"

the problem of safety, often leading to semi-algorithms with asymptotic properties. In view of the hardness of the task, it is critical to identify particular features of the system, which could make safety problems easier than in the general case. Indeed, for several special cases of hybrid systems, decidability or semi-decidability results are available. Let us mention reachability for timed automata, or the stability problem for switching systems (see, respectively, (Alur and Dill 1994; Jungers 2009), and (Doyen et al. 2018, Section 30.3) for precise statements). However, as a rule of thumb, one can argue that even elementary safety problems on very simple hybrid systems are often extremely hard.

Models

A *hybrid system* is a dynamical system consisting of both discrete (discrete event) and continuous dynamics, along with the corresponding sets where these two different dynamics are defined, often called *flow* and *jump* sets. A formal definition together with conditions for the existence of solutions is in Goebel et al. (2012, Sec. 1.1), covering, among others, switching and impulsive systems, and hybrid automata. For equivalence results between classes of hybrid systems, see Heemels et al. (2001), and for a formal definition for hybrid automata, see Doyen et al. (2018, Sec. 30.2.2). In this entry we consider the simple, however not restrictive, mathematical model representation

$$\begin{cases} \dot{x} = f(x, u, w), & x \in C, \\ x^+ = g(x, u, w), & x \in D, \end{cases} \quad (1)$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state vector, defined in a space that may consist of both discrete and continuous variables, $C \subset \mathbb{R}^n$, $D \subset \mathbb{R}^n$, $D \cap C = \emptyset$, are subsets of the state space and $u(t) \in \mathcal{U}$ and $w(t) \in \mathcal{W}$ are the control and exogenous inputs, respectively. We denote the solution of the system at time t^* for an initial condition $x_0 \in \mathcal{X}$ with $x(t^*; x_0)$. Further details on the model, as well as conditions and subtleties on the existence of solutions of (1), can be found in Goebel et al. (2012).

Safety

Safety is at the heart of many verification problems of hybrid systems. To introduce the concept, we first define the notion of *reachability*; a set $\mathcal{X}_e$ is *reachable* from x_0 if the state can reach $\mathcal{X}_e$ from x_0 after some finite time t , i.e., if $x(t; x_0) \in \mathcal{X}_e$. The notion of safety is related to *avoiding* regions of the state space; given a set of *unsafe states* $\mathcal{X}_f$ and a set of initial states $\mathcal{X}_0$, a system is *safe* if for any initial state x_0 in the set $\mathcal{X}_0$, there exists a sequence of actions such that it never reaches $\mathcal{X}_f$, i.e., $x(t; x_0) \notin \mathcal{X}_f$, for all $t > 0$ and all $x_0 \in \mathcal{X}_0$. We note that safety is meaningful also in the absence of control inputs, in the context of system analysis.

Safety Verification and Safety-Critical Control

As already mentioned, the literature on safety analysis for hybrid systems is huge, with ramifications in Mathematics, Computer Science, Robotics, etc. In the rest of the note, an attempt is made to address the challenge of classifying and briefly surveying the existing approaches. The following subsection presents perhaps the major and more natural technique, namely, set propagation. Next, optimization-based methods are presented, followed by an exposition of formal methods, which have received a strong influence from the Computer Science literature. The last subsection discusses probabilistic methods that seem particularly promising at the era of data-driven systems and massive computations.

Set Propagation Methods

Safety is connected to non-violation of constraints in the state space. Thus, a natural way of verifying it is to propagate relevant sets, e.g., constraint/initial/target sets, across time. This is called *reachability analysis* or *set propagation*.

Reachability analysis, e.g., Aubin et al. (2011), is strongly related to dynamic programming. In control, the connection has been made obvious and exploited with the model predictive control (MPC) paradigm and the utilization of

invariant sets for the enforcement of recursive feasibility, stability, and optimality for closed-loop control systems. Depending on whether one considers the propagation of dynamics of the system forward or backward in time, reachability may be utilized for different objectives, such as computing invariant sets and designing controllers (Blanchini and Miani 2008, Sections 5, 6), reaching a target set, verifying safety from an initial set, or satisfying more complex temporal specifications and objectives (Belta et al. 2017). We present below two popular instances of forward and backward reachability maps, noting however that several variants exist, which serve slightly different objectives and purposes.

Forward Reachability

Forward reachability can be utilized to study where the trajectories of the system will lie in the future when starting from some set of initial conditions of interest. Here, we restrict to systems with only exogenous inputs $w(\cdot)$, as it is the most common case where these mappings are utilized in the literature. Given an initial set $\mathcal{X}_0$, the approach consists in generating the sequence $\{\mathcal{R}_i\}$ (adapted from Doyen et al. 2018, Sec. 30.4.1) with

$$\mathcal{R}_{i+1} = \mathcal{R}_i \cup \mathcal{F}_C(\mathcal{R}_i) \cup \mathcal{F}_D(\mathcal{R}_i), \quad (2)$$

$\mathcal{R}_0 = \mathcal{X}_0$, and

$$\begin{aligned} \mathcal{F}_C(\mathcal{R}_i) &= \{x(t^*; x_0) : (\exists x_0 \in \mathcal{R}_i \cap C, \exists w(t) \in \mathcal{W} : x(t; x_0) \in C, t \in [0, t^*])\}, \\ \mathcal{F}_D(\mathcal{R}_i) &= \{g(x_0, w) : x_0 \in \mathcal{R}_i \cap D, w \in \mathcal{W}\}. \end{aligned}$$

If all elements $\mathcal{R}_i$ of the generated sequence do not intersect with the set of unsafe states, i.e., $\mathcal{R}_i \cap \mathcal{X}_f = \emptyset$, then the system is safe.

Backward Reachability

Going backwards in time, we can define the sequence (adapted from Aubin et al. 2011, Sec

1.2.1.1, Viability Kernel) $\{\mathcal{C}_i\}$ generated by

$$\mathcal{C}_{i+1} = \mathcal{C}_i \cap (\mathcal{B}_C(\mathcal{C}_i) \cup \mathcal{B}_D(\mathcal{C}_i)), \quad (3)$$

with $\mathcal{C}_0 = \mathcal{X} \setminus \mathcal{X}_f$, and

$$\begin{aligned} \mathcal{B}_C(\mathcal{C}_i) &= \{x_0 \in C : (\exists t^* \geq \kappa, \exists u(t) \in \mathcal{U}, \forall t \in [0, t^*] : x(t; x_0) \in C \setminus \mathcal{X}_f, x(t^*; x_0) \in \mathcal{C}_i)\}, \\ \mathcal{B}_D(\mathcal{C}_i) &= \{x_0 \in D : (\exists u \in \mathcal{U} : g(x_0, u, w) \in \mathcal{C}_i)\}. \end{aligned}$$

In the formula above, κ is an arbitrary positive constant, essentially limiting the dwell time for the continuous-time dynamics. Let us add that it is assumed in the formula above that the value u of the controller is valid for any possible value of w ; however one can model different situations, where, e.g., the controller knows the input noise. If the limit of the generated sequence exists and is not empty, then the system is safe within this limit set, in the sense that all trajectories starting from it can be controlled such that they do not enter the unsafe region. Together with answering the safety problem, a by-product of the sequence is the establishment of safe, set-valued controllers, see, e.g., (Tomlin et al. 2003; De Santis et al.

2004). Moreover, many connections to Lyapunov theory through the notions of invariance and set contractivity have been identified (Blanchini and Miani 2008; Belta et al. 2017). We also remark that in the literature, different notions of safety/invariance have been developed, depending on the knowledge of the controller of the states, outputs, and exogenous inputs. Moreover, problems closely related to safety, such as the reach-avoid problem, can be addressed by changing the set $\mathcal{C}_0$.

Computations

Reachability analysis produces sequences of sets generated by recursive updates. Thus, it

is crucial to work with set parameterizations that can be described in a computer program with finite memory, and such that all operations described above are practically implementable. Polyhedral sets, especially in cases where linear dynamics and convex constraints are involved, are particularly appealing, since they are closed under Minkowski addition, intersection, convex union, and affine transformation. There is extensive literature tackling a great deal of settings involving linear dynamics and polyhedral sets, including linear parameter varying systems and switching systems. Nevertheless, even for the linear-convex case, computational complexity of the resulting sets can explode, especially for Minkowski sums, projections, minimal representations, and conversions between half-space and vertex representations, see, e.g., Fukuda (2004). A number of works deal with this issue by providing efficient alternatives using, e.g., zonotopes (Girard 2005), support function representations, and other classes of parameterized polytopes/template polyhedra. On the other hand, semialgebraic sets can also be employed for reachability analysis, such as ellipsoids, sublevel sets of SOS polynomials, or polynomials in the Bernstein representation. Moreover, one can leverage combinatorial techniques to reduce computational complexity in reachability analysis (Athanasopoulos and Jungers 2018).

Software

A number of fairly robust software solutions are available, often accompanied by modeling interfaces that allow easy formulation analysis and synthesis problems for hybrid systems and complex specifications. For toolboxes related to reachability analysis, we mention *SpaceEx*, *HyPro*, *CORA*, *d/dT*, and *JuliaReach* and note that the *MPT3* and *PPL* toolboxes are suitable for polyhedral computations.

Lyapunov-Based Optimization Techniques

The development of interior point optimization methods, enabling the efficient resolution of *linear matrix inequalities* (LMIs), has unlocked a huge literature in control. Linear matrix

inequalities are particular algebraic optimization problems involving linear functions of the variables, and these variables can be nonnegative real numbers or positive definite matrices. In the 2000s, it was realized that the ability to efficiently solve such optimization problems actually allows to solve much more general ones, where the unknown variables are multivariate polynomials (of a fixed bounded degree), involving linear functions of these polynomials together with constraints requiring nonnegativity of these polynomials. This is the so-called sum-of-squares (SOS) programming (Parrilo 2000).

This new optimization technique had a great impact on *state-space methods* in control: State-space methods encode the objective function and the constraints directly in terms of the time, and of variables in the state space, by opposition to frequency domain techniques. In view of the intrinsic nonlinear and nonstandard behavior of hybrid systems, it is not surprising that such methods are well-adapted to solve control problems for hybrid systems (Parrilo 2000; Coogan and Arcak 2012). Even more, polynomials turn out to be efficient tools for representing *sets of points* in the state space. As a simple example, if, for some given multivariate polynomial $p(x)$, one defines the set

$$\mathcal{X}_p := \{x \in \mathbb{R}^n : p(x) \geq 0\},$$

then one can encode a sufficient condition for a safety constraint of the type $\forall x \in \mathcal{X}_p, Ax \in \mathcal{X}_q = \{x \in \mathbb{R}^n : q(x) \geq 0\}$ in the following formula:

$$\forall x \in \mathbb{R}^n, q(Ax) \geq p(x).$$

The above formula is *LMI-representable*, being composed of linear functions of the polynomials p, q . By combining such constraints, one can encode complicated control objectives in large SOS programs, which are then solved by standard LMI solvers (e.g., MOSEK, SeDuMi, or SDPT3). Last, we should mention the closely related class of barrier methods; see, e.g., Prajna and Jadabaie (2004), Sloth et al. (2012) and Maghenem and Sanfelice (2019).

Technicalities

Classical results from algebraic geometry imply that polynomials are universal, in the sense that they can approximate arbitrarily well any set, at the price of increasing the degree of the polynomial; see Henrion and Korda (2014) and references therein. This directly translates to universality of the SOS approach for representing sets in the state space.

It is important to mention that inequalities of the type $\forall x, p(x) \geq 0$ are in fact *not* efficiently solvable by LMI solvers. Instead, one classically proceeds to a *SOS relaxation*, requiring the weaker condition that the polynomial can be expressed as the sum of squares of other polynomials. This latter condition can be solved efficiently. Even though this relaxation induces conservativeness, one can show that by increasing the degree of the polynomial, the SOS relaxation is asymptotically non-conservative (Fig. 1).

Interior point methods are relatively efficient at solving sum-of-squares programs, as they essentially rely on the Newton method for convex functions optimization and hence have a cubic complexity at every step; see Boyd and Vandenberghe (2004, Section 11.5) for a detailed analysis. However, one should note that in practice, they suffer scalability issues for problems of moderate to large size. Even more, the size of the SOS program depends on the number of monomials of the maximal degree chosen for the unknown polynomials. Since this number of monomials grows exponentially with the degree, in practice, one is bound to limit the degree to a reasonable value (say, from 8 to 14, depending on the problems considered).

Scalability Improvements

Recent work has focused on addressing the scalability issues. Of particular relevance for hybrid systems is the *path-complete approach* which, rather than increasing the degree of the polynomial, makes use of a combinatorial tool in order to enhance the representing power of polynomials by limiting their degree to a small number (Philippe et al. 2019). See Legat et al. (2018) for an application to safety-critical control of hybrid systems. Let us also mention the DSOS

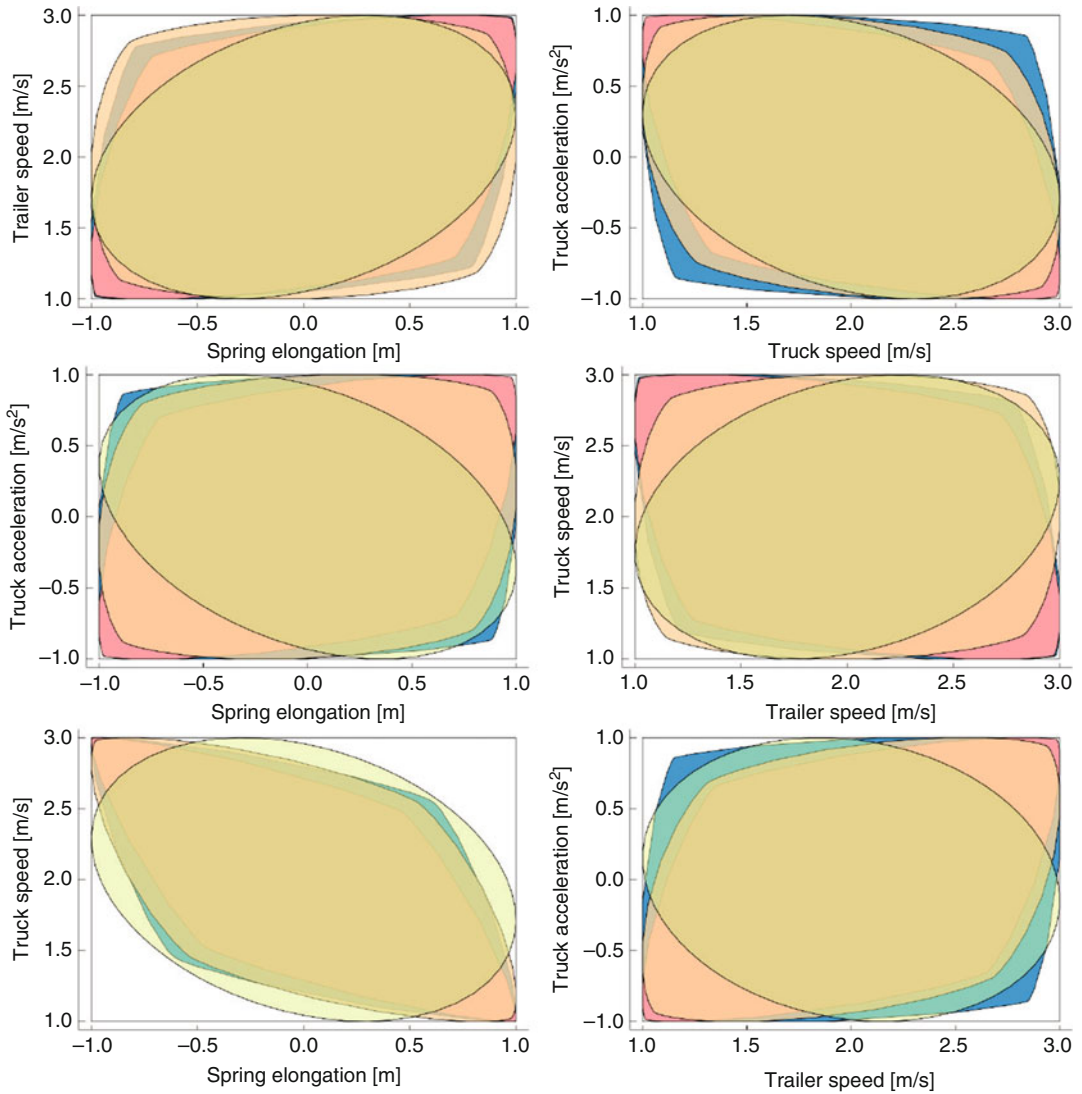
approach (Ahmadi and Majumdar 2014) which restricts SOS problems to particular subfamilies of polynomials, on which solvers perform better.

Software

The past decade has seen the creation of efficient and easy-to-use software for encoding polynomial constraints in a high-level language that are then automatically transformed in LMI programs and solved. See Gloptipoly3, Jump, cvx, SOS-TOOLS, MOSEK, SeDuMi, and SDPT3, along with the parsers *Jump* and *YALMIP*.

Formal Methods and Abstraction

In view of the difficulty and criticality of safety problems, formal approaches have been proposed, influenced by the Computer Science and Model Checking communities. In these approaches, one formulates the safety problem in terms of rigorous predicates in a well-defined syntax corresponding to a particular logic. Typically, *first-order temporal logic* (Pnueli 1977) is well suited to formulate safety problems for dynamical systems, in particular hybrid systems. Then one needs to solve the logical equations formulated to verify safety of a given system or to construct an appropriate controller. To solve such complicated equations, one typically proceeds by *abstraction*, that is, by partitioning the state space into cells and the set of admissible inputs into small intervals, ending up with a finite-state automaton representing the dynamical system. Relying on standard continuity assumptions, many classical control problems (reachability, safety, reference tracking) can then be solved in a finite time, provided that the discrete abstraction satisfies *simulation* properties with respect to the actual system. The simulation property ensures that the behavior of the discrete, abstract system is a truthful representation of the original system. See Alur et al. (2000) for precise definitions. Thanks to the finiteness of the obtained abstraction, the above problems reduce to a simple graph-theoretic one. For example, one can just find (or optimize) a path in the obtained automaton in order to solve the control problem. This confers



Safety Guarantees for Hybrid Systems, Fig. 1

Example of the increasing performance of the SOS technique for a safe set computation problem, when one increases the degree of the unknown polynomial (in yellow for quadratic, orange for quartic, red for sextic,

and blue for octic) (Legat et al. 2018). The problem is concerned with finding the range of safe speeds for a truck with trailers. The pictures are projections of the safe set in the four-dimensional state space, onto coordinate planes

a great advantage to the method in theory. See Haghverdi et al. (2005) and Alur (2011) for good introductions to formal methods.

The formal approach as applied until now suffers from important scalability problems. Indeed, a straightforward discretization of the state space in small cells inevitably leads to a huge set of nodes/edges in the automaton. More precisely, the size of the discretization step must be small, because the method relies on a local property,

namely, continuity; and for a fixed discretization step, the number of cells grows exponentially with the dimension of the state space. For this reason, except for systems of very low dimension (3 or 4), the approach does not work in practice. Nevertheless, these techniques are considered as among the most promising opportunities for complex problems on hybrid systems and are at the center of an intense research effort for the moment.

Software

Several software solutions are being developed and constantly improved, as, for instance, *Pes-soa*, *CoSyMa*, *Tulip*, *SCOTS*, *Hytech*, *HSolver*, *Flow**, *PHAVer*, and *PHAVerLite* for general systems, *Uppaal*, *Kronos* for timed-automata oriented tools, and *Stochy* or *Prism* for probabilistic hybrid systems.

Probabilistic Techniques

Many models of hybrid systems are probabilistic. The techniques above can be adapted to such models, while specialized techniques have also been developed for the analysis of probabilistic hybrid systems; see Katoen (2016).

This subsection is not about such systems. Here we refer to probabilistic *techniques* that provide safety guarantees on either deterministic or probabilistic hybrid systems. Such techniques have recently received increasing attention as alternatives to the aforementioned methods, and the reason is obvious; due to the extreme computational cost, one may want to resort to probabilistic, Monte Carlo-like approaches to tackle the safety analysis problem while controlling at the same time the computational cost. In the community of systems and control, probabilistic techniques like the *scenario approach* (Calafiore and Campi 2006) are taking increasing importance. To our knowledge, the application of such approaches to hybrid systems has remained embryonic until now, but we expect a fast-growing body of work along these lines in the hybrid systems literature in the forthcoming years. Among such attempts, let us mention (Kenanian et al. 2019), which provides theoretical guarantees on probabilistic stability analysis for the particular case of switching systems, and (Julius and Pappas 2008) which makes use of stochastic bisimulation functions to provide a partial verification for more general hybrid systems.

In view of the intrinsic difficulty of coming with firm safety guarantees with probabilistic techniques, the state of the art of software implementation is less mature too. Let us mention S-Taliro which uses several randomized techniques for critical trajectory generation.

Summary and Future Directions

The problem of providing safety guarantees on a control system, called verification or model-checking in the Computer Science literature, is a paradigmatic problem that has been of paramount importance in applications for a long time. Today, the complexification of (data-driven, embedded, networked, etc.) control systems to hybrid systems makes it a challenge more than ever. It also makes it more important than ever, in view of the many emerging control systems interacting with our daily lives. In view of the theoretical computational barriers (undecidability, NP hardness, and others), one cannot hope that a single off-the-shelf technique could solve the problem, and as of today, ad hoc techniques have to be developed in order to cope with, and leverage, the specificities of the particular control problem at stake. Nevertheless, engineers have at their disposal several sound theoretical approaches, with solid software toolboxes, which can provide easily implementable solutions on simple models. We believe that in the future, these solutions will keep improving in performance and generality, as testified by the constant improvement on the state of the art over the past decades.

Cross-References

- [Discrete Event Systems and Hybrid Systems, Connections Between](#)
- [Hybrid Dynamical Systems, Feedback Control of](#)
- [Stability Theory for Hybrid Dynamical Systems](#)

Bibliography

- Ahmadi AA, Majumdar A (2014) DSOS and SDSOS optimization: LP and SOCP-based alternatives to sum of squares optimization. In: 2014 48th annual conference on information sciences and systems (CISS). IEEE, pp 1–5
- Alur R (2011) Formal verification of hybrid systems. In: 2011 proceedings of the ninth ACM international conference on embedded software (EMSOFT). IEEE, pp 273–278

- Alur R, Dill DL (1994) A theory of timed automata. *Theor Comput Sci* 126(2):183–235
- Alur R, Henzinger TA, Lafferriere G, Pappas GJ (2000) Discrete abstractions of hybrid systems. *Proc IEEE* 88(7):971–984
- Athanasopoulos N, Jungers RM (2018) Combinatorial methods for invariance and safety of hybrid systems. *Automatica* 98:130–140
- Aubin JP, Bayen AM, Saint-Pierre P (2011) *Viability theory: new directions*. Springer, Heidelberg/Dordrecht/London/New York
- Belta C, Yordanov B, Gol EA (2017) Formal methods for discrete-time dynamical systems. In: *Studies in systems, decision and control*. Springer, Heidelberg
- Blanchini F, Miani S (2008) *Set-theoretic methods in control*. Birkhäuser, Boston
- Boyd S, Vandenberghe L (2004) *Convex optimization*. Cambridge University Press, Cambridge
- Calafiore GC, Campi MC (2006) The scenario approach to robust control design. *IEEE Trans Autom Control* 51:742–753
- Coogan S, Arcak M (2012) Guard synthesis for safety of hybrid systems using sum of squares programming. In: *2012 IEEE 51st annual conference on decision and control (CDC)*. IEEE, pp 6138–6143
- De Santis E, Di Benedetto MD, Berardi L (2004) Computation of maximal safe sets for switching systems. *IEEE Trans Autom Control* 49(2):184–195
- Doyen L, Frehse G, Pappas GJ, Platzer A (2018) Verification of hybrid systems. In: *Handbook of model checking*. Springer, Cham, pp 1047–1110
- Fukuda K (2004) *Frequently asked questions in polyhedral computation*. Technical report, Swiss Federal Institute of Technology
- Girard A (2005) Reachability of uncertain linear systems using zonotopes. In: *Proceedings of the international workshop on hybrid systems: computation and control*, pp 291–305
- Goebel R, Sanfelice RG, Teel AR (2012) *Hybrid dynamical systems: modeling stability, and robustness*. Princeton University Press, Princeton
- Haghverdi E, Tabuada P, Pappas GJ (2005) Bisimulation relations for dynamical, control, and hybrid systems. *Theor Comput Sci* 342(2–3):229–261
- Heemels WP, De Schutter B, Bemporad A (2001) Equivalence of hybrid dynamical models. *Automatica* 37:1085–1091
- Henrion D, Korda M (2014) Convex computation of the region of attraction of polynomial control systems. *IEEE Trans Autom Control* 59(2):297–312
- Julius AA, Pappas GJ (2008) Probabilistic testing for stochastic hybrid systems. In: *2008 47th IEEE conference on decision and control*. IEEE, pp 4030–4035
- Jungers R (2009) *The joint spectral radius: theory and applications*, vol 385. Springer Science & Business Media, Berlin
- Katoen JP (2016) The probabilistic model checking landscape. In: *Proceedings of the 31st annual ACM/IEEE symposium on logic in computer science*. ACM, pp 31–45
- Kenanian J, Balkan A, Jungers RM, Tabuada P (2019) Data driven stability analysis of black-box switched linear systems. *Automatica*, 109, p. 108533
- Legat B, Tabuada P, Jungers RM (2018) Computing controlled invariant sets for hybrid systems with applications to model-predictive control. arxiv preprint: <https://arxiv.org/abs/180204522>
- Maghenem M, Sanfelice RG (2019) Characterizations of safety in hybrid inclusions via barrier functions. In: *22nd ACM international workshop on hybrid systems: computation and control*, pp 109–118
- Parrilo PA (2000) *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. Ph.D. thesis, California Institute of Technology
- Philippe M, Athanasopoulos N, Angeli D, Jungers RM (2019) On path-complete lyapunov functions: geometry and comparison. *IEEE Trans Autom Control* 64:1947–1957
- Pnueli A (1977) The temporal logic of programs. In: *18th annual symposium on foundations of computer science (sfcs 1977)*. IEEE, pp 46–57
- Prajna S, Jadbabaie A (2004) Safety verification of hybrid systems using barrier certificates. In: *International workshop on hybrid systems: computation and control*. Springer, Berlin, pp 477–492
- Sloth C, Pappas GJ, Wisniewski R (2012) Compositional safety analysis using barrier certificates. In: *International workshop on hybrid systems: computation and control*, pp 15–24
- Tomlin CJ, Mitchell I, Bayen AM, Oishi M (2003) Computational techniques for the verification of hybrid systems. *Proc IEEE* 91(7):986–1001

Sampled-Data H-Infinity Optimization

Tongwen Chen

Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada

Abstract

$\mathcal{H}_\infty$ optimization is central in robust control. When controllers are implemented by computers, sampled-data control systems arise. Designing $\mathcal{H}_\infty$ -optimal controllers in purely continuous time or in purely discrete time is standard in robust control; in this entry, we discuss the process of sampled-

data optimization, namely, designing digital controllers based on a continuous-time $\mathcal{H}_\infty$ performance measure.

Keywords

Computer control · $\mathcal{H}_\infty$ discretization · Robust control · Sampled-data systems

Introduction

Robust control deals mainly with controller design against uncertainties in system modeling and disturbances. The central tool used is $\mathcal{H}_\infty$ optimization.

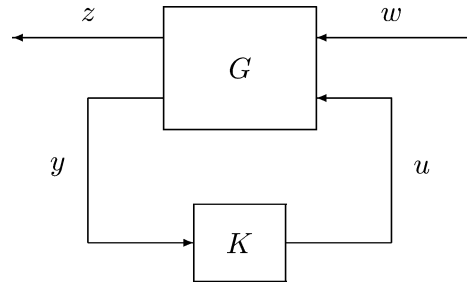
In continuous time, consider the standard setup in Fig. 1, where G is the generalized plant and K is the controller; G has two inputs (w , the exogenous input, and u , the control input) and two outputs (z , the output to be controlled, and y , the measured output); K processes y to generate u . The $\mathcal{H}_\infty$ -optimal control problem is to design K to stabilize G and minimize the $\mathcal{H}_\infty$ norm of the closed-loop system in Fig. 1 from w to z , denoted T_{zw} . When both G and K are continuous-time, linear time-invariant (LTI), the $\mathcal{H}_\infty$ norm, $\|T_{zw}\|$, relates to the frequency response matrix $\hat{T}_{zw}(j\omega)$ as follows:

$$\|T_{zw}\| = \sup_{\omega} \bar{\sigma} [\hat{T}_{zw}(j\omega)],$$

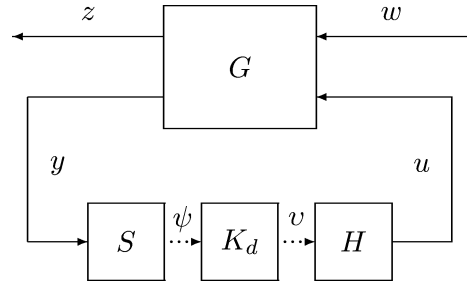
where $\bar{\sigma}$ indicates the maximum singular value. This $\mathcal{H}_\infty$ -optimal control problem in the LTI case is solvable by many techniques, e.g., Riccati equations and linear matrix inequalities – see robust control textbooks by Zhou et al. (1996) and Dullerud and Paganini (2000).

Sampled-Data Control

When controllers are implemented by digital computers, periodic samplers and zero-order holds are used to model analog-to-digital and digital-to-analog conversion. Replacing K in Fig. 1 by sampler S (with period h), discrete-time controller K_d , and zero-order hold H



Sampled-Data H-Infinity Optimization, Fig. 1
Standard control setup in continuous time



Sampled-Data H-Infinity Optimization, Fig. 2
Sampled-data control setup

(synchronized with S), we obtain a sampled-data control system shown in Fig. 2; here, S converts y into a discrete-time sequence ψ ; K_d , a real-time algorithm in the computer, inputs ψ and computes another sequence v , which is converted by H into u .

There are in general three approaches to design a digital controller K_d : design a continuous-time controller K and then implement digitally via approximation, discretize the plant and then design K_d in discrete time, and finally, design K_d directly based on continuous-time performance specifications (Chen and Francis 1995). The last approach is followed in the $\mathcal{H}_\infty$ optimization framework.

Sampled-Data $\mathcal{H}_\infty$ Discretization

The sampled-data $\mathcal{H}_\infty$ control problem is to design K_d directly to stabilize G in Fig. 2 and minimize $\|T_{zw}\|$. Notice that even if G is LTI in

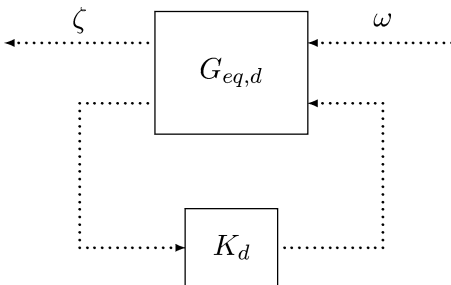
continuous time and K_d is LTI in discrete time, the closed-loop system T_{zw} is no longer LTI, due to the presence of S and H in the control loop; in this case, the $\mathcal{H}_\infty$ norm is interpreted as the $\mathcal{L}_2$ -induced norm:

$$\|T_{zw}\| = \sup\{\|z\|_2 : \|w\|_2 = 1\};$$

here, $\|\cdot\|_2$ represents the $\mathcal{L}_2$ norm on signals.

The sampled-data $\mathcal{H}_\infty$ control problem has been shown to be equivalent to a purely discrete-time $\mathcal{H}_\infty$ control problem (Kabamba and Hara 1993; Bamieh and Pearson 1992; Toivonen 1992); the process is known as sampled-data $\mathcal{H}_\infty$ discretization: for $\gamma > 0$, construct an LTI discrete-time system $G_{eq,d}$ connected to K_d as in Fig. 3; the two systems, T_{zw} in Fig. 2 and $T_{\zeta\omega} : \omega \mapsto \zeta$ in Fig. 3, are *equivalent* in that $\|T_{zw}\| < \gamma$ if $\|T_{\zeta\omega}\| < \gamma$, where the latter norm is ℓ_2 -induced, and since $T_{\zeta\omega}$ is LTI in discrete time, it equals the $\mathcal{H}_\infty$ norm of the corresponding transfer function $\hat{T}_{\zeta\omega}(z)$. Thus, pure discrete-time techniques are immediately applicable.

There are several ways to present this discretization. However, the computation is quite involved and hence is not given here; interested readers can find details in the papers by Kabamba and Hara (1993), Bamieh and Pearson (1992), and Toivonen (1992), or the book by Chen and Francis (1995). Note that the $\mathcal{H}_\infty$ discretization process is not quite *exact* in the sense that $G_{eq,d}$ depends on γ (Chen and Francis 1995).



Sampled-Data H-Infinity Optimization, Fig. 3 The equivalent discrete-time system

Summary and Future Directions

In sampled-data $\mathcal{H}_\infty$ optimization, the key idea is to address the hybrid nature of the problem, considering intersample behavior in formulation; the main tool is the so-called continuous lifting (Yamamoto 1994; Bamieh and Pearson 1992), making use of periodicity of sampled-data systems.

The ideas and tools developed in sampled-data control theory are still being used in emerging areas such as hybrid systems and networked control systems. For example, in event-triggered control systems, information exchange and control updating are not time driven but are done by certain event-triggering schemes, resulting in necessarily nonlinear and time-varying closed-loop dynamics; the analysis and synthesis issues in such systems are still challenging.

Cross-References

- [H-Infinity Control](#)
- [LMI Approach to Robust Control](#)
- [Optimal Sampled-Data Control](#)
- [Optimization-Based Robust Control](#)

Recommended Reading

The continuous-time $\mathcal{H}_\infty$ control problem and its solutions are discussed extensively in several textbooks, e.g., Zhou et al. (1996) and Dullerud and Paganini (2000). The discrete-time $\mathcal{H}_\infty$ control problem was solved via the approach of Riccati equations in Iglesias and Glover (1991). The sampled-data $\mathcal{H}_\infty$ control problem was solved simultaneously with different methods in Kabamba and Hara (1993), Bamieh and Pearson (1992), and Toivonen (1992); details of the solution discussed here can be found in the book by Chen and Francis (1995).

Bibliography

- Bamieh B, Pearson JB (1992) A general framework for linear periodic systems with application to $\mathcal{H}_\infty$ sampled-data control. *IEEE Trans Autom Control* 37:413–435

- Chen T, Francis BA (1995) Optimal sampled-data control systems. Springer, London
- Dullerud GE, Paganini F (2000) A course in robust control theory: a convex approach. Springer, New York
- Iglesias P, Glover K (1991) State-space approach to discrete-time $\mathcal{H}_\infty$ control. Int J Control 54:1031–1073
- Kabamba PT, Hara S (1993) Worst case analysis and design of sampled-data control systems. IEEE Trans Autom Control 38:1337–1357
- Toivonen HT (1992) Sampled-data control of continuous-time systems with an $\mathcal{H}_\infty$ optimality criterion. Automatica 28:45–54
- Yamamoto Y (1994) A function space approach to sampled data control systems and tracking problems. IEEE Trans Autom Control 39:703–713
- Zhou K, Doyle J, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River, New Jersey

Sampled-Data Systems

Panos J. Antsaklis¹ and H.L. Trentelman²

¹Department of Electrical Engineering,
University of Notre Dame, Notre Dame,
IN, USA

²Johann Bernoulli Institute for Mathematics and
Computer Science, University of Groningen,
Groningen, AV, The Netherlands

Abstract

For digital devices to interact with the physical world, an interface is needed that transforms the signals from analog to digital and vice versa. Ideal samplers and zero-order hold devices are incorporated to derive discrete-time models of continuous-time systems. State variable descriptions and transfer functions are used.

Keywords

Continuous-time approximations · Digital control · Discrete-time approximations · Quantization · Reconstruction · Sampled-data systems · Sampling

Introduction

Sampled-data systems are discrete-time models of continuous-time processes useful in the digital

control of continuous-time systems. A digital controller cannot communicate directly with a continuous system and an interface is needed.

Consider a continuous-time system having $u(t)$ as its input and $y(t)$ as its output.

A/D Converter: The continuous-time signal $y(t)$ is converted into a discrete-time signal $\{\bar{y}(k)\}$, $k \geq 0$, $k \in \mathbb{Z}$, which is a sequence of values $\{\bar{y}(0), \bar{y}(1), \dots\}$ determined by the relation

$$\bar{y}(k) = y(t_k). \quad (1)$$

This is the ideal A/D (analog to digital) converter that samples $y(t)$ at times $t_0, t_1, t_2 \dots$ producing the sequence $\{y(t_0), y(t_1), \dots\}$ also denoted as $\{y(t_k)\}$.

D/A Converter: The D/A (digital to analog) converter receives as its input a sequence $\{\bar{u}(k)\}$, $k = 0, 1, 2, \dots$ and outputs a (piecewise) continuous-time signal $u(t)$ determined by

$$u(t) = \bar{u}(k), \quad t_k \leq t < t_{k+1}, \quad k = 0, 1, 2, \dots \quad (2)$$

That is, this D/A converter keeps the value of $u(t)$ constant at the last value of the sequence entered, until a new value comes in. Such a device is called a zero-order hold (ZOH) device.

Higher-Order Hold

The ZOH device described above implements a particular procedure of *data reconstruction or extrapolation*. The general problem is as follows:

Given a sequence of real numbers $\{\bar{f}(k)\}$, $k = k_0, k_0 + 1, \dots$ derive $f(t)$, $t \geq t_0$ so that

$$f(t_k) = \bar{f}(k), \quad k = k_0, k_0 + 1, \dots$$

Clearly, there is a lot of flexibility in assigning values to $f(t)$ in between the samples $\bar{f}(k)$; in other words there is a lot of flexibility in assigning the *intersample behavior* in $f(t)$.

A way to approach the problem is to start by writing a power series expansion of $f(t)$ for t , $t_k \leq t < t_{k+1}$, namely,

$$f(t) = f(t_k) + f^{(1)}(t_k)(t - t_k) + \frac{f^{(2)}(t_k)}{2!}(t - t_k)^2 + \dots$$

where $f^{(n)}(t_k) = \frac{d^n f(t)}{dt^n} \big|_{t=t_k}$, that is, the n th order derivative of $f(t)$ evaluated at $t = t_k$ (assuming that the derivatives exist).

Now if the function $f(t)$ is approximated in the interval $t_k \leq t < t_{k+1}$ by the constant value $f(t_k)$ taken to be equal to $\bar{f}(k)$, then

$$f(t) = f(t_k) \quad (= \bar{f}(k)), \quad t_k \leq t < t_{k+1}$$

which is exactly the relation implemented by a ZOH. Note that here the zero-order derivative of the power series is used which leads to an approximation by a constant which is a zero-degree polynomial.

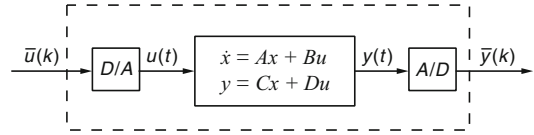
It is clear that more than the first term in the power series can be taken to approximate $f(t)$. If, for example, the first two terms are taken, then

$$\begin{aligned} f(t) &= f(t_k) + f^{(1)}(t_k)(t - t_k) \\ &= f(t_k) + \frac{f(t_k) - f(t_{k-1})}{t_k - t_{k-1}}(t - t_k) \\ &= \bar{f}(k) + \frac{\bar{f}(k) - \bar{f}(k-1)}{t_k - t_{k-1}}(t - t_k) \end{aligned}$$

for $t_k \leq t < t_{k+1}$, where an approximation for the derivative $f^{(1)}(t)$ has been used. The approximation between t_k and t_{k+1} is a ramp with slope determined by $f(t_k) = \bar{f}(k)$ and the previous value $f(t_{k-1}) = \bar{f}(k-1)$. Here the first-order derivative of the power series is used which leads to an approximation by a first-degree polynomial. A device that implements such approximation is called a *first-order hold (FOH)*. Similarly, we can define a *second-order hold*. Note that the formula of the above FOH is derived if we decide to use a first-degree polynomial to approximate $f(t)$ on $t_k \leq t < t_{k+1}$ and then enforce $f(t_k) = \bar{f}(k)$ and $f(t_{k-1}) = \bar{f}(k-1)$. This approach is known as *polynomial interpolation*.

Obtaining a continuous (or piecewise continuous) function from given discrete values may be seen as a *continualization procedure*. Contrast

this with the *discretization procedure* introduced by sampling earlier in this section.



The continuous-time system with input $u(t)$ and output $y(t)$ together with the interface A/D and D/A converters can be seen as a system that receives a sequence of values $\{\bar{u}(k)\}$ as its input and produces a sequence of output values $\{\bar{y}(k)\}$. A digital controller can receive the system output $\{\bar{y}(k)\}$ as input and produce a $\{\bar{u}(k)\}$.

Quantization: The sampled output $\bar{y}(k) \in \mathbb{R}$ and it can take on an infinite number of values. In a digital device, however, a variable can take on only a finite number of values – this is because of the finite wordlength that is of the finite number of bits in the registers. So for $\{\bar{y}(k)\}$ to be used by a digital controller, an additional step is needed, that is, $\bar{y}(k)$ needs to be quantized. Under quantization, for example, values 2.315, 2.308, 2.3 with a 0.1 quantization step are all represented as 2.3. Quantization is an approximation and for short wordlengths, fewer number of levels, may lead to significant errors. Here we do not consider quantization.

Discrete-Time Models

Let a linear, continuous-time, time-invariant system be described by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t). \end{aligned} \quad (3)$$

If we consider some initial time t_k , its state response for $t \geq t_k$ is

$$x(t) = e^{A(t-t_k)}x(t_k) + \int_{t_k}^t e^{A(t-\tau)}Bu(\tau)d\tau. \quad (4)$$

In view of (2), in a ZOH the input $u(t)$ will remain constant and equal to $u(t_k) (= \bar{u}(k))$ for a time period $t_{k+1} - t_k$. So

$$x(t) = e^{A(t-t_k)}\bar{x}(k) + \left[\int_{t_k}^t e^{A(t-\tau)} B d\tau \right] \bar{u}(k), \quad (5)$$

where $\bar{x}(k) = x(t_k)$, $\bar{u}(k) = u(t_k)$. For $t = t_{k+1}$, (5) becomes

$$\bar{x}(k+1) = \bar{A}(k)\bar{x}(k) + \bar{B}(k)\bar{u}(k) \quad (6)$$

where $\bar{A}(k) \triangleq e^{A(t_{k+1}-t_k)}$ and $\bar{B}(k) \triangleq \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} B d\tau$.

Consider now the output $y(t)$ and assume that it is sampled at times t'_k that do not necessarily coincide with the instants t_k at which the input is adjusted ($t_k \leq t'_k < t_{k+1}$). Then if $\bar{y}(k) \triangleq y(t'_k)$,

$$\bar{y}(k) = \bar{C}(k)\bar{x}(k) + \bar{D}(k)\bar{u}(k), \quad (7)$$

where

$$\begin{aligned} \bar{C}(k) &= C e^{A(t'_k - t_k)} \\ \bar{D}(k) &= C \left[\int_{t_k}^{t'_k} e^{A(t'_k - \tau)} d\tau \right] B + D. \end{aligned}$$

In the case when all $k = 0, 1, 2, \dots$, $t'_k = t_k$ and $t_{k+1} - t_k = T$ a constant period, called the *sampling period*. Then the sampled-data system is given by

$$\begin{aligned} \bar{x}(k+1) &= \bar{A}\bar{x}(k) + \bar{B}\bar{u}(k) \\ \bar{y}(k) &= \bar{C}\bar{x}(k) + \bar{D}\bar{u}(k) \end{aligned} \quad (8)$$

where

$$\begin{aligned} \bar{A} &= e^{AT}, \quad \bar{B} = \left[\int_0^T e^{A\tau} d\tau \right] B, \\ \bar{C} &= C, \quad \bar{D} = D. \end{aligned}$$

The intersample behavior of the continuous system can be determined using (5).

Example Let the continuous-time system be given by (3) where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = [1 \quad 0], D = 0,$$

and let T denote the sampling period. The transfer function of the continuous-time system is $\hat{H}(s) = C(sI - A)^{-1}B = 1/s^2$, the double integrator. The discrete-time state-space representation of the system, which represents the continuous-time system preceded by a zero-order hold (D/A converter) and followed by a sampler [an (ideal) A/D converter], both sampling synchronously at a rate of $1/T$, is given by $\bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{B}\bar{u}(k)$, $\bar{y}(k) = \bar{C}\bar{x}(k)$, where

$$\begin{aligned} \bar{A} &= e^{AT} = \sum_{j=1}^{\infty} (T^j/j!) A^j = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} T = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \\ \bar{B} &= \left(\int_0^T e^{A\tau} d\tau \right) B \\ &= \left(\int_0^T \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} d\tau \right) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} T & T^2/2 \\ 0 & T \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}, \\ \bar{C} &= C = [1 \quad 0]. \end{aligned}$$

The transfer function (relating $\bar{y}$ to $\bar{u}$) is given by

$$\begin{aligned} \hat{H}(z) &= \bar{C}(zI - \bar{A})^{-1}\bar{B} \\ &= [1 \ 0] \begin{bmatrix} z-1 & -T \\ 0 & z-1 \end{bmatrix}^{-1} \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \\ &= [1 \ 0] \begin{bmatrix} 1/(z-1) & T/(z-1)^2 \\ 0 & 1/(z-1) \end{bmatrix} \\ &\quad \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \\ &= \frac{T^2}{2} \frac{(z+1)}{(z-1)^2}. \end{aligned}$$

If we focus on single-input, single-output systems and consider ideal sampler A/D and ZOH D/A, then given the transfer function $G(s)$ of the continuous system, there is a direct formula to determine the transfer function of its discrete approximation $H(z)$, namely,

$$H(z) = (1 - z^{-1})Z\{G(s)/s\}. \quad (9)$$

Here $Z\{G(s)/s\}$ means that first the inverse Laplace transform of $G(s)/s$ is taken to obtain $f(t) \triangleq [\mathcal{L}^{-1}(G(s)/s)]$. The function $f(t)$ is then sampled to obtain $f(kT)$, $k = 0, 1, 2, \dots$ and the z -transform of $f(kT)$ is evaluated. To illustrate, in the above example $G(s) = \frac{1}{s^2}$, $G(s)/s = \frac{1}{s^3}$, and $f(t) = \mathcal{L}^{-1}(\frac{1}{s^3}) = \frac{1}{2}t^2$, $t \geq 0$. Then

$$\begin{aligned} H(z) &= (1 - z^{-1})Z\{\frac{1}{2}(kT)^2\} \\ &= (1 - z^{-1})\frac{T^2}{2}Z\{k^2\} \\ &= \frac{T^2}{2} \frac{z+1}{(z-1)^3} \end{aligned}$$

as before.

Summary

Sampled-data systems arise in the digital control of systems and include both continuous and discrete-time dynamics. Discrete-time approximations of continuous-time systems using ideal samplers and ZOH devices were derived using state variable descriptions. Extensions include quantization and lead to hybrid dynamical systems which include both continuous and discrete variable dynamics.

A variation of the approach described in this entry of deriving sampled-data systems uses the discrete-time delta operator. This approach has the advantage that as the sampling period $T \rightarrow 0$, the discrete-time model reverts to the original continuous-time model, which is not the case with the more common approach described above.

Cross-References

- [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)
- [Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions](#)

Recommended Reading

State variable and transfer function descriptions are covered in a variety of textbooks including Antsaklis and Michel (2006), Kailath (1980), Chen (1984), and DeCarlo (1989). For additional material on sampled-data systems, refer to Aström and Wittenmark (1990), Franklin et al. (1998), Jury (1958), and Ragazzini and Franklin (1958).

Bibliography

- Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston
- Aström KJ, Wittenmark B (1990) Computer-controlled systems: theory and design. Prentice-Hall, Englewood Cliffs
- Chen CT (1984) Linear system theory and design. Holt, Rinehart and Winston, New York
- DeCarlo RA (1989) Linear systems. Prentice-Hall, Englewood Cliffs
- Franklin GF, Powell DJ, Workma ML (1998) Digital control of dynamic systems, 3rd edn. Addison-Wesley Longman Inc., Menlo Park, CA
- Jury EI (1958) Sampled-data control systems. Wiley, New York
- Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs
- Ragazzini JR, Franklin GF (1958) Sampled-data control systems. McGraw-Hill, New York
- Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs

Satellite Control

Finn Ankersen
European Space Agency, Noordwijk,
The Netherlands

Abstract

Spacecraft control systems are described for single and distributed space systems. The attitude dynamics is formulated including flexible and sloshing phenomena, followed by a description of attitude sensors and actuators. $\mathcal{H}_\infty$ and robust controls are formulated as

signal-based two degree-of-freedom control architectures. The equations are given for the relative motion dynamics between spacecraft on elliptical orbits with the generic Yamanaka-Ankersen state transition matrix. Formulations are provided for rendezvous and docking scenarios and formation flying control, maneuvers, avionics, and laser metrology systems together with the onboard autonomy needs.

Keywords

Flexible modes · Formation flying · Fractionated spacecraft · $\mathcal{H}_\infty$ control · Multivariable systems · Relative dynamics · Rendezvous and docking · Robust control · Sloshing · Spacecraft attitude control · Spacecraft position control

Introduction

This entry explains the control needs of spacecraft after they have been separated from the launch vehicle and injected onto their initial orbit.

Actuators and sensors are explained followed by the control objectives. The state-of-the-art control techniques and architectures are addressed.

Spacecraft are classically well-known physical systems that can be described by first principles. The advantage is fairly precise plant models and uncertainty characterization of physical parameters. This is well suited for a model-based control design approach.

Mission Types

From a control point of view, space missions can be split into two main categories according to which physical states need to be controlled:

Attitude Control: This is needed by any spacecraft irrespective of the mission objectives. Such missions are typically low earth orbit (LEO) missions for astronomy, observations, and, in higher orbits, constellations for navigation and communication. Further, there are

interplanetary and planetary exploration science missions. The pointing requirements vary from a few degrees to milli-arc seconds.

Relative Position Control: Within distributed space systems, this is relevant for rendezvous and docking (RVD) and formation flying (FF) missions. It leads to a 6 degree-of-freedom (DOF) control problem as the relative attitude is also needed. The former is mostly for missions to space station logistics infrastructures and the latter for scientific missions. Relative position can also be required during the final stages of controlled planetary landings. Another category is missions with ultrahigh control performance requirements, where the spacecraft platform and the science instrument need to be considered as one coupled system.

Attitude Control

Fundamentally the three attitude angles θ and angular rates ω need to be controlled to a certain reference. See Fig. 1 for definition.

The general rigid body dynamics expressed in a rotating frame(*), which is mostly the case when orbiting a central body, can be expressed as

$$\mathbf{N} = \frac{d^*(\mathbf{I}\omega^*)}{dt} + \omega \times \mathbf{I}\omega^* \quad (1)$$

where $\mathbf{I}$ is the constant inertia matrix, ω is the inertial angular velocity, and $\mathbf{N}$ is the torque acting on the spacecraft (Wie 1998).

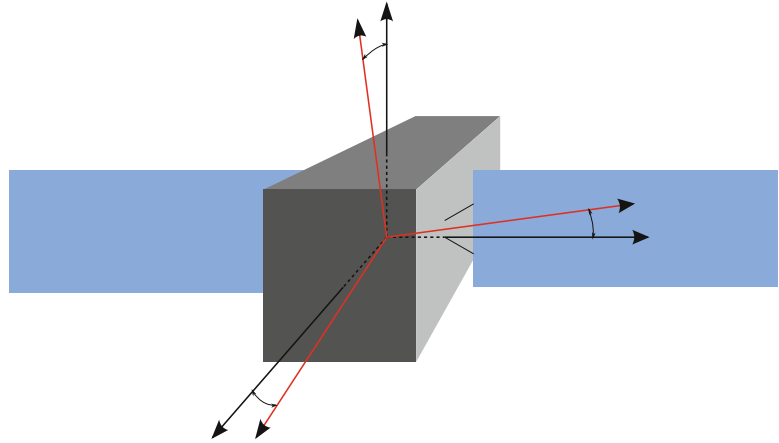
The kinematics can be described by one of the 12 sets of Euler angles (can have singularities) or the hypercomplex quaternion vector (no singularities) (Hughes 1986).

The dynamics and kinematics equations need to be linearized and are in the general form of a coupled 12th order system. It is the fundamental model for the rigid body spacecraft control design.

Most modern spacecraft have large flexible appendices in the form of solar panels and large antennae reflectors. Fuel sloshing is a similar lightly damped oscillatory phenomena, which often needs to be taken into consideration.

Satellite Control, Fig. 1

Spacecraft body (*black*)
and reference (*red*) frames.
The frames coincide for
 $\theta = 0$



The incorporation of dynamic elements such as flexible panels, antennae, and sloshing fuel can be modeled by Eqs. (2) and (3) provided the overall rotation rate $\dot{\omega}$ and linear accelerations $\ddot{\mathbf{x}}$ are not too large.

$$\mathbf{M}_T \begin{bmatrix} \ddot{\mathbf{x}} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{N} \end{bmatrix} - \mathbf{L}\ddot{\eta} \quad (2)$$

$$\ddot{\eta}_k + 2\zeta_k \Omega_k \dot{\eta}_k + \Omega_k^2 \eta_k = -\frac{1}{m_k} \mathbf{L}^T \begin{bmatrix} \ddot{\mathbf{x}} \\ \dot{\omega} \end{bmatrix} \quad (3)$$

where

- $\mathbf{M}_T$: rigid body mass/inertia matrix
- $\ddot{\mathbf{x}}, \dot{\omega}$: linear and angular acceleration
- $\mathbf{F}, \mathbf{N}$: forces and torques on the spacecraft
- η_k : the k th flexible state
- ζ_k : the k th flexible damping factor
- Ω_k : the k th flexible eigen frequency
- m_k : the k th modal mass (normalized to 1)
- $\mathbf{L}$: participation matrix of the k th mode

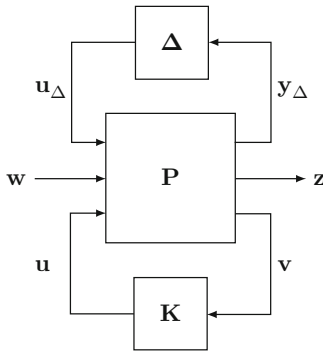
For attitude only the second row of Eq. (2) is needed, but translation is included here for the sake of completeness and later use.

The sensors utilized are typically gyroscopes for measuring the inertial angular rate, sun sensors to measure orientation at low accuracy, and star trackers for high-precision angular attitude measurements. All of those sensors are linear in their normal operational range and it suffices to use bias noise models for synthesis. Gyros do need a drift estimation and compensation to

function properly over longer time. All sensors utilize redundancy for providing measurements around all three axes as well as providing fault tolerance. Some scientific observatory spacecraft use their telescopes for attitude measurements in order to obtain the required precision beyond the capability of star trackers.

The actuators producing pure torques are magnetic torquers, reaction wheels, and control momentum gyros. The last can produce large torques used for rapid slew maneuvers with little power. The last two types have nonlinear issues around low to zero speed due to friction issues. They accumulate angular momentum from asymmetric disturbances. This leads to a need for thrusters for angular momentum off-loading. Thrusters are also used to control the attitude directly on many spacecraft. They are mostly of on-off type, though continuous ones exist, and will need to be pulse width modulated (PWM) to obtain quasi-linear behavior. The nonlinear on-off nature needs to be taken into account for the control closed loop analysis. It is done by use of the negative inverse describing function (Ogata 1970) for stability analysis and nonlinear modeling for verification simulations in the time domain. For larger numbers of thrusters, an optimization-based selection algorithm is applied to the controller output.

Before using the plant model in Eq. (2) for a flexible spacecraft, a simpler multivariable model of a rigid spacecraft is used as in Eq. (4):



Satellite Control, Fig. 2 Robust control formulation, where Δ is the structured uncertainty, K is the controller, P the partitioned formulation of the plant with weights, and w and z are exogenous inputs and outputs, respectively

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{0} & \mathbf{B}_k \\ \mathbf{0} & \mathbf{A}_d \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_d \end{bmatrix} \mathbf{N} \quad (4)$$

where $\mathbf{x} = [\theta_x, \theta_y, \theta_z, \omega_x, \omega_y, \omega_z]^T$, $\mathbf{B}_k$ is identity, $\mathbf{B}_d = \mathbf{I}^{-1}$, and $\mathbf{A}_d$ is the general Jacobian for the dynamics having a real right half-plane (RHP) pole. See Ankersen (2011). The model describes the angular deviation from some reference frame, whose orientation can be arbitrary. It uses the Euler (3, 2, 1) rotation in the kinematics.

The state of the art of attitude control is today mostly based on $\mathcal{H}_\infty$ type of robust controllers with synthesis performed in the frequency domain. Requirements are often specified in the time domain, but formal methods exist to transform them into frequency domain weighting functions (ESA Handbook 2011) enhancing both synthesis and analysis. System uncertainties can be formulated as structured linear fractional transformations (LFT) with a general control configuration as illustrated in Fig. 2.

Commonly the $\mathcal{H}_\infty$ controller K is designed, and the lower loop in Fig. 2 is closed via a lower LFT such that $\mathbf{N} = F_l(\mathbf{P}, \mathbf{K})$ and robust stability (RS) and robust performance (RP) analysis is performed on the $\mathbf{N}, \Delta$ system (Skogestad and Postlethwaite 1996).

On high performance pointing spacecraft, active vibration suppression of, e.g., cryocoolers is needed. The implementation of control design and recursive system identification can achieve

significantly better attenuation compared to classical passive isolation techniques.

Lately optimization-based codesign of structures and control has been performed successfully. A joint performance function is formulated (mass, stiffness, pointing, fuel, etc.) and an optimization is performed (differential evolution algorithm) iterating on control design and finite element models (FEM). A μ -synthesis controller is synthesized, the pointing performance is fulfilled, and 15–20 % mass saving is obtained on the flexible structures. The entire process is fully automated (Falcoz et al. 2013).

Relative Position Control

For all distributed space systems, relative dynamics is important. Rendezvous and formation flying missions need tracking or maintenance of the desired relative separation, orientation, and position between or among the spacecraft. This is common and independent of the mission type and will be described in general terms ahead of the specific RVD and FF missions.

The general relative position dynamics between centers of mass (COMs) is in Eq. (5), where it is observed that the in-plane motion (x, z) is decoupled from the out-of-plane motion (y).

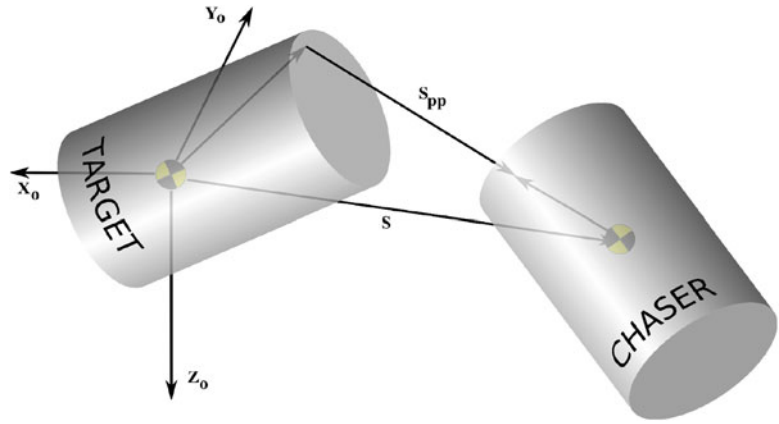
$$\begin{aligned} \ddot{x} - \omega^2 x - 2\omega\dot{z} - \dot{\omega}z + k\omega^{\frac{3}{2}}x &= \frac{1}{m_c} F_x \\ \ddot{y} + k\omega^{\frac{3}{2}}y &= \frac{1}{m_c} F_y \\ \ddot{z} - \omega^2 z + 2\omega\dot{x} + \dot{\omega}x - 2k\omega^{\frac{3}{2}}z &= \frac{1}{m_c} F_z \end{aligned} \quad (5)$$

where $\omega = \omega(t)$ is the orbital angular rate, m_c is the chaser mass, F_{xyz} is the force on the chaser, and k is a constant determined by the orbit and is valid for any Keplerian orbit with eccentricity $\varepsilon < 1$.

The Yamanaka-Ankersen equations (Yamanaka and Ankersen 2002) provide the generalized homogeneous solution in the form of the transition matrix Φ , where the solution can be

Satellite Control,

Fig. 3 Definition of COM-to-COM and port-to-port positions, $\mathbf{s}$ and $\mathbf{s}_{pp}$, respectively, between two spacecraft



written as

$$\mathbf{x}(t) = \mathbf{\Lambda}^{-1}(\nu) \mathbf{\Phi}(\nu) \mathbf{\Phi}_0^{-1}(\nu_0) \mathbf{\Lambda}(\nu_0) \mathbf{x}(t_0) \quad (6)$$

where ν is the orbital true anomaly and $\mathbf{\Lambda}$ are transformation matrices to and from the time domain. The elements of $\mathbf{\Phi}$ in Eq. (6) are detailed in (Ankersen 2011), where relevant particular solutions are also to be found. Equation (6) reduces to the well-known Clohessy-Wiltshire equations for circular orbits ($\varepsilon = 0$) (Clohessy and Wiltshire 1960). Equation (6) is used for feedforward control and trajectory propagation in the guidance function. During the final approach (see Fig. 3), a model accounting for the docking port-to-port relative position and the couplings from the relative attitude to the position is utilized and formulated in Eqs. (7) and (8) (Ankersen 2011):

$$\mathbf{R} = \begin{bmatrix} \mathbf{A}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_c \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{B}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_c \end{bmatrix} \mathbf{u} \quad (7)$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{B}_{dc1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{B}_{dc2} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{x} \quad (8)$$

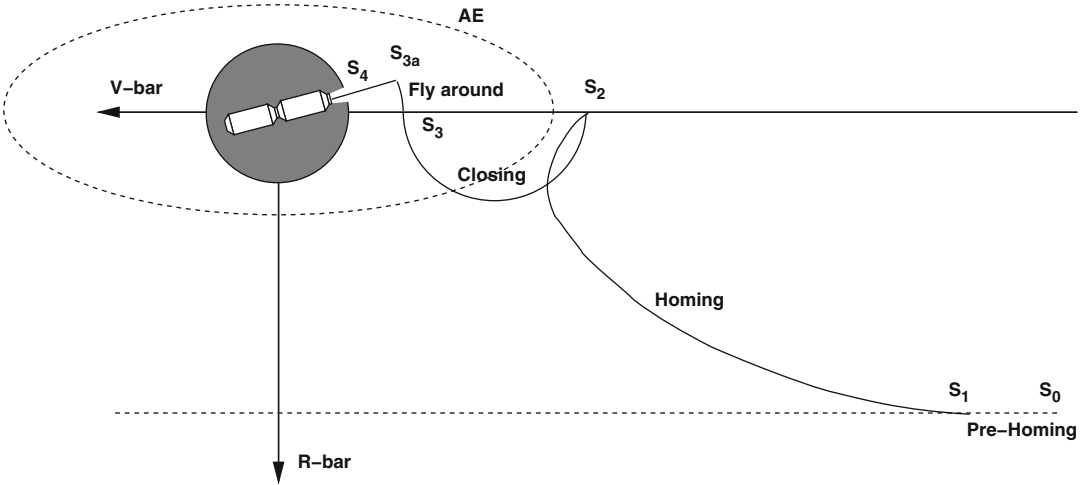
where $\mathbf{x} = [\mathbf{x}_p, \dot{\mathbf{x}}_p, \boldsymbol{\theta}_c, \boldsymbol{\omega}_c]^T$, $\mathbf{y} = [\mathbf{x}_{pp}, \dot{\mathbf{x}}_{pp}, \boldsymbol{\theta}_c, \boldsymbol{\omega}_c]^T$, index p refers to COM positions, index c to chaser attitude, index pp to port-to-port position, and $\mathbf{B}_{dc1}$, $\mathbf{B}_{dc2}$ are the coupling matrices of the docking port.

A relative motion scenario for a typical RVD mission looks like in Fig. 4. During the final approach ($<300\text{m}$ range), the chaser relative attitude and relative position are controlled. During the other phases, the chaser attitude is Earth pointing and the relative position is controlled at the station-keeping (SK) points, $s_0, \dots, s_4$ in Fig. 4. The trajectories are typically open loop feedforward controlled (often with midcourse corrections).

The avionics sensors for the attitude control part are generally similar to those described earlier under attitude control in connection with Fig. 1. Active laser CCD type of sensors is used to measure the relative position (range and line-of-sight (LOS) angles) and at short range ($<50\text{m}$) the relative attitude. They require a target pattern to provide precise measurements at short range. Accelerometers are used, particularly for pulsed maneuvers. The next generation of RVD GNC systems, test flown, will utilize Lidar, infrared cameras, and visual cameras in combination with advanced image processing providing RVD capabilities with both cooperative and passive target spacecraft.

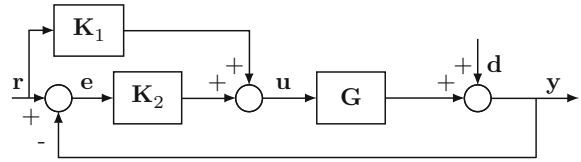
The actuators are mostly thrusters arranged to achieve controllability for all the 6DOF maneuvers needed. Based upon the controller output, the active thrusters are selected by means of some type of fuel optimization algorithm. The selected thrusters are then pulse width modulated (PWM) within the sampling time.

The controllers are frequently of multivariable $\mathcal{H}_\infty$ type. They are similar to what is described



Satellite Control, Fig. 4 This figure shows the phases of typical relative motion approach. The shaded area is a keep-out zone (KOZ) defined for safety reasons. V-bar is the x-axis and R-bar is the z-axis

Satellite Control, Fig. 5 Principal structure of the 2 degree-of-freedom controller



in connection with Fig. 2. Flexible modes and in particular sloshing need to be taken into account using Eq. (2). Sloshing pendulum models are used during boost maneuvers and spring mass damper models during other modes. The couplings between relative attitude and relative position in Eq. (8) can be analytically decoupled setting the matrix $\mathbf{C}$ to identity and premultiplying with a decoupling matrix $\mathbf{V}_d$, such that

$$\mathbf{V}_d \mathbf{C} = \mathbf{I} \Leftrightarrow \mathbf{V}_d = \mathbf{C}^{-1} \quad (9)$$

and by the inversion theorem for partitioned matrices the upper right partition just changes sign. The designed controller then needs to be premultiplied by $\mathbf{V}_d^{-1}$, which facilitates a simpler control design maintaining the 6DOF performance after 2 times 3DOF synthesis.

A 2 degree-of-freedom control architecture as in Fig. 5 is beneficial since much of the performance is achieved by controller $\mathbf{K}_1$. The structure of the synthesis formulation is a signal-based model-reference configuration for the $\mathcal{H}_\infty$

control rather than the more classical mixed sensitivity type. It has proven to have higher robustness and performance for this type of applications. As an example, consider a controller that has to follow a sawtooth motion of the docking port of the International Space Station (ISS) with an amplitude of 0.4 m and reversal times of 8 s. The signal-based model-reference controller manages to track such a motion with errors less than 0.01 m compared to the best operational performance of 0.08 m.

Formation flying usually includes more than two spacecraft with the need to be controlled relative to each other. The objective of FF is to form an instrument in space, not possible with fixed structures, like a synthetic aperture or an interferometer of large size.

The performance needs are high and require innovative high-precision ($<1 \mu\text{m}$) metrology sensors. They are based on divergent laser beams for the coarse part to be able to transit from lower to higher accuracy. The fine metrology uses a laser beam and internal interferometers to reach the μm domain. Actuators are in the

range of μN thrust, which can be achieved with either cold gas or electrical propulsion thrusters.

The maneuvers realized by entire formations are rotation, resizing, and slew while maintaining the formation in most cases (Alfriend et al. 2010).

Formation flying missions with the highest performance requirements have optical payloads, which need to have internal control loops at component level. To reach the performance required for applications such as optical interferometry, the formation and payload must be considered as one system. The synthesis of a multivariable controller then handles all the cross couplings in the system needed to reach performance. Beyond flexible modes, such systems might also have a need for active vibration damping for systems using cryocoolers.

The GNC architecture is often centralized for nominal science operational modes. For the formation deployment and contingency situations, a decentralized control architecture is needed. This leads to a dual architecture GNC system in general for formation flying systems. The onboard autonomy needs to be fairly high in order to cope with the contingencies in the formation without ground intervention.

Finally there is an emerging concept of fractionated spacecraft. There, a formation consists of a large number of small simple vehicles maneuvering relative to each other fully autonomously based upon the nearest neighbor knowledge and not necessarily information about the entire formation (Cornford 2012).

Summary and Future Directions

The control of spacecraft has been described for pure attitude control needs and for spacecraft performing relative proximity maneuvers like rendezvous and formation flying. The focus has been on sensors, actuators, dynamics, and the robust control methods applied today.

The further development direction of the field is expected to be increased on board autonomy with replanning capabilities and fault-tolerant GNC designs. Model predictive control (MPC)

will enter in particular on the guidance functions. More integrated GNC system-level designs, of multidisciplinary nature, are expected.

Cross-References

- [Fault-Tolerant Control](#)
- [H-Infinity Control](#)
- [Model Predictive Control in Practice](#)
- [Nominal Model-Predictive Control](#)

Bibliography

- Alfriend K, Vadali S, Gurfil P, How J, Breger L (2010) *Spacecraft formation flying*. Elsevier, Amsterdam/Boston/London
- Ankersen F (2011) *Guidance, navigation, control and relative dynamics for spacecraft proximity maneuvers*. Aalborg University, Denmark. ISBN:978-87-92328-72-4
- Bryson A (1999) *Control of spacecraft and aircraft*. Princeton University Press, Princeton
- Clohesy W, Wiltshire R (1960) Terminal guidance system for satellite rendezvous. *J Aerosp Sci* 27(9): 653–658
- Cornford S (2012) Evaluating a fractionated spacecraft system: a business case tool for DARPA's F6 program. In: *Aerospace conference, Big Sky*. IEEE, Big Sky, MT, pp 1–20
- D'Errico M (2012) *Distributed space missions for earth system monitoring*. Springer, New York
- ESA Handbook (2011) *ESA pointing error engineering handbook*. European Space Agency. <http://peet.estec.esa.int>
- Falcoz A, Watt M, Yu M, Kron A, Menon P, Bates D, Ankersen F, Massotti L (2013) Integrated control and structure design framework for spacecraft applied to BIOMASS satellite. In: *19th IFAC conference on automatic control in aerospace*, Würzburg, Germany, 2–6 Sept 2013
- Fehse W (2003) *Automated rendezvous and docking of spacecraft*. Cambridge University Press, Cambridge/New York
- Hughes P (1986) *Spacecraft attitude dynamics*. Wiley, New York
- Kaplan M (1976) *Modern spacecraft dynamics & control*. Wiley, New York
- Ogata K (1970) *Modern control engineering*. Prentice-Hall, Englewood Cliffs
- Sidi M (2000) *Spacecraft dynamics and control: a practical engineering approach*. Cambridge University Press, Cambridge
- Skogestad S, Postlethwaite I (1996) *Multivariable feedback control*. Wiley, Chichester/New York

- Wertz J (1980) Spacecraft attitude determination and control. Kluwer, Dordrecht
- Wie B (1998) Space vehicle dynamics and control. American Institute of Aeronautics and Astronautics, Reston
- Yamanaka K, Ankersen F (2002) New state transfer matrix for relative motion on an arbitrary elliptical orbit. *J Guid Control Dyn* 25(1):60–66

Scale-Invariance in Biological Sensing

Eduardo D. Sontag

Departments of Bioengineering and Electrical and Computer Engineering, Northeastern University, Boston, MA, USA

Abstract

The phenomenon of fold-change detection, or scale invariance, is exhibited by a variety of sensory systems, in both bacterial and eukaryotic signaling pathways. This entry gives a short introduction to the subject.

Keywords

Perfect adaptation · Fold-change detection · Scale invariance

Introduction

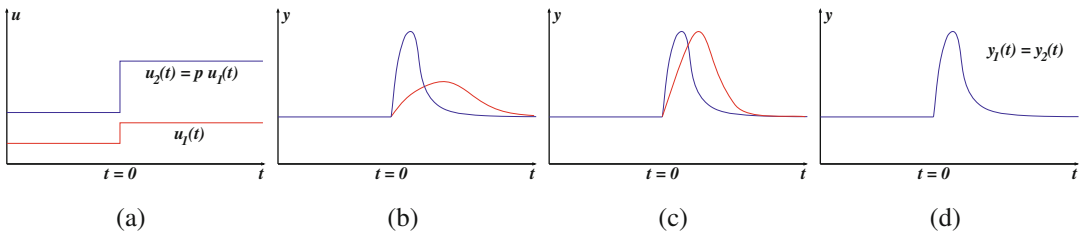
It is often the case that a physiological signal returns to a pre-stimulus value after a transient input has been sensed. This input, for example, an impulse or pulse, could be physical or biochemical in nature. Examples are a ligand to an olfactory receptor or a light input to a photoreceptor. This is a stability property. A much more interesting phenomenon is that of *exact adaptation* to a persistent input, the subject of much investigation from both a modeling and experimental perspective (Alon 2007). This phenomenon occurs when the return to such a steady-state value of the output happens even in the face of a *sustained* step or periodic excitation. Physiological adaptation is a trait of many sensory systems. It allows them

to accurately detect changes in input signals and to distinguish meaningful information from background, by appropriately shifting their dynamic range. For example, the human eye distinguishes features across nine orders of magnitude, even though its sensors can only detect a three order of magnitude contrast; this is achieved through both the pupillary light reflex and the adjustment of sensitivity of rods and cones. Similarly, humans adapt to constant touches, smells, or background noises, detecting new information only when a substantial change occurs. At a cellular scale of behavior, a very well-studied example of physiological adaptation is that of the response of the *E. coli* chemotactic pathway response to stepwise addition and subsequent removal of attractant.

In terms of control theory, perfect adaptation can be thought of as a particular instance of *disturbance rejection*, which for linear systems is translated as zero gain at zero or other frequencies. Disturbance rejection implies, for linear as well as some classes of nonlinear systems, the existence of “internal models” of inputs (Huang et al. 2018). For simplicity, we restrict to step responses in this short discussion, but similar questions can be studied for persistent oscillatory inputs.

Scale Invariance

There is a finer property than mere adaptation, called scale invariance of responses or “fold-change detection”. To explain this property intuitively, consider two step inputs u_1 and u_2 which are scaled versions of each other: $u_2(t) = pu_1(t)$, for some positive number or “scale factor” p , see Fig. 1a. Adaptation means that, whether the input is u_1 or u_2 , the output will return to the same value (Fig. 1b). Scale invariance means that the entire actual transient response will be the same under either excitation (Fig. 1d). An intermediate property between mere adaptation and scale invariance is the “Weber-like” property in which temporal, transient responses may be different but peak intensities coincide (Fig. 1c). This intermediate property is one version of the Weber-Fechner law in psychophysics of human sensing, which relates physical magnitudes of



Scale-Invariance in Biological Sensing, Fig. 1 (a) Scaled step inputs and corresponding responses; (b) perfect adaptation; (c) Weber-like (same peak amplitude responses); (d) scale invariance (same transient responses)

stimuli to perceived intensities. Ernst Weber in the 1840s performed experiments in which subjects were asked to hold a weight, and the weight was gradually increased until a change was first noticed. He found that the smallest noticeable difference was proportional to the starting value, and not to the absolute weight: two scaled versions of the input result in the same reaction. A similar phenomenon has been observed in other sensory systems, including perception of pitch in sound, light intensity, smell, pain, and taste. Gustav Fechner went on to establish a logarithmic relation between physical and perceived quantities (Weber 1905).

A pair of papers published in late 2009 led to a focus on scale invariance in cell biology. These papers dealt with the Wnt and EGF signaling pathways, which are highly conserved eukaryotic signaling pathways that play roles in embryonic patterning, stem cell homeostasis, cell division, and other central processes, the misregulation of which results in diseases including several types of cancer. The paper Goentoro and Kirschner (2009) focused on the effect of binding of Wnt ligand on the levels of a key protein in the Wnt signal transduction pathway, β -catenin, which in turn activates transcription of specific target genes. It was observed that, in a given population, cells might differ substantially in the β -catenin level after stimulation by Wnt but that the effects downstream, measured either through gene expression or phenotype (in *Xenopus* embryos), appear to be a function only of the relative changes in Wnt, and not its absolute amount. Analogous results, for an EGFR pathway, were reported in the paper Cohen-Saidon et al. (2009). Scale invariance is also found in certain bacterial

signaling systems. A prediction, for the *E. coli* chemotaxis sensory circuit in response to the ligand α -methylaspartate, was made in Shoval et al. (2011), based on a model proposed by Tu, Shimizu, and Berg (2008). This prediction was later verified in a microfluidics population experiment carried out in Stocker's lab as well as an in FRET measurements on genetically altered bacteria in Shimizu's lab (Lazova et al. 2011).

Scale invariance means that the system cannot distinguish between an input $u(t)$ and a scaled version $v(t) = pu(t)$. For step inputs that jump at $t = 0$, we can reformulate this property by saying that the response can *only* depend on the “fold change” of the input at time 0: $v(t)/v(0) = pu(t)/pu(0) = u(t)/u(0)$, hence motivating the alternative terminology “fold change detection” (FCD). Another way to phrase this is that FCD systems only depend on $\log(u(t)) - \log(u(0))$, that is, on logarithmic changes in inputs (hence sometimes the use of the term “log sensing”).

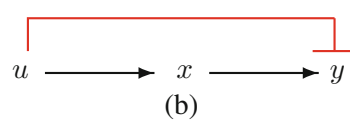
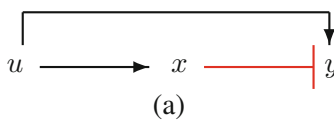
Example: Feedforward Circuits

Let us focus on one rich source of practical examples of FCD systems, feedforward motifs. These circuits, popularized by the textbook Alon (2007), play a central role in metabolic pathways, signaling networks, and genetic circuits. Feedforward loops come in two flavors, coherent and incoherent. The *IFFL* (*incoherent feedforward loop*) motif, represented by the graphs in Fig. 2, is one of the two main biomolecular mechanisms (another is nonlinear integral feedback) that can lead to FCD (Goentoro et al. 2009; Shoval et al. 2010, 2011). In these circuits, the input u activates a regulatory variable (molecular species) x that in turn activates or represses a downstream

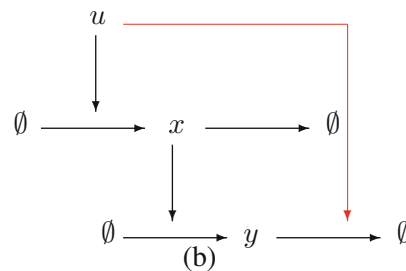
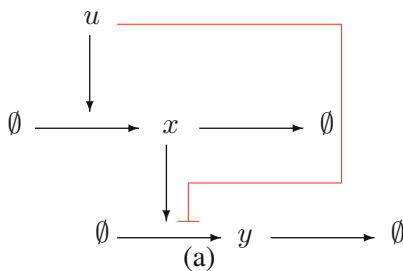
species y . Through a different path, the signal u represses or activates, respectively, the species y . This antagonistic (“incoherent”) effect endows the IFFL motif with powerful signal processing properties (Alon 2007). Similar circuits may play a role in how the immune system distinguishes between “self” and “nonself” antigens (Shoval et al. 2017).

An important subtlety is that the purely conceptual diagrams in Fig. 2 may obscure the fact that alternative molecular realizations may have different properties when viewed in the scale invariance setting. As an illustration, take the two realizations shown in Fig. 3 of the diagram in Fig. 2b. These two realizations differ in a fundamental way in regard to their scale invariance (FCD) properties. The biological mechanism in Fig. 3a exhibits FCD, but the one in Fig. 3b does not. To be more precise, we study the simplest ordinary differential equation (ODE) models for these processes, in which the concentrations of the input u and species x and y are described by scalar time-dependent quantities. Suppose that $(x(t), y(t))$ is any solution corresponding to the input $u(t)$, for the system described by Fig. 3a. Then, $(px(t), y(t))$ is a solution corresponding to the input $pu(t)$:

$$\dot{x} = \alpha u - \delta x \Rightarrow (\dot{px}) = \alpha(pu) - \delta(px)$$



Scale-Invariance in Biological Sensing, Fig. 2 Incoherent feedforward: (a) Input activates and intermediate species represses output; (b) Input represses and intermediate species activates output



Scale-Invariance in Biological Sensing, Fig. 3 Two realizations of the “input repressing output” motif in Fig. 2b: (a) Input inhibits the formation of output; (b) Input enhances the degradation of output

$$\dot{y} = \beta \frac{x}{u} - \gamma y \Rightarrow \dot{y} = \beta \frac{px}{pu} - \gamma y. \quad (1)$$

In particular, given a step input that jumps at time $t = 0$ and an initial state at time $t = 0$ that has been preadapted to the input $u(t)$ for $t < 0$ (that is, $x(0) = \alpha u_0 / \delta$, where u_0 is the value of u for $t < 0$), the solution is the same as when instead applying $pu(t)$ for $t > 0$ but starting from the respective pre-adapted state $p\alpha u_0 / \delta$. On the other hand, the FCD property *fails for the system in which the input enhances the degradation of output*, shown in Fig. 3b. Scaling states x by p does not work for this second system:

$$\begin{aligned} \dot{x} &= \alpha u - \delta x \\ \dot{y} &= \beta x - \gamma u y \end{aligned}$$

since $x \mapsto px$ and $u \mapsto pu$ do not leave the y equation invariant. Actually, one can prove that no possible equivariant group action on states is compatible with output invariance, which means that no possible symmetries are satisfied by the input/output behavior of this system. These issues are carefully discussed in Shoval et al. (2011), which carried out a systematic analysis of the FCD property and proved a necessary and sufficient characterization of FCD in terms of groups of symmetries acting on the system equations.

The above negative remarks notwithstanding, it has been observed that systems such as the one in Fig. 3b satisfy an *approximate* FCD property provided that the parameters β and γ are large enough so that a time-scale separation property holds. Multiple time scales, corresponding to slow and fast subsystems, are typically inherent in cellular systems. See Skataric et al. (2015) for a rigorous discussion of this topic.

Cross-References

- [Deterministic Description of Biochemical Networks](#)
- [Monotone Systems in Biology](#)
- [Reverse Engineering and Feedback Control of Gene Networks](#)
- [Robustness Analysis of Biological Models](#)
- [Spatial Description of Biochemical Networks](#)
- [Stochastic Description of Biochemical Networks](#)

Acknowledgments This work was supported by NSF Grants 1716623 and 1849588 and AFOSR Grant FA9550-14-1-0060.

Bibliography

- Alon U (2007) An introduction to systems biology: design principles of biological circuits. Chapman & Hall/CRC Mathematical & Computational Biology, Boca Raton
- Cohen-Saidon C, Cohen AA, Sigal A, Liron Y, Alon U (2009) Dynamics and variability of ERK2 response to EGF in individual living cells. *Mol Cell* 36:885–893
- Goentoro L, Kirschner MW (2009) Evidence that fold-change, and not absolute level, of β -catenin dictates Wnt signaling. *Mol Cell* 36:872–884
- Goentoro L, Shoval O, Kirschner MW, Alon U (2009) The incoherent feedforward loop can provide fold-change detection in gene regulation. *Mol Cell* 36:894–899
- Huang J, Isidori A, Marconi L, Mischiati M, Sontag ED, Wonham WM (2018) Internal models in control, biology and neuroscience. In: Proceedings of the IEEE conference on decision and control, pp 5370–5390
- Lazova MD, Ahmed T, Bellomo D, Stocker R, Shimizu TS (2011) Response rescaling in bacterial chemotaxis. *Proc Natl Acad Sci USA* 108:13870–13875
- Shoval O, Goentoro L, Hart Y, Mayo A, Sontag ED, Alon U (2010) Fold change detection and scalar symmetry of sensory input fields. *Proc Natl Acad Sci USA* 107:15995–16000

Shoval O, Alon U, Sontag ED (2011) Symmetry invariance for adapting biological systems. *SIAM J Appl Dyn Syst* 10:857–886

Shoval ED (2017) A dynamical model of immune responses to antigen presentation predicts different regions of tumor or pathogen elimination. *Cell Syst* 4:231–241

Skataric M, Nikolaev EV, Sontag ED (2015) A fundamental limitation to fold-change detection by biological systems with multiple time scales. *IET Syst Biol* 9:1–15

Tu Y, Shimizu TS, Berg HC (2008) Modeling the chemotactic response of *Escherichia coli* to time-varying stimuli. *Proc Natl Acad Sci USA* 105(39):14855–14860

Weber EH (1905) *Tatsinn und Gemeingefuhl*. Verlag von Wilhelm Englemann, Leipzig

Scanning Probe Microscope Imaging Control

Juan Ren¹ and Qingze Zou²

¹Mechanical Engineering Department, Iowa State University, Ames, IA, USA

²Mechanical and Aerospace Engineering Department, Rutgers, The State University of New Jersey, New Brunswick, NJ, USA

Abstract

Scanning probe microscope (SPM) imaging control is to maintain a micromachined cantilever tip at the desired position with respect to the sample at nanoscale resolution while the probe is scanning the sample. The main objective is to drive the cantilever probe using piezoelectric actuators to follow the sample topography profile during the scanning process. Beyond conventional PI-feedback control, different approaches have been developed, ranging from robust control, adaptive control, to inversion-based feedforward-feedback control and iterative-based feedforward-feedback control. These

Supported by NSF CAREER Award 1751503 (Ren) and NSF Grants 1353890 and 1663055 (Zou)

developments have demonstrated various levels of improvements over PI-control.

Keywords

Scanning probe microscope · Tapping-mode imaging · Iterative learning control · Nanopositioning control · Adaptive multi-loop mode

Introduction

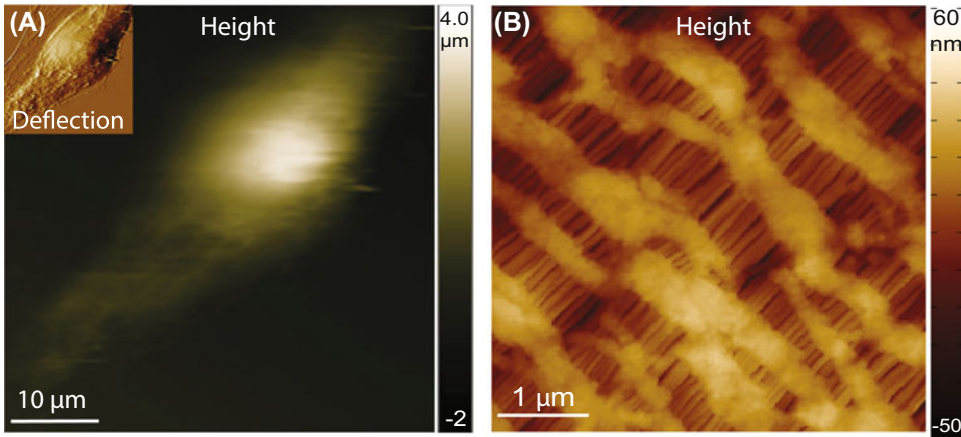
Scanning probe microscopy (SPM) has become a powerful tool for nanoscale imaging, property characterization, and patterning/fabrication. Particularly, based on the nature of the tip motion, SPM imaging can be categorized into static (i.e., contact) mode and dynamic (non-contact or tapping) mode. In contact mode, a micromachined cantilever probe with nanometer tip radius scans the sample line by line (usually in a raster pattern). During the scanning, the sample topography can be qualitatively captured using the deflection of the cantilever when the SPM z -axis scanner is kept at a constant height (i.e., the constant height contact mode). Or, more commonly, the topography can be quantified from the vertical z -axis displacement of the cantilever by maintaining the cantilever deflection closely around a pre-chosen set point through feedback control (i.e., the constant force contact mode). A major drawback of contact mode is that the sliding of the probe on the surface can result in sample deformation and damage. This issue can be mitigated through dynamic mode SPM imaging, by oscillating the cantilever around its resonance (with a piezoelectric actuator) during the scanning. The change of the oscillation amplitude or frequency, caused by the sample topography variation during the scanning, is then used as the feedback signal. The height of the cantilever is adjusted to maintain the oscillation amplitude or the frequency shift around the set point value, respectively, and the surface topography image is quantified from the z -axis displacement. Both types of imaging modes have

been broadly used in surface characterizations (see Fig. 1). Generally, contact mode is employed for samples with relatively high stiffness or those that are sensitive to impulse-like interaction (e.g., living cells), and dynamic mode such as tapping mode has been the de facto choice for relatively soft materials such as polymers.

As in other SPM operations, precision control of the probe position with respect to the sample is essential. Not only does the precision control of the probe-sample positioning ensure the imaging quality but also is required to avoid sample deformation and damage. Maintaining such a nanoscale positioning precision during high-speed imaging, however, becomes challenging due to the adverse effects of the vibrational dynamics and hysteresis of the piezoelectric actuators (Clayton et al. 2009), the complicated non-linear probe-sample interaction (in tapping-mode imaging), and the ultra-fragile surface of the sample (such as living cell) prone to deformation and damage. High-speed SPM imaging is needed to capture dynamic evolutions of the sample (such as the biological processes of living cells) – in addition to the improvement of imaging efficiency/throughput. These adverse effects in high-speed SPM imaging, although might be alleviated via hardware improvement (e.g., using piezoelectric actuators of higher bandwidth Shibata et al. 2010), need to be addressed via advanced control.

SPM Imaging Control

The control objective is to, during the imaging process, accurately track both the scanning trajectories in the lateral x - y - axes directions and the sample topography in the vertical z -axis direction. Although the desired lateral scanning trajectory (e.g., triangle trajectory – the default choice in almost all commercial SPM systems) is given a priori, the desired trajectory in z -axis – the sample topography profile on each scan line – is unknown a priori and tends to be largely irregular. This difference renders it more challenging to track the sample topography in the z -axis than the lateral scanning trajectory.



Scanning Probe Microscope Imaging Control, Fig. 1 SPM topography images: (a) contact mode images of a living NIH/3T3 cell in cell culture medium, reprinted from

Liu et al. (2019), copyright by Elsevier and (b) tapping mode topography image of a celgard sample, respectively

Lateral Scanning Control

In lateral scanning, the control issue mainly is to compensate for the vibrational dynamics and hysteresis effects of the piezoelectric actuators, coupled with environmental disturbances and system changes (e.g., the age of the piezoelectric actuators) (Clayton et al. 2009). Feedback control techniques such as PID control have been broadly used for their robustness against system variations, modeling error, and disturbances. PID control, currently widely employed in commercial SPM systems, is adequate for SPM imaging at relatively low speed and small range. However, the performance of PID deteriorates as the imaging speed increases, particularly at relatively large scan range. To enlarge the closed-loop bandwidth, robust control techniques have been developed (Sebastian and Salapaka 2005; Schitter et al. 2004). Alternatively, the adverse effects of creep, hysteresis, and vibrational dynamics in SPM piezo-positioning system can be quantitatively characterized and compensated for via model-based feedforward control, particularly the inversion-based feedforward control (Croft et al. 2001; Clayton et al. 2009). The robustness of the inversion-based feedforward approach can be enhanced by integrating the feedforward control to a feedback control loop (Schitter et al. 2004; Zou et al. 2004).

Further tracking improvement in lateral scanning has been witnessed through the development of iterative learning control (ILC) to SPM imaging applications (Tien et al. 2005). The repetitive nature of lateral scanning – with the entire desired trajectory known a priori – provides an ideal scenario for ILC application. For example, the following inversion-based iterative control (IIC) (Tien et al. 2005) has been shown to be effective to compensate for both the dynamics and hysteresis effects of piezoelectric actuators (Wu and Zou 2007).

$$u_{k+1}(j\omega) = u_k(j\omega) + \rho G^{-1}(j\omega) [y_d(j\omega) - y_k(j\omega)] \quad (1)$$

where “ $j\omega$ ” denotes the Fourier transform of the corresponding signal, $u_k(j\omega)$ and $y_k(j\omega)$ are the input and output signals in the k th iteration, $y_d(j\omega)$ is the desired trajectory, $G(j\omega)$ is the dynamics model of the piezoelectric actuator, and $\rho > 0$ is the iterative gain to be designed, respectively. As the system variation and the environment uncertainty in SPM operations are largely quasi-static (i.e., the input-output mapping of the system remains largely unchanged during each operation, even though the day-to-day variation can be significant), ILC avoids the performance-

robustness trade-off as that in usual feedback framework and attains both high performance and robustness (e.g., the quasi-static system changes can be accounted for via a couple of iterations right before the operation). The ILC approach has also been extended through a data-driven approach (Kim and Zou 2013; Wang and Zou 2015), by replacing and updating the a priori measured inverse model in Eq. (1) with the input-output data measured in the previous iteration, e.g., replace $G^{-1}(j\omega)$ with $u_k(j\omega)/y_k(j\omega)$ (Kim and Zou 2013), and, furthermore, with those measured data in past iterations (Wang and Zou 2015). Such an extension eliminates the modeling process and further enhances the tracking. Experimental results demonstrated superior tracking performance in large-range, high-speed tracking (Kim and Zou 2013; Wang and Zou 2015).

Efforts have also been made to increase the scanning speed through trajectory design, i.e., design non-raster, smooth scanning trajectory, to mitigate and remove the high-frequency components contained. Different scanning trajectories including cycloid, spiral, and lissajous patterns have been proposed (Mahmood and Moheimani 2009; Yong et al. 2010). Combining with advanced feedback control, lateral scanning speed can be increased at the cost of having a varying-speed scanning (e.g., nonuniform spatial sampling).

Vertical Sample Topography Tracking

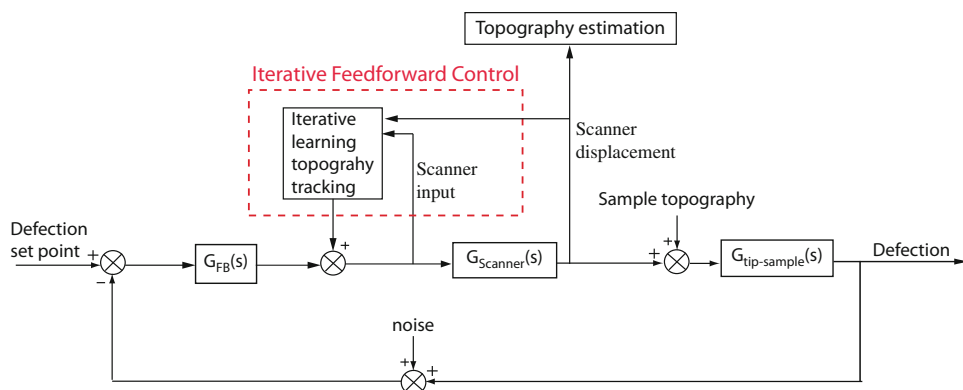
Compare to the control of lateral scanning, tracking the sample topography in the vertical z -axis direction is much more challenging. Unlike in the lateral scanning, the sample topography – the desired trajectory to track – is, in general, unknown a priori, and sample topography tracking can be sensitive and disturbed by the probe-sample interaction force, particularly in tapping-mode imaging. Moreover, accuracy in topography tracking is more critical to both the imaging quality and preserving the sample (soft samples).

Traditionally, PID control has been commonly used in commercial SPM systems for z -axis topography tracking. As the scanning speed increases, H_∞ -based robust control and model

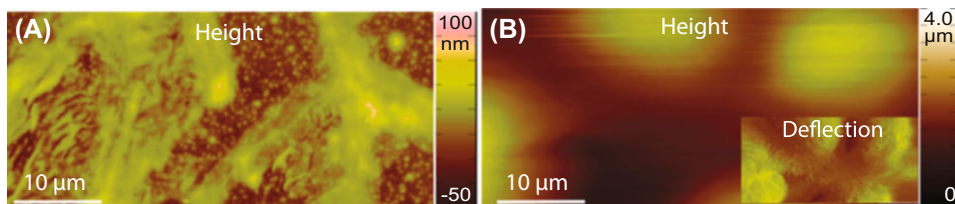
predictive control techniques have also been developed to improve the topography tracking (Quant et al. 2009; Rana et al. 2012). The effectiveness of these feedback control methods, however, is limited due to the adverse effects of the piezoelectric actuator becoming more severe in high-speed imaging, and the trade-off between the performance and the robustness requirement (resulting in a limited closed-loop bandwidth). Then, feedforward control has been augmented to the FB-loop in the feedforward-feedback (FF-FB) control configuration (see Fig. 2). The feedforward controller has been developed via approaches including the stable-inversion and the robust control, and the desired trajectory to be tracked has been approximated as the topography profile measured on the preceding scan line (Schitter et al. 2004). To further improve the topography tracking at higher scanning rate, ILC, particularly, the IIC technique has been extended and implemented online as the feedforward part of the FF-FB controller (Wu and Zou 2009). The basic idea is to repetitively image on the first line until the convergence is reached (usually within only 3 to 5 iterations), then continue on the rest of the scan lines without iteration. As having been analytically quantified and experimentally demonstrated, this online iteration scheme converges in practice as long as the convergence is attained faster than the line-to-line topography variations (Wu and Zou 2009). The ILC-based FF-FB control has also been further extended by improving the feedforward part with the modeling-free, data-driven ILC approach mentioned above (Ren and Zou 2014), or the feedback part with techniques such as model predictive control (Xie and Ren 2019).

Recent Trends

Recent efforts in control of SPM imaging have been focused on optimization of the imaging process, further increase of the imaging speed in large range scanning, high-speed dynamic mode imaging, non-conventional mode of imaging, and



Scanning Probe Microscope Imaging Control, Fig. 2 Schematic block diagram of iterative learning-based feedforward-feedback sample topography tracking control



Scanning Probe Microscope Imaging Control, Fig. 3 AFM image of (a) a poly(tert-butyl acrylate) sample with irregular patterns obtained by using the AMLM technique (Scan rate: 25 Hz). Reprinted from Ren et al. (2014).

Copyright (2014) by APS. And (b) prostate cancer cells (PC-3 cells) obtained by using the adaptive scanning mode technique at averaged scan rate of 0.86 Hz. Reprinted from Ren and Zou (2018). Copyright (2018) by Elsevier.

combination of imaging with other surface characterization operations.

Optimization of SPM Imaging Process

First, attention has been steered towards the optimization of the imaging process itself. For example, the topography tracking error can be accounted for in the sample image generation to improve the quality. This concept – account of the tracking error in the topography quantification – has also been extended to adaptively tune/adjust the set point of the cantilever deflection (i.e., the probe-sample force) – instead of using a pre-chosen *fixed* value as in conventional SPM imaging. With this online tuning of the deflection set point, the probe-sample interaction force can be minimized, i.e., around the minimal value needed for maintaining a stable probe-sample contact (Ren and Zou 2014). This notion of adaptive deflection (force) tuning has also been extended to tapping-mode imaging, by

regulating not only the tapping amplitude, but also the mean value of the probe tapping, leading to a multi-loop FF-FB control approach (Zou et al. 2017). The efficacy of such an adaptive SPM imaging approach has been experimentally demonstrated (see Fig. 3). Also, to reduce the probe-sample force – critical in imaging soft samples, particularly live biological species, it has been proposed to adaptively adjust the scanning speed to accommodate the topography variation (Zhang et al. 2011), such that the sample deformation can be minimized while the overall imaging throughput can be improved (Ren and Zou 2018). Alternatively, the scanning pattern can also be designed to explore the sparsity of the sample – rather than scanning the entire imaging area, the total scanning might be reduced by only focusing around the features of interests (e.g., the DNA strands) (Andersson 2007), or scanning randomly-selected short-length pieces based on the compressed sensing concept. This

local-scan approach, however, requires that some priori knowledge of the sample pattern has been acquired, or rapid probe engagement and withdrawn can be attained (Wang and Zou 2018).

Simultaneous Topography Imaging and Property Mapping

Secondly, efforts have also been explored to combine topography imaging with mapping of other sample properties such as the nanomechanical properties of the sample (Li and Zou 2017; Li et al. 2019). This simultaneous imaging and mapping of the sample allows real-time correlation between the topography and other sample properties, needed particularly when the sample is undergoing dynamic evolution. The basic idea is to inject an excitation stimulus input (for the nanomechanical mapping) into the imaging feedback control loop and utilize Kalman filtering technique (Ruppert et al. 2016) to decouple and split the responses (Li and Zou 2017; Li et al. 2019). Aiming to broaden and expand the applications of SPM imaging, these efforts bring more control challenges and opportunities in emerging implementations (e.g., dynamic biological process).

Cross-References

- [Control Systems for Nanopositioning](#)
- [Non-raster Methods in Scanning Probe Microscopy](#)
- [Sensor Drift Rejection in X-Ray Microscopy: A Robust Optimal Control Approach](#)

Bibliography

- Andersson SB (2007) Curve tracking for rapid imaging in AFM. *IEEE Trans Nanobiosci* 6(4):354–361
- Braker RA, Luo Y, Pao LY, Andersson SB (2018) Hardware demonstration of atomic force microscopy imaging via compressive sensing and μ -path scans. In: 2018 annual American control conference (ACC). IEEE, pp 6037–6042
- Clayton GM, Tien S, Leang KK, Zou Q, Devasia S (2009) A review of feedforward control approaches in nanopositioning for high-speed SPM. *ASME J Dyn Syst Meas Control* 131:061101–1 to 061101–19
- Croft D, Shedd G, Devasia S (2001) Creep, hysteresis, and vibration compensation for piezoactuators: atomic force microscopy application. *ASME J Dyn Syst Meas Control* 123(1):35–43
- Kim K, Zou Q (2013) A modeling-free inversion-based iterative feedforward control for precision output tracking of linear time-invariant systems. *IEEE/ASME Trans Mechatron* 18(6):1767–1777
- Li T, Zou Q (2017) Simultaneous topography imaging and broadband nanomechanical mapping on atomic force microscope. *Nanotechnology* 28(50):1–11
- Li T, Zou Q, Singer J, Su C (2019) Adaptive simultaneous topography and broadband nanomechanical mapping of heterogeneous materials on atomic force microscope. In: American control conference, ACC 2019. IEEE
- Liu Y, Mollaeian K, Ren J (2019) Finite element modeling of living cells for AFM indentation-based biomechanical characterization. *Micron, Elsevier*, 116:108–115
- Mahmood I, Moheimani SR (2009) Fast spiral-scan atomic force microscopy. *Nanotechnology* 20(36):365503
- Quant M, Elizalde H, Flores A, Ramírez R, Orta P, Song G (2009) A comprehensive model for piezoceramic actuators: modelling, validation and application. *Smart Mater Struct* 18(12):125011
- Rana M, Pota HR, Petersen IR (2012) Model predictive control of atomic force microscope for fast image scanning. In: 2012 IEEE 51st annual conference on decision and control (CDC). IEEE, pp 2477–2482
- Ren J, Zou Q (2014) High-speed adaptive contact-mode atomic force microscopy imaging with near-minimum-force. *Rev Sci Instrum* 85(7):073706
- Ren J, Zou Q, Li B, Lin Z (2014) High-speed atomic force microscope imaging: Adaptive multiloop mode. *Physical Review E* 90(1):012405
- Ren J, Zou Q (2018) Adaptive-scanning, near-minimum-deformation atomic force microscope imaging of soft sample in liquid: live mammalian cell example. *Ultramicroscopy* 186:150–157
- Ruppert MG, Karvinen KS, Wiggins SL, Moheimani SR (2016) A kalman filter for amplitude estimation in high-speed dynamic mode atomic force microscopy. *IEEE Trans Control Syst Technol* 24(1):276–284
- Schitter G, Stemmer A, Allgöwer F (2004) Robust two-degree-of-freedom control of an atomic force microscope. *Asian J Control* 6(2):156–163
- Sebastian A, Salapaka S (2005) Design methodologies for robust nano-positioning. *IEEE Trans Control Syst Technol* 13(6):868–876
- Shibata M, Yamashita H, Uchihashi T, Kandori H, Ando T (2010) High-speed atomic force microscopy shows dynamic molecular processes in photoactivated bacteriorhodopsin. *Nat Nanotechnol* 5(3):208–212
- Tien S, Zou Q, Devasia S (2005) Control of dynamics-coupling effects in piezo-actuator for high-speed AFM operation. *IEEE Trans Control Syst Technol* 13(6):921–931
- Wang Z, Zou Q (2015) A modeling-free differential-inversion-based iterative control approach to simulta-

neous hysteresis-dynamics compensation: high-speed large-range motion tracking example. In: American control conference (ACC). IEEE, pp 3558–3563

- Wang J, Zou Q (2018) Rapid probe engagement and withdrawal with online minimized probe-sample interaction force in atomic force microscopy. In: ASME 2018 dynamic systems and control conference, DSCC 2018. American Society of Mechanical Engineers (ASME)
- Wu Y, Zou Q (2007) Iterative control approach to compensate for both the hysteresis and the dynamics effects of piezo actuators. *IEEE Trans Control Syst Technol* 15:936–944
- Wu Y, Zou Q (2009) An iterative based feedforward-feedback control approach to high-speed atomic force microscope imaging. *ASME J Dyn Syst Meas Control* 131:061105–1 to 061105–9
- Xie S, Ren J (2019) High-speed AFM imaging via iterative learning-based model predictive control. *Mechatronics* 57:86–94
- Yong Y, Moheimani S, Petersen I (2010) High-speed cycloid-scan atomic force microscopy. *Nanotechnology* 21(36):365503
- Zhang Y, Fang Y, Yu J, Dong X (2011) Note: a novel atomic force microscope fast imaging approach: variable-speed scanning. *Rev Sci Instrum* 82(5):056103
- Zou Q, Leang K, Sadoun E, Reed M, Devasia S (2004) Control issues in high-speed AFM for biological applications: collagen imaging example. *Asian J Control* 6(2):164–178
- Zou Q, Ren J, Liu J (2017) High speed adaptive-multi-loop mode imaging atomic force microscopy (13 July 2017), US Patent App. 15/326,237

Scheduling of Batch Plants

John M. Wassick
The Dow Chemical Company, Midland,
MI, USA

Abstract

For manufacturers operating batch plants, production scheduling is a critical and challenging problem. A thorough understanding of the problem and the variety of solutions approaches is needed to achieve a successful application. This entry will present a brief overview of batch operations and the state of the art of batch plant scheduling for nonexperts in the field.

Keywords

Dispatching rules · Optimization · Process networks · Production sequencing · Product wheel

Introduction

Batch plants, manufacturing operations composed of unit operations that operate in batch mode, are the primary manufacturing operations for the production of high margin products such as pharmaceuticals, specialty chemicals, and advanced materials. The scheduling of the sequence of operations over time has a significant impact on the overall performance of a batch plant (White 1989). The economic importance of batch plants, and the importance of scheduling for batch plants, has spawned a large body of research on the topic and a variety of commercial offerings.

The Nature of Batch Plants

In batch operations, the material transformation takes place in stages and the operation of each stage occurs over a specified time while the material remains in a particular unit operation performing that stage of production. (A familiar batch operation is baking a cake. Ingredients and their amounts, specified by a recipe, are combined and then subjected to a constant temperature over specified period of time to produce a cake.) A batch plant may have parallel units for some stages. Other stages may be operated in a continuous flow mode with a storage unit feeding the stage and another storage unit receiving the stage output. The path through the unit operations may be product dependent. Batch plants have highly diverse operational characteristics.

There are two broad categories of batch processes: (1) sequential where a batch moves from one stage to another without losing its identity and (2) networked where batches can be combined or split to feed downstream units (Mendez et al. 2006). Sequential processes can be further

classified as single stage, multi-stage, or multi-purpose.

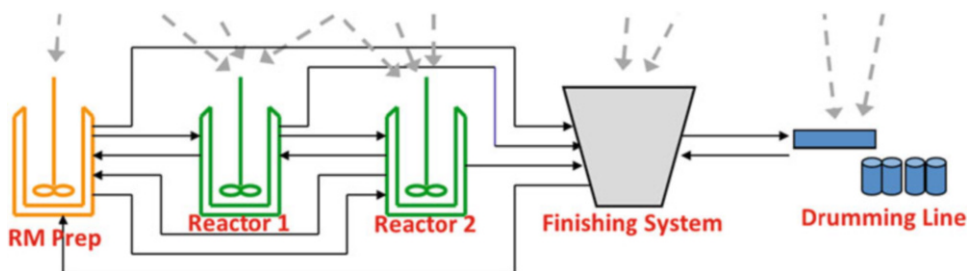
The nature of a batch process and the different process structures can be explored by referring to the process depicted in Fig. 1 (Chu et al. 2013). As drawn, this batch plant operates as a multi-stage sequential process where a batch starts in raw material preparation stage (selected raw materials are loaded and then blended for a specified time), moves to the reaction stage with two parallel units (prepared raw materials plus additives react at a constant temperature for a specified period of time), moves to the finishing stage (intermediate product is subjected to a vacuum for a specified period of time to remove volatile by-products), and finally is processed in the drumming stage (finished product is packaged in drums). If finished product storage tanks were placed between finishing and drumming to allow the drumming stage to be scheduled independently of the first three stages, then the drumming operation would represent a single stage sequential process. If we further assume that for some finished products Reactor 1 produces a batch of precursor for Reactor 2 and that some products produced in the reactors bypass the finishing stage and go directly to drumming, then the underlying plant would be a multi-purpose sequential process. Finally, if intermediate storage tanks exist for storing multiple batches of the precursors produced by Reactor 1 and the contents of the tanks are drawn off to produce multiple, subsequent batches in both reactors then the underlying plant is a networked process.

Besides the general structure of a batch plant, the specific processing requirements,

resources needs, and process constraints have significant impact on the complexity of the scheduling problem. One important aspect is limited resources that are shared between different operations. The availability and capacity of shared resources place a severe constraint on the timing of competing operations. Another significant factor is intermediate storage between stages and the inventory policies that are enforced. Like shared resources, intermediate storage places hard constraints on the timing of upstream and downstream stages, especially when no storage is available. A third important constraint on scheduling is product transition policies that dictate what operations need to be performed to move from one product to another in a given stage. Such operations, sometimes called setups, might involve cleaning, or producing buffer batches to isolate the chemistry of one product from another. These operations involve costs and subtract from the productive use of the equipment so they have significant impact on the sequencing of products through the plant.

Production Scheduling of Batch Plants

Production scheduling in a batch plant involves three fundamental decisions: (1) determining the size of each batch in each stage, (2) assigning a batch to a processing unit in each stage, and (3) determining the sequence and timing of processing on each unit. These decisions are well illustrated by a graphical planning board or Gantt



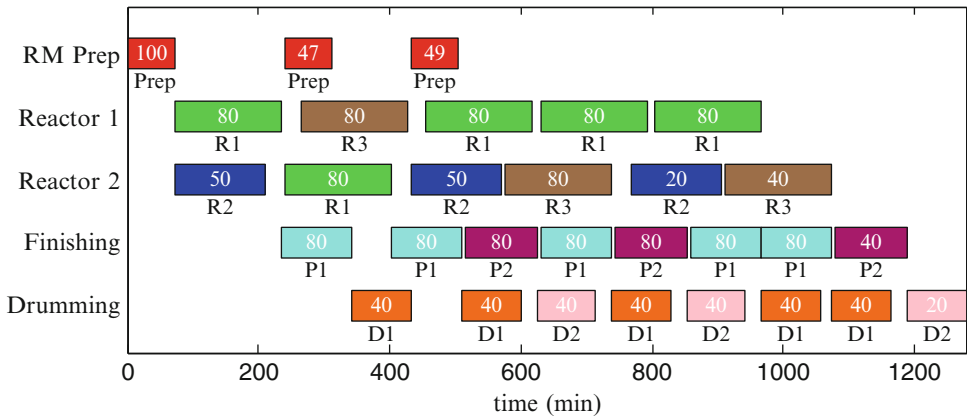
Scheduling of Batch Plants, Fig. 1 Example batch plant (Solid lines represent material flows from limited inventory. Dashed lines represent material flow from unlimited inventory)

chart as shown in Fig. 2 (Chu et al. 2013). Personnel charged with creating and managing production schedules often rely on such a graphical tool to construct, analyze and report the schedule. Generally production schedules are determined using the information listed in Table 1.

The scope of the scheduling decisions is defined by the level of process detail considered in the scheduling problem. This idea can be examined by referring to Figs. 1 and 2. Such a Gantt chart could apply to a batch plant with four stages of production: raw material preparation, reaction, finishing, and drumming, with two parallel reactors in the reaction stage. If dedicated finished product storage exists with large enough capacity to cover the process lead time then one schedule could be confined to the first three stages of production and a different schedule applied to the drumming stage. The scope of the scheduling problem could be further reduced if raw material preparation only takes place just in time to load a reactor rather than execute as soon as possible. In this situation, the time for raw

material preparation could be added to the reactor batch time and the schedule would involve only the reactors and the finishing system with the raw material unit or units schedule implied by the reactor schedule. At a higher level still, the first three stages of production could be considered a production train and scheduling could then be reduced to planning campaigns of batches for each product over time with the detailed synchronization of the individual stages left to operations personnel. Obviously with each level of abstraction some efficiency in the schedule is lost and subsequently the opportunity to increase throughput of the plant.

In most batch plants a person with a title such as “production scheduler” is charged with the scheduling decisions. In general, the production scheduler is responsible for delivering a production schedule that meets customer orders on time and maintains finished product inventory while dealing with rush orders, late deliveries, equipment breakdowns and other contingencies. Generally schedulers develop and publish a schedule



Scheduling of Batch Plants, Fig. 2 Gantt chart of a production schedule

Scheduling of Batch Plants, Table 1 Information generally used to construct a production schedule

Scheduling information	Examples
Detailed production recipes	Batch times, processing rates, unit ratios, sequence dependencies
Equipment data	Capacities, availabilities, product suitability
Facility information	Shared resource availability and capacities, storage capacities
Production costs	Raw materials, utilities, setups, cleanings, manpower
Production targets	Inventory replenishments, customer orders with due dates
Current process status	Current inventories, operations in progress, schedule items fixed in future time

to manufacturing on a regular basis (e.g., every 2 days, once a week, etc.) and then monitor ongoing circumstances (e.g., actual production vs. plan, new demand, etc.) to determine if minor adjustments to the schedule are needed or if a complete new schedule needs to be published. The construction of a schedule can be an iterative process involving negotiations with manufacturing, supply chain, sales, maintenance and logistics. The tools available to the production scheduler can have a significant impact on the quality of schedules they produce.

It is evident from the description above that production scheduling of batch plants is really carried out as an exercise in rescheduling in response to disturbances identified through feedback from the process and market. Under these circumstances production scheduling serves as a form of high level feedback control of the process. In this regard the manipulated variables are the production amounts for each product and the controlled variables are the inventory levels and customer service levels for each product. A scheduling problem can be converted to a state-space formulation and compared to model predictive control (Subramanian et al. 2012).

Solution Approaches

The solution approaches applied to scheduling batch plants cover a wide spectrum of sophistication. A very simple form is nothing more than a sequence of batches maintained on a white board in the plant control room. A level above this would be the use of custom spreadsheets for arranging batches chronologically and computing finished product inventory. Another step up is the use of a manually manipulated Gantt chart as illustrated in Fig. 2, possibly pre-populated by an automated planning application that determines the volume to be produced across the units of production while leaving the detailed sequencing and timing decisions to the production scheduler. The highest level of sophistication involves an automatically generated schedule with the application retrieving all the necessary data from the

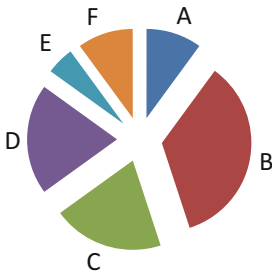
appropriate business databases and plant control system.

Regardless of the level of sophistication, all solution approaches rely on two fundamental components for developing a schedule. One is the modeling paradigm used to represent the physical system in a more abstract way. The primary components are: material balances in terms of batches or units of measure (e.g., pounds), and timing information as either precedence-based describing the order of operations or time grid-based describing the instant at which any operation takes place. Time can either be described by a continuous representation or divided into discrete increments. Within these two aspects of the modeling framework, significant freedom exists to describe the scheduling problem. The second fundamental component is the solution method used to generate the schedule. Each method has its strengths; therefore solutions combining methods are also used. The essential problem is to produce the information needed to draw the Gantt chart in Fig. 2 given the information in Table 1.

Product Wheel

While the primary objective of production scheduling is to meet customer orders while managing finished product inventory, other operational issues need to be managed, such as minimizing product transition costs, minimizing variability in manufacturing operations, keeping the scheduling process simple, and balancing the tradeoff between production lead times, inventory, and transition losses. The product wheel is a practical approach widely used in industry to address these competing issues. A product wheel is a regular repeating sequence of products made on a specific unit operation or an entire production process. A product wheel is typically depicted as a pie chart as shown in Fig. 3. Segments of the pie, called spokes of the wheel, represent a production campaign of a particular product. The size of the spoke represents the length of the campaign relative to the overall duration, or cycle time, of the wheel.

A product wheel has specific design parameters to address various operations objectives. The sequences is fixed and optimized for minimum



Scheduling of Batch Plants, Fig. 3 Product wheel

transition costs. The overall cycle time is fixed and optimized to balance lead time and inventory costs. The campaign size or spokes for each product are sized to match average demand for each product. The fixed pattern of the product wheel provides manufacturing with a predictable operational rhythm and the production scheduler with a very structured decision framework. Refer to King and King (2013) for a complete treatment of product wheels.

In practice, the duration of a campaign for a given product will vary from cycle to cycle as it will be sized to replenish any inventory consumed in the previous cycle. Low volume products may not be made on every cycle, although they will have a fixed location in the sequence. This same approach applies to make-to-order products that are not inventoried but produced to fill specific orders. Thus, in some cases a product wheel may be composed of several different but repeating cycles.

Dispatching Rules Used in Discrete Manufacturing

Batch processes are closely related to discrete manufacturing. Batches processed on a unit are analogous to jobs processed on a machine. Much of the literature on machine scheduling has focused on the analysis of the specifics encountered in general classes of problems such as single machines, parallel machines, flow shops and job shops, and developing constructive scheduling rules where a schedule is built up by adding one job at a time (Blackstone et al. 1982). Under certain circumstances these rules used for machine scheduling can be applied to

scheduling batch plants. This allows one to take advantage of a great body of literature, and at times, very simple scheduling rules that have proven optimality or worst case performance limits.

Consider again the batch process referred to in Fig. 1 which has two parallel reactors. The two reactors can be modeled as a single stage process and scheduled like parallel machines using the simple *shortest processing time first* (SPT) rule if the following circumstances hold: (1) raw material preparation can be included in the batch time of reactors, (2) significant storage exists between the reactors and finishing to essentially isolate the two stages, (3) product specific batch times are identical for both reactors, (4) the number of batches of each product is given (perhaps the result of an inventory policy for make to stock products), and (5) the objective is to minimize the total completion time for all batches. The SPT rule is simply to select, whenever a reactor is free, the batch with the shortest processing time from those yet to be processed. This can be proven to produce an optimal schedule for the given conditions.

Another simple dispatching rule to mention is the *earliest due date first* (EDD) rule. This rule is designed for single stage processes without parallel units where each batch has an associated due date. The rule simply orders the batches in increasing order of their due dates to minimize the maximum lateness of all orders.

The conditions needed for the SPT rule or the EDD rule to produce an optimal schedule can be quite restrictive when considering batch processes, however these rules and others found in the machine scheduling literature (Baker and Trietsch 2009) can still produce a good initial schedule even in cases where optimality conditions are not satisfied. Once generated, the schedule can be improved by manual manipulation of the Gantt chart or the application of improvement heuristics.

Improvement Heuristics

Improvement heuristics try to improve the current schedule by searching for alternative solutions either in the neighborhood of the current schedule

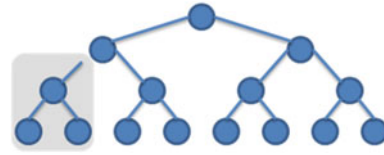
or by broadly exploring the solution space. The behavior of these algorithms is determined by tuning parameters that balance the use of the two search techniques and the underlying algorithm that performs the search. Improvement heuristics generally have the following basic procedure:

- Step 1: Initialize – determine a starting schedule
- Step 2: Generate alternatives – build modifications to the current schedule
- Step 3: Check for improvements in modified schedule – if no improvement is found return to Step 2 otherwise proceed to Step 4
- Step 4: Check for termination – terminate the algorithm if the number of iterations is exceeded or minimal improvement is obtained.

Many improvement heuristics are inspired by processes found in nature. Two of the more popular heuristics are simulated annealing which mimics the crystal formation during the cooling process of dense matter (Ryu et al. 2001) and genetic algorithms that mimic the evolution of a species over time (Löhl et al. 1988). A key aspect of improvement heuristics is the representation of the schedule in context of the algorithm used. For problems with complicated constraints this becomes a challenge. Nevertheless, when tuned properly and used where they fit the problem, improvement heuristics can produce very good schedules quickly.

Tree Search Methods

The scheduling solutions considered so far have taken a relatively simple view of a batch process as a single stage process or a flow shop. In situations where a batch plant involves shared resources, complicated transition rules or is a process network, tree search methods are better suited because they can deal with a large number of degrees of freedom and many types of constraints. Tree search methods rely on representing alternative schedules as the final nodes in a tree where intermediate nodes represent partial solutions of the schedule. To be practical, these methods must be able to effectively search through the tree while pruning non promising branches (see Fig. 4). Three of the most popular techniques are



Scheduling of Batch Plants, Fig. 4 Trimming the solution tree

mathematical programming, constraint programming, and beam search.

Mathematical programming solution techniques for scheduling generally convert the problem to a mixed integer linear programming (MILP) formulation where branching at nodes of the tree represent alternative values of the integer or binary variables. The tree is searched by a branch-and-bound algorithm which eliminates a node and the branch that emanates from it if the lower bound of the objective function represented by the terminal nodes of the branch is larger than the current best schedule. The MILP formulation can be stated generically as

$$\begin{aligned} \min \quad & z = cx + fy \\ \text{s.t.} \quad & Ax + By \geq b \\ & x \in \mathbb{R}_+^n, y \in \{0, 1\}^p \end{aligned}$$

where c , f , b are vector of constants, A and B are matrices of constants, and the solution is defined by the vector variables x and y . A key feature of using mathematical programming is to represent the relationships implied in Table 1 and Fig. 1 in terms of algebraic descriptions. The advantage of this approach is that a proven optimal solution exists for a problem stated this way. This provides the means to assess the quality of the solution and the impact of implementing the solution. The drawback of this approach is that since binary variables are used to represent the assignment of a batch to a processing unit, and the sequence and timing of processing on each unit, their number grows rapidly with the number of units and the length of the scheduling horizon. However, the performance of modern computing hardware and commercial solvers for MILP problems has allowed industrial size problems to be tackled.

A large variety of modeling paradigms have been developed to produce a MILP solution (Floudas and Lin 2004; Mendez et al. 2006). They address both sequential and networked processes using continuous time or discrete time representations. For sequential processes, time slot approaches have been developed. For networked processes, the resource task network and the state task network have been investigated by many researchers and have been used in industrial applications.

Constraint programming (CP) formulates a problem by writing constraints; but unlike the MILP method, the CP method stresses the feasibility of solutions rather than optimality. Another important difference is that constraints in the CP method do not have to be formulated as algebraic relationships but can be a more general form, thus making it easier in CP to represent complicated constraints. CP processes the constraints sequentially to reduce the space of possible solutions. At each node in the tree, CP processes one constraint after another, reducing the search space at each constraint. Being much newer than mathematical programming, constraint programming has a smaller body of literature to review but excellent performance has been reported in the literature (Baptiste et al. 2001).

In the beam search method, the branch-and-bound algorithm is modified to only evaluate the most promising nodes at any given level of the search tree (Ow and Morton 1988). The number of nodes evaluated is called the beam width and it is a key tuning parameter of the method. Another important element of the method is the technique used to retain nodes for complete evaluation. The technique must balance speed versus thorough evaluation to keep the method practical without discarding promising nodes. The beam search method applied to scheduling has been investigated by many authors (Sabuncuoglu and Bayiz 1999).

Simulation

The simulation approach to scheduling batch plants relies on representing the plant and the relationships inferred by Table 1 in a computer program whose algorithms recreate the behavior

of the plant when executed. Generally, the simulators used for batch operations apply discrete event simulation (DES) where entities that have attributes like size, due date, priority, etc. are operated on by activities for a specified duration. Fundamental to DES are the use of queues to hold entities until conditions in the simulation allow them to proceed to their next activity. Time in a DES does not proceed in a continuous manner but rather advances when activities occur. Simulation has the advantage of being able to describe processes and operating policies of arbitrary complexity and model variability in the process operation. Simulators can be used to evaluate manually created schedules or can be combined with optimization and heuristics to produce schedules by simulation-based optimization (Pegden 2011).

An alternative to DES for batch scheduling is the use of multi-agent simulators which are composed of semiautonomous agents assigned to represent the operation of the process and the associated decision making. Each agent has a local goal and communicates with other agents to accomplish it. Like DES, multi-agent simulators are capable of describing very complicated processes. A production schedule can be built through negotiations between agents (Chu et al. 2013).

Selecting a Solution Approach

The selection of the approach for a given batch plant should be value-based, balancing improved revenue with long term cost of ownership by considering such factors as the technical competency of the production scheduler, the expected capacity utilization of the plant, the operational complexity of the plant, and the cost to maintain the scheduling application. The key is to obtain the least complicated solution by reducing the scheduling problem to the highest level of abstraction and by using the simplest solution method that provides an effective schedule. See Harjunkski et al. (2013) and Pinedo (2008) for a survey of methods and recommendations for their practical application.

Summary and Future Directions

While there are a great variety of solution methods for scheduling, there are still promising research areas to be investigated. The recent introduction of sophisticated, object oriented process control systems with ties to enterprise management systems sets the stage for the development of automatic, real time scheduling. It is here that the principles of feedback control can be applied to batch plant scheduling. Pursuit of this goal will require continued development of fast, adaptive scheduling methods, real time assessment techniques of schedule performance, and tight integration of scheduling with the process control.

Cross-References

- [Control and Optimization of Batch Processes](#)
- [Models for Discrete Event Systems: An Overview](#)

Bibliography

- Baker KR, Trietsch D (2009) Principles of sequencing and scheduling. Wiley, Hoboken
- Baptiste P, Le Pape C, Nuijten W (2001) Constrained-based scheduling: applying constraint programming to scheduling problems. Kluwer Academic, Dordrecht
- Blackstone JH, Phillips DT, Hogg GL (1982) A state-of-the art survey of dispatching rules for manufacturing job shop operations. *Int J Prod Res* 20:27–45
- Chu Y, Wassick JM, You F (2013) Efficient scheduling method of complex batch processes with general network structure via agent-based modeling. *AIChE J*. doi:10.1002/aic.14101 (accepted)
- Floudas CA, Lin XX (2004) Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Comput Chem Eng* 28:2109–2129
- Harjunkski I, Maravelias C, Bongers P, Castro P, Engell S, Grossmann I, Hooker J, Méndez C, Sand G, Wassick J (2013, submitted) Scope for industrial applications of production scheduling models and solution methods. *Comput Chem Eng* 60:277–296
- King PL, King JS (2013) The product wheel handbook: creating balanced flow in high-mix process operations. Productivity Press, New York
- Löhl T, Schulz C, Engell S (1988) Sequencing of batch operations for a highly coupled production process: genetic algorithms versus mathematical programming. *Comput Chem Eng* 22:S579–S585
- Mendez CA, Cerda J, Grossmann IE, Harjunkski I, Fahl M (2006) State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput Chem Eng* 30:913–946
- Ow PS, Morton TE (1988) Filtered beam search in scheduling. *Int J Prod Res* 26:35–62
- Pegden DD (2011) Business benefits of Simio's risk-based planning and scheduling (RPS). In: Simio – resources – white papers. <http://www.simio.com/resources/white-papers/>. Accessed 1 June 2013
- Pinedo ML (2008) Scheduling: theory, algorithms, and practice, 3rd edn. Springer, New York
- Ryu JH, Lee HK, Lee IB (2001) Optimal scheduling for a multiproduct batch process with minimization of penalty on due date period. *Ind Eng Chem Res* 40:228–233
- Sabuncuoglu I, Bayiz M (1999) Job shop scheduling with beam search. *Eur J Oper Res* 118:390–412
- Subramanian K, Maravelias CT, Rawlings JB (2012) A state-space model for chemical production scheduling. *Comput Chem Eng* 47:97–110
- White CH (1989) Productivity analysis of a large multiproduct batch processing facility. *Comput Chem Eng* 13:239–245
- Wong TN, Leung CW, Mak KL, Fung RYK (2006) Integrated process planning and scheduling/rescheduling – an agent-based approach. *Int J Prod Res* 44:3627–3655

Sensor Drift Rejection in X-Ray Microscopy: A Robust Optimal Control Approach

Sheikh Mashrafi¹, Curt Preissner¹, and Srinivasa Salapaka²

¹Argonne National Laboratory, Lemont, IL, USA

²University of Illinois at Urbana-Champaign, Urbana, IL, USA

Abstract

High-resolution x-ray imaging requires the relative motion between the zone-plate optics and sample stages to accurately track a reference trajectory with large temporal bandwidth and high positioning resolution. Recent advances in feedback control designs have enabled large tracking bandwidth, high positioning resolution, disturbance

rejection, noise attenuation, and robustness to unmodeled plant dynamics. However, these efforts have been impeded by sensor drifts, which are not adequately addressed by existing designs. Especially since high-resolution x-ray imaging experiments are typically of long duration, the uncompensated effects of drifts result in imaging artifacts and substantial degradation in achievable spatial image resolution. In this entry, a method is presented for countering sensor drift in real-time through drift measurements and incorporating them in an optimal control architecture.

Keywords

Drift measurement · Real-time compensation

Introduction

X-ray microscopes provide critical advantages over optical counterparts since they achieve higher image resolutions on the order of nanometers due to their shorter wavelengths and enable imaging the internal structure of samples since they penetrate matter far easier than visible light. In scanning transmission x-ray microscopy (STXM), x-rays are produced by passing a high-energy electron beam through a periodic array of magnets, where the Lorentz force causes undulations, as shown in Fig. 1. These undulations elicit the production of synchrotron radiation, an intense beam of x-ray light. This beam, after being filtered through a monochromator, which selects a narrow bandwidth around a chosen frequency, is focused on the sample by appropriately positioning the zone-plate optics stage. This focused x-ray spot is then scanned along a predefined trajectory to cover a target region of the sample. The high-energy x-ray photons interact with the sample molecules and are diffracted in various directions as they pass through the sample, which are then detected downstream. A two-/three-dimensional image of the scanned sample is reconstructed from the recorded diffraction intensity data.

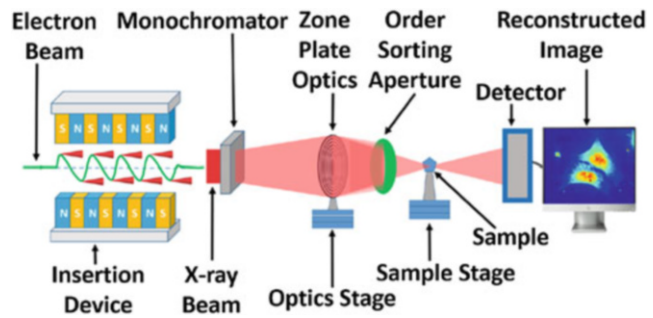
The image resolution is dependent not only on the quality of the x-ray beam but the precision of the relative motion between optics stage and sample stage. In Mashrafi et al. (2017), authors present a robust optimal control framework for the fine positioning of zone-plate optics stages, which was demonstrated on x-ray microscopes at Advanced Photon Source (APS) at Argonne National Laboratory (ANL). These experiments demonstrated reduction in scanning time by over four orders of magnitude, when compared to existing designs. This control-design framework for high-resolution, high-bandwidth, and robust positioning and tracking is well poised to remove positioning as a bottleneck for achieving high image resolution.

However, this framework relies heavily on how precisely the position of the moving optics stages are known with respect to the global reference frame. The high-resolution laser interferometric sensors are a great choice for implementing the feedback laws, except for one important limitation. The metal alloy fixtures that hold the sensor drift by hundreds of nanometers are due to the cyclic temperature changes, especially in long imaging experiments that can last many hours. For instance, temperature variations of about 0.6°C in a day can lead to a drift over 500 nm. These drifts are significant since the required positioning resolutions are only on the order of few nanometers. Since there is no way to distinguish between the actual motion of the scanning stage and the sensor drift, the control system does not even recognize the drift and fails to compensate for it. This entry presents a method to counteract the sensor drift.

Figure 2 illustrates the effect of sensor drift on position measurements of an optics stage. Such drift effects are present for the sample stage too; here we do not explicitly discuss its design since the framework presented for the optics stage directly extends to the sample stage. If there were no drift, and sensors were ideal, then the changes in the stage displacement y relative to the reference base are identical to the changes in the sensor measurement y_m ; however these are not the same when due to thermal

Sensor Drift Rejection in X-Ray Microscopy: A Robust Optimal Control

Approach, Fig. 1 A schematic diagram of the scanning transmission x-ray microscopy

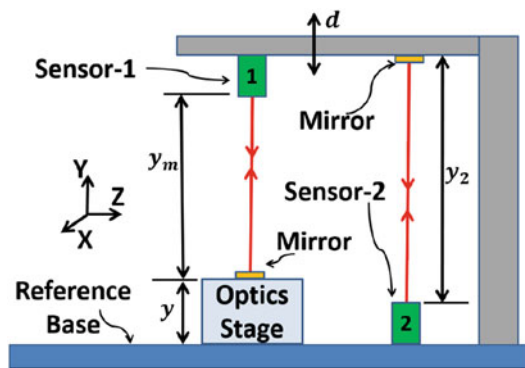


drift of the sensor fixtures, the sensor head itself drifts by a distance d . The existing literature covers primarily image post-processing methods such as using linear and nonlinear drift model to detrend drift effects or filtering and averaging over multiple exposures to increase signal to noise ratio (Beckers et al. 2013). These methods do not address the thermal drift of sensor in real time and result in images where it is not easy to distinguish artifacts from true features in the images.

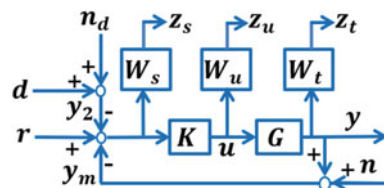
Drift Compensation Through Feedback Control

In the proposed concept, we add a sensor that measures the relative distance between a reference base and the sensor fixture, which allows for estimating drift of the primary sensor (sensor-1) and compensating for the drift effects. Figure 2 demonstrates the scheme where the optics-stage displacement y_m is measured with sensor-1 and its drift d is measured with sensor-2. Here the sensor-2 measures the motion of the sensor fixture to which sensor-1 is attached with respect to a fixed reference base and therefore gives a measure of the drift d . For nonideal sensors, the differential measurements about operating points are given by $y_m = y + n$ and $y_2 = d + n_d$, where n and n_d are the measurement noises in the two sensors. The measurement y_2 is incorporated in the optimal control design, whose objective is to counter the effects of the drift d .

Figure 3 represents the resulting closed-loop nanopositioning system. In this figure, G



Sensor Drift Rejection in X-Ray Microscopy: A Robust Optimal Control Approach, Fig. 2 Optics-stage measurement system. Primary sensor (sensor-1) drifts due to thermal drift of sensor fixture. Another sensor is added to detect this drift



Sensor Drift Rejection in X-Ray Microscopy: A Robust Optimal Control Approach, Fig. 3 Transfer function block diagram for the h-infinity minimization problem

represents the transfer function of the scanner which comprises the positioning flexure stage, the actuating, and the sensing components of the positioning system. Here y, u, r , and d represent the stage displacement, the input given to the actuator, the reference to be tracked, and the drift; n and n_d denote sensors' noise, and $y_m = y + n$ and $y_2 = d + n_d$ are the corresponding measurements. The main objective for the design

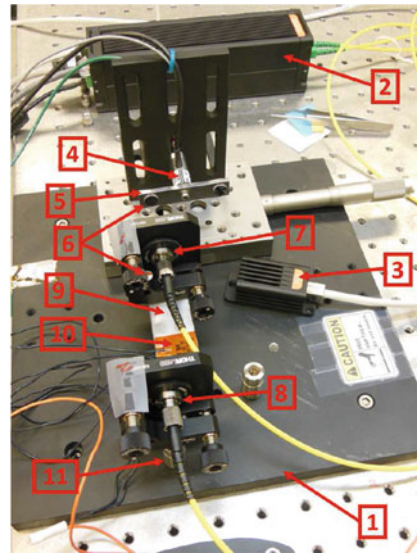
of the feedback control transfer function K is to make the tracking error small (how small is determined by the resolution requirement) over as large a bandwidth as possible while accounting for modeling uncertainties. The tracking error is given by $e = r - y - d = S(r - d) + T(n + n_d)$, where $S = 1 / (1 + G K)$ and $T = 1 - S$ are the sensitivity and the complementary sensitivity transfer functions. For good noise attenuation, the controller K needs to be such that $T \approx 1$ within the tracking bandwidth and small in high frequencies where measurement noise n and n_d are predominant. Similarly for drift compensation, S should be small over the desired tracking bandwidth and in low frequencies where drift d is predominant. Also, low values of the peak in the magnitude plot of sensitivity S ensures robustness to modeling uncertainties.

The robust optimal control theory provides an apt framework for incorporating these objectives. In this framework, it is possible to determine if a set of design specifications are feasible, and when feasible the control law K is obtained by posing and solving an optimization problem using easily accessible software routines (e.g., hinfsvy in Matlab). The main advantage of using this optimization framework is that it incorporates performance objectives directly into its cost function. This eliminates the tedious task of tuning gains (in trial-and-hit manner) as in the proportional-integral-derivative (PID) control designs, where even the exhaustively tuned gains may fail to yield acceptable performance. In our context, the optimal control algorithm K is obtained by solving the following minimization problem, $\min_{stab.K} \|T_{wz}\|_\infty$, where, T_{wz} is the closed-loop matrix transfer function from exogenous inputs $w = [r - d \ n \ n_d]$ to regulated outputs $z = [z_s \ z_u \ z_r]$. Here the weighted tracking error $z_s = W_s (r - y - d)$, weighted stage displacement $z_t = W_t y$, and weighted control effort $z_u = W_u u$ are the output signals that need to be made small; accordingly W_s is designed to be high over the desired tracking bandwidth, W_t is high over the frequency range where sensor noise is dominant, and W_u is made high in those frequency ranges where the control effort needs to be small.

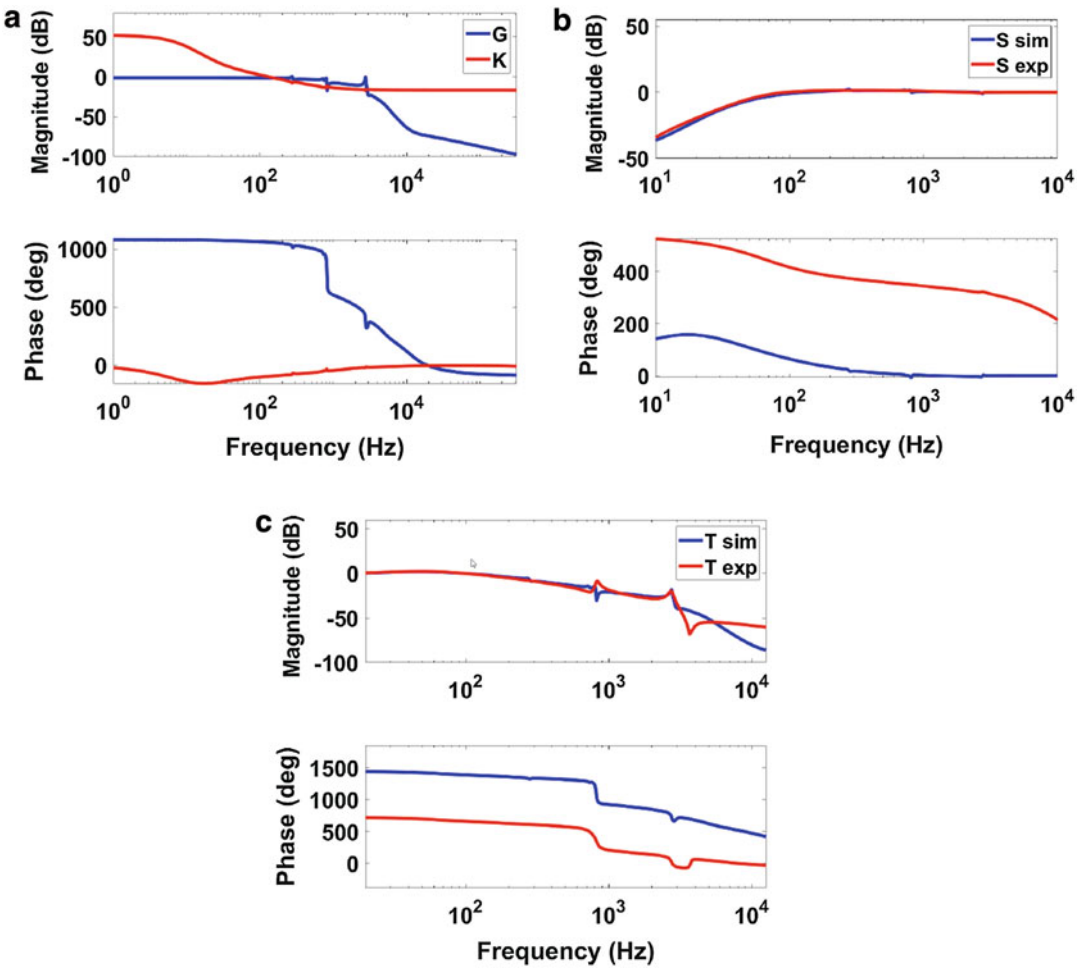
Experimental Setup and Results

Figure 4 illustrates the experimental setup. Here laser interferometric sensor is used to measure the displacements, where mirrors are fitted on the target surfaces and on the back of each sensor for reflecting the laser beam coming from the sensors. Such sensor shown as [7] with corresponding mirror [6] is used to measure the position of the piezo actuated [4] Al-alloy target [5], and sensor [8] measures the drift of the sensor [7]. Sensor [7] is positioned at one end of Al-alloy bar [9] (sensor fixture), which is attached on an Al-alloy post [11]. Sensor [8] is attached on another Al-alloy post. To replicate the sensor thermal drift, two heat sheets [10] were attached on both surfaces of the Al-alloy bar [9] and were turned on/off for certain time periods. The sensor measurements were used for the feedback which were implemented on a field-programmable gate array-based digital signal processor running at 40 MHz.

To determine the piezo actuator and stage dynamics, a band-limited uniform white noise with amplitude 1500 nm was given as input to the actuator, and the corresponding displacement was measured. A 18th order model G was fit to



Sensor Drift Rejection in X-Ray Microscopy: A Robust Optimal Control Approach, Fig. 4 Setup for sensor drift rejection



Sensor Drift Rejection in X-Ray Microscopy: A Robust Optimal Control Approach, Fig. 5 Bode plots of (a) the positioning system G and the designed controller K ;

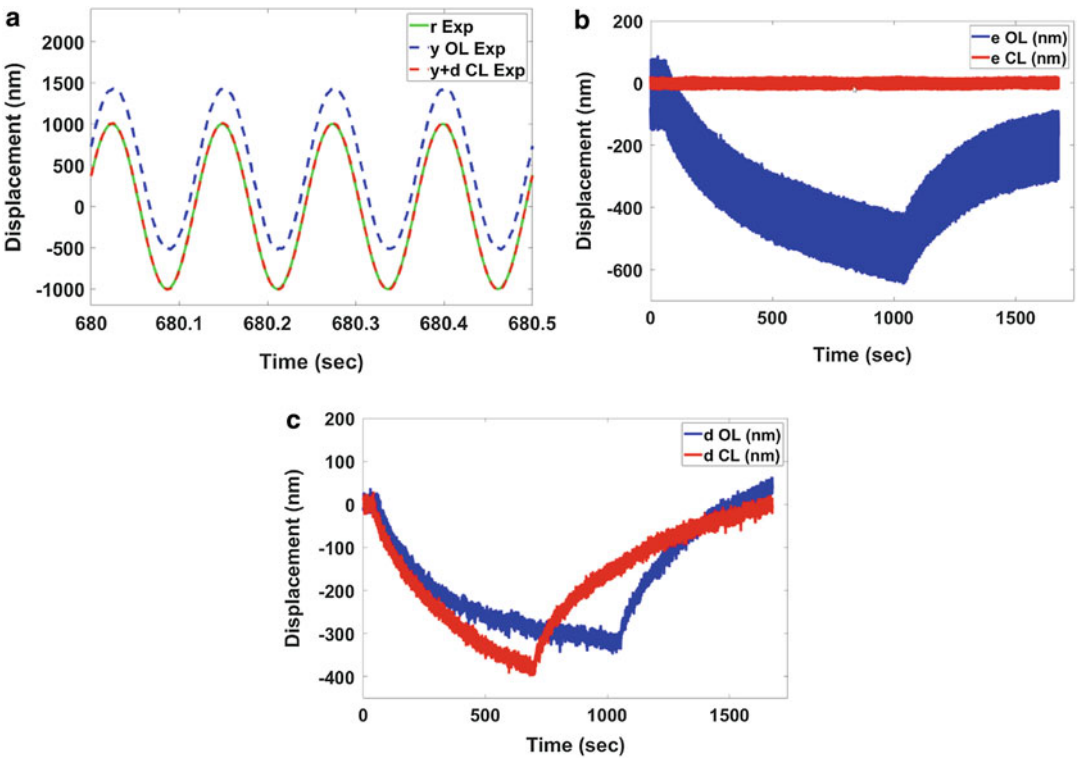
(b), (c) simulated and experimental sensitivity (S) and complementary sensitivity (T) maps

the nonparametric frequency response function, which was estimated from the time-domain input-output data (Fig. 5a). The first resonance peak of the actuator is around 800 Hz. Accordingly, the optimal control problem presented in the previous section was solved to obtain K shown in Fig. 5a; the resulting sensitivity S and complementary sensitivity T transfer functions are shown in Fig. 5b, c.

To replicate a 24-h thermal cycle of the APS beamline at ANL, the heat sheets were turned on for 10 min. Data was collected for 30 min to let the setup cool down to ambient temperature. The open-loop (OL) response of a sinusoidal signal with amplitude 1000 nm and frequency 8 Hz is shown in Fig. 6a. It does not distinguish sensor

drift (d_{OL} , (Fig. 6c)) from the actuator motion, and, consequently, the tracking error $e = r - y$ (Fig. 6b) is quite large and mimics the sensor drift. The closed-loop tracking error on the other hand is only about 2% of the input amplitude, of which 1% is due to the sensor noise.

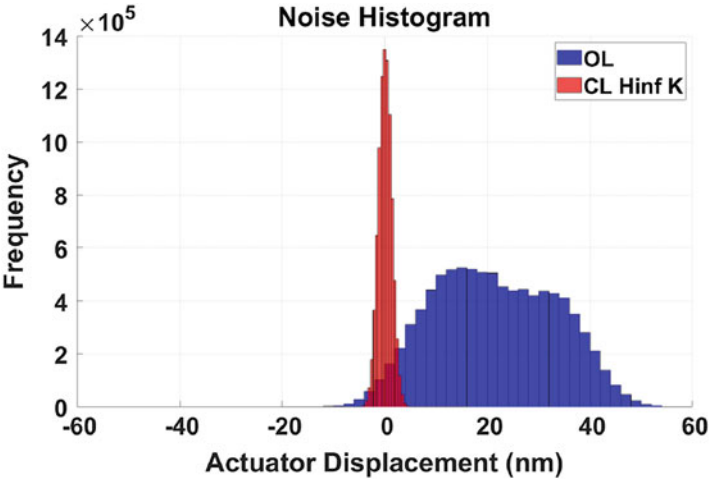
The positioning resolution of the actuator in open-loop (no knowledge of sensor drift) and in closed-loop (with knowledge of sensor drift) was calculated by giving a zero reference signal, where the actuator is solely driven by external disturbance and noise. Respective noise histograms of the measured displacements in Fig. 7 demonstrate an improvement of over 180% in 3σ -resolution – from 35.6 nm in open-loop to 3.8 nm in the closed-loop.



Sensor Drift Rejection in X-Ray Microscopy: A Robust Optimal Control Approach, Fig. 6 Open- (OL) and closed-loop (CL) tracking results. The control design achieves much better tracking as seen in (a), (b); the sensor drift is seen in (c)

Sensor Drift Rejection in X-Ray Microscopy: A Robust Optimal Control Approach, Fig. 7

Resolution experimental results. Histograms of open-loop and closed-loop responses to external disturbances and noise



Summary

To counter sensor drift, a key limitation of x-ray imaging resolution, an optimal control scheme is presented, which designs a feedback law that uses drift measurements to compensate for the drift

effects. The experimental tracking results demonstrate practical elimination of the drift effects with the design and substantial reduction of over 180% in positioning resolution. This design will significantly improve x-ray microscopy in terms of spatial resolution and quality of images.

Cross-References

- [Control Systems for Nanopositioning](#)
- [Control for Precision Mechatronics](#)
- [Mechanical Design and Control for Speed and Precision](#)
- [Scanning Probe Microscope Imaging Control](#)

Acknowledgments This research used resources of the Advanced Photon Source, a U.S. Department of Energy (DOE) Office of Science User Facility operated for the DOE Office of Science by Argonne National Laboratory under Contract No. DE-AC02-06CH11357.

Bibliography

- Beckers M, Senkbeil T, Gorniak T, Giewekemeyer K, Salditt T, Rosenhahn A (2013) Drift correction in ptychographic diffractive imaging. *Ultramicroscopy* 126:44–47
- Mashrafi S, Preissner C, Salapaka S (2017) The velociprobe: pushing the limits with fast and robust control. In: American society of precision engineering 32nd annual meeting, 2017, Vol. 67, pp 468–473
- Mashrafi S, Preissner C, Salapaka S (2018) Countering sensor drift in x-ray microscopy with fast and robust optimal control. In: Annual american control conference, 2018, pp 5119–5124
- Rodenberg J, Hurst A, Cullis A, Dobson B, Pfeiffer F, Bunk O, David C, Jefimovs K, Johnson I (2007) Hard-X-Ray lensless imaging of extended objects. *Phys Rev Lett* 98(1–4):034801

Simulation of Hybrid Dynamic Systems

Pieter J. Mosterman, Akshay Rajhans, Anastasia Mavrommati, and Roberto G. Valenti
Advanced Research and Technology Office, The MathWorks, Inc., Natick, MA, USA

Abstract

Hybrid dynamic systems combine continuous and discrete behavior. An overview of semantic domains of dynamic behavior is presented in terms of the evolution domain, the value domain, and the execution domain. Simulation

of piecewise continuous behavior interspersed with discontinuities is investigated in detail and illustrated by an example that motivates the use of simulation support for hyperdense time as an evolution domain. Engineered systems are characterized by combinations of compute elements, network connectivity, sensors and actuators, and a physical component. The various different elements benefit from modeling formalisms based on various semantic domains. Efficient execution is achieved by a combination of time-driven and event-driven execution engines.

Keywords

Cyber-physical systems · Dynamic systems · Hybrid systems · Modeling and simulation · Modeling physics · Multiparadigm modeling · Numerical simulation

Introduction

Hybrid dynamic systems combine continuous and discrete dynamics. In the literature, hybrid dynamic systems are often referred to simply as hybrid systems where system is to be taken as a mathematical system (as opposed to a physical system). Hybrid systems as a term, however, are also used to refer to systems such as a hybrid powertrain where a combustion engine and electric motor connect to the driveline. Therefore, to avoid confusion, the term hybrid dynamic systems is used.

Hybrid dynamic systems enable a selective choice in how to represent modeled phenomena. Modeling a system under study by hybrid dynamics allows improved efficiency in terms of simulation. Moreover, the ability to include behavior in a coarse sense prevents the need for detailed parameter knowledge. For example, when an electrical switch is opened, its current flow does not fall to 0 instantaneously, but the opening creates a quickly decreasing contact area that results in a quickly decreasing change in current. Moreover, a quickly increasing air gap may support a brief moment of current flow with a corresponding spark. The details of these effects

(the properties of the contact area and the air gap as they change during a brief period of time) may be unknown and difficult to measure while being irrelevant to the overall behavior of interest (the current changes to 0). Other examples of physical components that may be well modeled by discrete on/off type behavior include diodes, valves, latches, clutches, relays, and so forth. Likewise, components that effect quick changes compared to the overall behavior of interest such as an analog to digital converter may be modeled by discontinuous change in their behavior.

Even though systems that include components with on/off or more generally discontinuous behavior may be intuitively modeled as hybrid dynamic systems (e.g., an antilock braking system, a rectifier circuit, an overflowing tank), a system under study is not intrinsically a hybrid dynamic system. A model is a chosen representation of a system under study depending on a specific purpose. For example, to achieve real-time simulation, a model that abstracts fast temporal behavior into discontinuities oftentimes is preferred. However, to gain insight in the physics of a system, it is conceptually preferred to capture fast behavior in detail (Breedveld 1996).

A key aspect of being a representation is that a model must be simpler than the system under study. Thus, a perfect copy would not be considered a model. This notion may be confounded with the model being “wrong,” yet if the model serves its purpose, it helps solve a problem, it is in fact “right.” The quality of the model reflects in its efficiency to solve the problem. Combining continuous dynamics with discrete dynamics provides flexibility in modeling to include the detail necessary but no more (conform the principle of parsimony or Occam’s razor).

Models of hybrid dynamic systems can be studied and leveraged in multiple ways, for example, formal property proving to guarantee the absence of design errors or automatic generation of embedded code. The focus of this entry, however, is on numerical simulation of such models, that is, given a hybrid system model and the initial conditions, numerically simulating the execution of the dynamics on a computer. Such an execution of hybrid dynamic systems requires

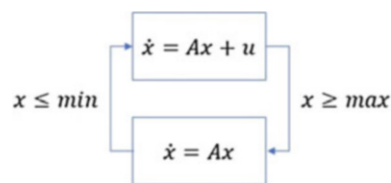
support for intricacies that bring to bear a unique set of challenges beyond the execution of continuous and discrete dynamics alone. In the proceeding, a number of motivating examples are presented to introduce various different forms of hybrid dynamic systems. Next, a range of semantic domains for dynamic systems and their execution semantics are introduced. The class of piecewise continuous systems is then discussed in detail as it embodies the broadest range of behaviors specific to hybrid dynamic systems. Next, the utility of the various different semantic domains in multiparadigm modeling is sketched followed by a discussion of execution engines to achieve efficient behavior generation. Finally, conclusions and open challenges are presented.

Motivating Examples

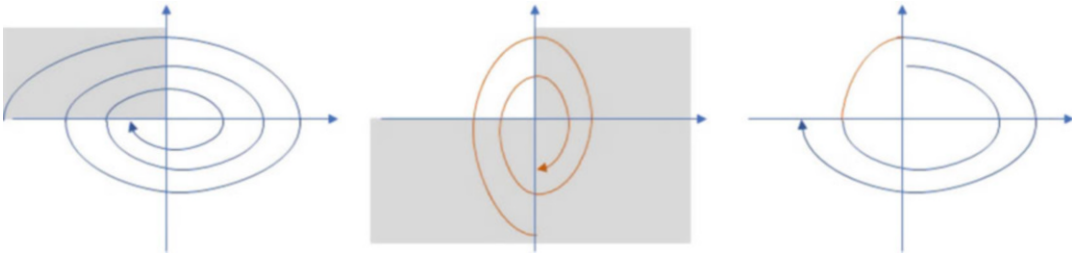
Examples of hybrid dynamic systems often used in the literature include a thermostat, a bouncing ball, and a freewheeling diode. Each of these allows concentrating on a particular aspect of hybrid behavior, namely, a discontinuous forcing function, state reinitialization, and event iteration, respectively.

Switched Control: A Thermostat

Switched control is a popular approach, for example, for tackling nonlinear system behavior. Piecewise linearization by approximating or abstracting the nonlinear behavior using a number of different modes of linear behavior allows synthesis methods to find control laws in each of the modes. Switching logic then selects



Simulation of Hybrid Dynamic Systems, Fig. 1
Thermostat model. © The MathWorks, Inc.



Simulation of Hybrid Dynamic Systems, Fig. 2 Combining asymptotically stable behavior leads to unstable behavior. © The MathWorks, Inc.

the corresponding mode as the system under control (the process or plant) evolves.

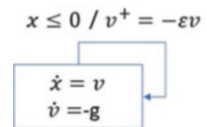
Another example is a thermostat that employs bang-bang control. Here, the actuation is at its maximum (maximum heating) or its minimum (no heating). Figure 1 shows a hybrid dynamic system model of a thermostat as a finite state machine with a differential equation that models the behavior in each discrete state. The thermal behavior of the room is represented by the differential equation $\dot{x} = Ax + u$ with x being the temperature and u the heating control input. The heating u is active or not, depending on the state. When the temperature falls below a minimum ($x \leq \min$), the heating turns on, and when the temperature exceeds a maximum ($x \geq \max$), the heating turns off.

From a simulation perspective, there are two important considerations: (i) does the model always remain in one of the states for a duration of time (i.e., is $\max > \min$) and (ii) is it necessary to reset the numerical solver that generates behavior for the differential equations? The second concern depends on the solver type, where a multistep solver exploits computed values of previous time steps and assumes a certain order of continuity between time steps (Cellier and Kofman 2006).

While the simulation considerations are relatively straightforward, from a control perspective, hybrid behavior may evidence problematic behavior. For example, Fig. 2 shows two asymptotically stable behaviors, one on the left-hand side and one in the center. When switched control selects the behavior in the top-left quadrant of the center behavior to produce the combined

Simulation of Hybrid Dynamic Systems,

Fig. 3 Bouncing ball model. © The MathWorks, Inc.



behavior on the right-hand side, the resultant behavior becomes unstable. Issues of stability, reachability, nondeterminism, and so forth are particularly difficult in case of hybrid dynamic systems but beyond the scope of the simulation focus.

State Reinitialization: A Bouncing Ball

A key behavior to support in simulation of hybrid dynamic systems is the reinitialization of states in differential equations (the continuous state). Figure 3 shows a model of a bouncing ball, a much studied example of this behavior. The differential equations that are active in the discrete state capture the velocity of the ball, v , as subject to the gravitational acceleration, g . When the position of the ball, x , falls below 0 (to represent the floor level), an elastic collision sets the velocity of the ball to a new value v^+ based on the velocity upon collision and a restitution coefficient ε .

The discontinuity as introduced by reinitializing the continuous state requires the same numerical solver considerations as the discontinuity introduced by switching input of the thermostat problem. The bouncing ball, however, highlights detecting the collision event as a challenge. Consider ε to be positive but less than 1. As the ball bounces repeatedly, the maximum height of each bounce progressively decreases. When this

maximum height becomes less than the numerical tolerance set to detect whether $x \leq 0$, the transition is always enabled. At this point, further simulation becomes ill defined.

Event Iteration: A Freewheeling Diode

A key feature of hybrid dynamic systems is the occurrence of a sequence of discrete state changes with no duration of continuous-time behavior in between. Figure 4a shows an electrical circuit with two elements modeled with discontinuous behavior: (i) the switch, S , enforces a 0 voltage drop when closed and a 0 current flow when open, and (ii) the diode, D , enforces a 0 voltage drop when on and a 0 current flow when off. When the switch is closed, the battery, B , supplies a voltage that builds up a flux in the inductor, L . The corresponding current flow creates a voltage drop across the resistor, R , and when less than the battery voltage, this results in a positive voltage across the diode in its off state.

Figure 4b shows a model of the electrical circuit as a state machine with three parts. At the top, the equations for the inductor are shown where the inductor flux, p , builds up based on the voltage drop, V_D , across the parallel branch. The current through the switch, I_S , minus the current through the diode, I_D , is the inductor current and directly represented as the flux per inductance, $\frac{p}{L}$.

Below the equations are two state machines, one to model the behavior of the switch and one for the diode. The switch is opened when time, t , reaches 1. At the point in time when the condition $t \geq 1$ becomes true, the switch changes its behavior from allowing arbitrary current with 0 voltage drop to allowing an arbitrary voltage with 0 current flow. Given that the diode also enforces 0 current flow, the inductor current is forced to 0, which requires an instantaneous discharge of its flux, p .

Because of the discontinuous change in flux, according to the differential equation $V_D = \dot{p}$, a negative voltage spike across the parallel branch would occur. This voltage, however, enables the $V_D \geq 0$ condition (notice the orientation of the diode with its positive port connected to ground), and the diode comes on such that the diode

provides the inductor current instead of forcing it to 0.

In addition to the complexity of handling such event sequences, hybrid dynamic system simulation must handle impulsive behavior such as the voltage spike. Moreover, the simulation must be able to compute discontinuous changes in continuous state from a system of equations (i.e., the reinitialization value is not explicitly provided as an assignment).

Semantic Domains

Given the notion that hybrid dynamic systems combine continuous and discrete behavior, it is instructive to investigate in more detail the semantic domains used in modeling and simulation. To establish a fundamental structure of these semantic domains, note that the dynamic nature of the systems implies there is a domain on which behavior evolves. A behavior, β , is then a mapping of an evolution to values where a hybrid dynamic system can be thought of as a partial function with the evolution domain, $\mathcal{E}$, as its domain and the value domain, $\mathcal{V}$, as its codomain

$$\beta : \mathcal{E} \mapsto \mathcal{V} \quad (1)$$

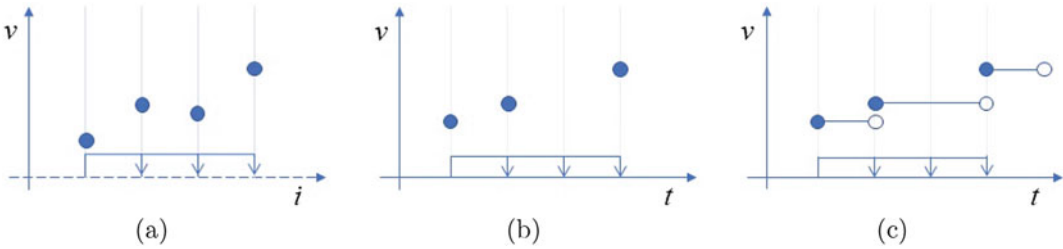
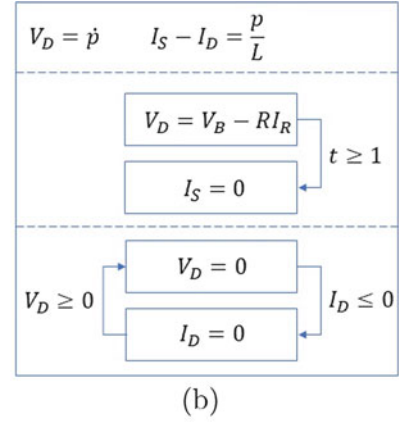
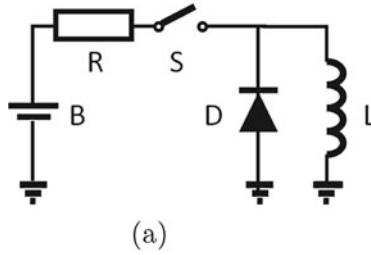
In addition to the evolution and value domains, dynamic systems often allow value changes at specific instances only. These instances form the execution domain, $\mathcal{X}$.

Integer Evolution Behavior

In its basic form, an evolution domain comprises a set of linearly ordered instances. This can be represented by the natural numbers, $\mathbb{N}$, even if the metric notion may not be relevant. For example, program code such as C, Java, and MATLAB® (MathWorks® 2019), when single threaded, follows a linear sequence of operations that can be indexed by integers. Execution of each operation follows completion of the previous operation, and so a program counter progresses along the integers without the need for independent dynamic scheduling. The discrete evolution domain is depicted in Fig. 5a by a

Simulation of Hybrid Dynamic Systems,

Fig. 4 Freewheeling diode. (a) Circuit diagram. (b) Dynamics model. © The MathWorks, Inc.



Simulation of Hybrid Dynamic Systems, Fig. 5 Integer evolution semantic domains. (a) Program code. (b) Sampled time. (c) Discrete time. © The MathWorks, Inc.

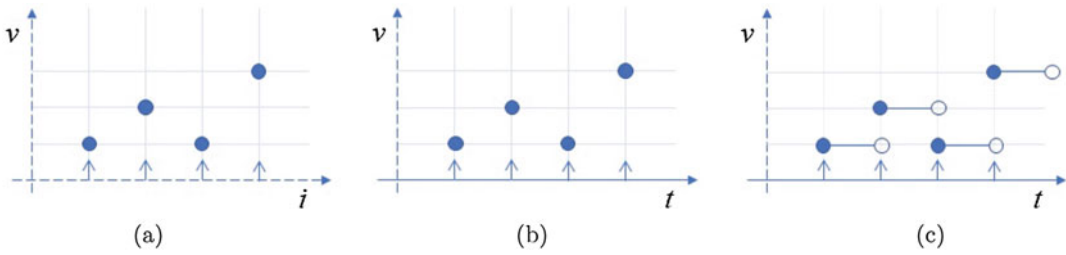
dashed horizontal axis, while the execution ordering is represented by squared arrows between consecutive execution instances. The solid vertical arrow representing the value domain captures the continuous values of computational operations. The solid circles depict values computed at discrete execution instances. It should be acknowledged that computational operations approximate continuous values by floating point representations, and so the domain is not truly continuous. In the value domain, this distinction is not important to the discussion.

Time is an evolution domain that is integral to hybrid dynamic systems. In computational implementations, it is often preferred to discretize time in periodic steps of equal size, the sample time. Figure 5b depicts time as a discrete evolution domain such that it can be specified by difference equations or synchronous data flow (Benveniste et al. 2003). Note that the base rate of the scheduler has a sample time that is represented by the vertical lines, but not every base rate sample hit

corresponds to the execution of an operation. For example, in case of a system with operations that have a sample time of 2 s and operations that have a sample time of 3 s, the base rate has a sample time of 1 s (the greatest common divisor). At 5 s, the base rate has a sample hit but neither of the other sample times does. As in programming code, the value domain is continuous.

Because of the integer nature of the execution, in case there is no real-time execution requirement, the scheduler may operate similar to the execution of program code and progress along an integer schedule. The sample times of operations are harmonics of the base rate and, therefore, execute at integer multiples of the execution index. In case real-time behavior is required, each integer advance is invoked based on a periodic interrupt from the compute platform.

Similar in execution but different in behavior is the discrete-time evolution domain shown in Fig. 5c. Here, the value of a behavior changes at a periodic rate, but the value is held and



Simulation of Hybrid Dynamic Systems, Fig. 6 Discrete value semantic domains. (a) Transition system. (b) Sampled transition system. (c) Discrete-time transition system. © The MathWorks, Inc.

accessible in between sample times. This is indicated as a continuous evolution domain by the solid horizontal arrow. For digital control systems, the zero-order hold (ZOH) approach is often preferred, whereas for digital image processing systems, a sampled time approach is preferred because it does not affect the signal frequencies.

Finite State Value Behavior

Finite state transition systems behave similar to the integer execution of general computations in Fig. 5. The corresponding semantic domains are shown in Fig. 6 where the finite state notion is depicted as a discrete value domain by the dashed vertical coordinate axis. Note that even though in a computational implementation the states of finite state transition systems are typically identified by integers, semantically the discrete values that represent states are not ordered and are without a metric.

Figure 6a illustrates the semantic domain of transition systems (e.g., automata Alur and Dill (1994) and Petri nets Murata (1989)) based on a discrete progression of discrete state changes. The states as well as the events that effect state changes can be indexed by integer values. In contrast to program code, the events are dynamically generated and not preassigned as the completion of a previous operation. For example, in a Petri net, the tokens in input places of a transition determine whether the transition is enabled. Thus, a scheduler dynamically schedules the transition to be active as the token distribution evolves. This dynamic execution behavior

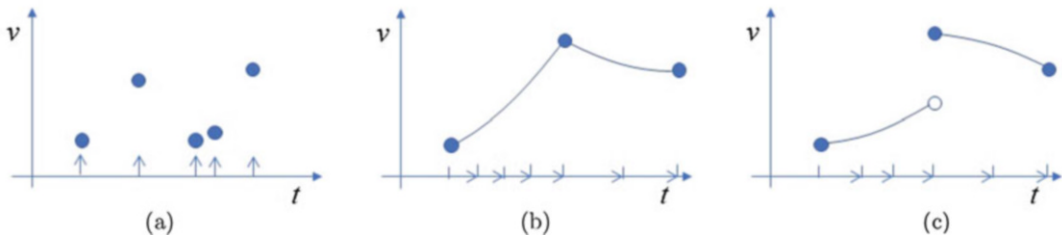
is depicted in Fig. 6 by the upward pointing arrows.

The semantic domains of the sampled and discrete-time versions of transition systems are illustrated in Figs. 6b, c, respectively. Associating time with the evolution domain enables sample times on a continuous domain, while the execution behavior is discrete and periodic. Given the dynamic scheduling nature, real-time behavior is not a determining factor whether a scheduler is necessary. In its execution, the sampled transition system corresponds to Mealy-type behavior where actions are associated with transitions. The discrete-time transition system corresponds to Moore-type behavior where actions are associated with states.

Continuous Execution Behavior

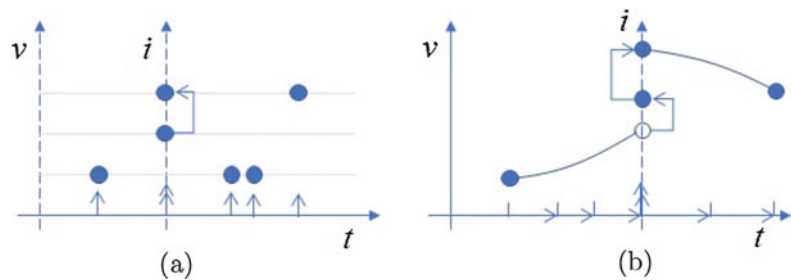
Executing events on a continuous schedule that is not periodic is the hallmark of discrete-event systems such as the discrete-event system specification (DEVS) (Zeigler et al. 2000). Figure 7a illustrates the semantic domain as a continuous evolution domain and a continuous value domain. Moreover, the discrete events at which value changes occur can be scheduled on a continuous execution domain. Note that transition systems with events occurring on a continuous domain would lead to a discrete value domain and bridge the semantic domains in Figs. 6b and 7a. However, in such transition systems, time itself is part of the state as clocks that can be reset, which creates a continuous value domain.

Behavior that is continuous in all its facets is illustrated in Fig. 7b. This behavior can be specified by switched differential equations, and



Simulation of Hybrid Dynamic Systems, Fig. 7 Continuous evolution and value semantic domains. (a) Discrete-event system. (b) Continuous behavior. (c) Discontinuities. © The MathWorks, Inc.

Simulation of Hybrid Dynamic Systems, Fig. 8 Multidimensional evolution domain. (a) Multiple events. (b) Event sequence. © The MathWorks, Inc.



the evolution domain in hybrid dynamic systems typically is time. Generating the behavior from the differential equations is based on gradients with respect to time. From an existing point in time, the gradients are used to extrapolate the behavior trace to a new point in time that approximates the underlying solution of the differential equation within a certain tolerance. If the behavior changes quickly against time, the behavior gradient is steep, and a small time step may be necessary to meet the tolerance constraint. For slow behavior, a larger time step may be possible. The varying step size along the (horizontal) time axis is depicted in Fig. 7b by horizontal arrows.

Piecewise continuous behavior interspersed with discontinuities is shown in Fig. 7c. The corresponding semantic domain is the same as the continuous behavior in Fig. 7b with the same execution domain and scheduler.

The piecewise continuous behavior with or without discontinuities may be considered to be of a hybrid dynamics nature. Indeed, the change from one piecewise segment of continuous-time behavior to the next may be invoked by the occurrence of a discrete event at the point in time of the transition.

Multidimensional Evolution Domains

In discrete-event systems such as the one illustrated in Fig. 7a, it is often desirable for two events to occur at the same point in time, either because they were scheduled as such independently or because one causes the other with no time delay.

Figure 8a sketches how multiple events at a given point in time would lead to a two-dimensional evolution domain. The double-headed arrow pointing up indicates two such events. Because both of those events have the same time associated with them, in order to distinguish their moment of occurrence (e.g., to order them), an additional index is necessary. Hence, the moment of each event occurrence is represented by a couple, $\langle t, i \rangle$, that comprises the time, t , and an index, i . The index is an element from an ordered set for which the integers may be chosen (even if the metrics are irrelevant). Figure 8a indicates the additional integer dimension by a dashed axis at the time when multiple events occur. Note that many events may be dynamically scheduled at the same point in time before the next time advance.

Similar to discrete-event systems, in hybrid dynamic systems such as illustrated in Fig. 7c,

multiple events may occur at the same point in time. Likewise, an additional dimension would be required in order to separate the moment of occurrence for the multiple events. Figure 8b illustrates the semantic domain with an additional axis to represent the integer part of the evolution domain at the point in time of the discontinuity where the sequence of events takes place.

Piecewise Continuous Systems

Simulation behavior that is piecewise continuous interspersed with discontinuities (e.g., the semantic domains in Figs. 7c and 8b) introduces complications that require special attention from a hybrid dynamic system perspective. Referring to Fig. 8b, four stages of execution can be recognized:

- Continuous behavior evolution that may require a solver reset in case continuity assumptions about magnitude and higher-order derivatives are violated when a discontinuity occurs.
- Detecting when events are generated, which introduces numerical challenges around threshold crossings.
- Support for multiple mode transitions at the same point in time and handling of initial states that are inadmissible or inconsistent.
- Reinitializing state based on discontinuities that are implicit and require handling instantaneous changes as having either 0 or infinitesimal duration.

Of these, first, the numerical challenges of event detection are considered in more detail. Next, challenges in the interaction between mode transition behavior and state reinitialization are studied after which a hyperdense evolution domain is introduced. Finally, caution is drawn to a number of pathological behaviors.

Event Detection

As a behavior evolves continuously, discontinuities are specified to occur when either time or

behavior values exceed certain thresholds. The former are referred to as *time events*, and their time of occurrence is known when they are scheduled. The latter are referred to as *state events* because the behavior values derive from the continuous state and the time of occurrence must be dynamically determined (i.e., as the behavior evolves) (Cellier 1979).

The threshold behavior of state events is specified by inequalities. For example, the thermostat model in Fig. 1 generates events when the temperature falls below *min* as specified by $x \leq \text{min}$ or when the temperature exceeds *max* as specified by $x \geq \text{max}$. Note that the inequalities include the threshold value (they include the equal sign) so as to have clearly defined left limit values to follow the progression of time. These causal (in a temporal sense) limits imply that the intervals of continuous evolution will be left closed and right open (Mosterman et al. 1998a).

Detecting whether the value of a variable crosses a threshold can be reformulated to detect whether a function of that variable crosses 0. This function is called an indicator function, g_α , that takes as input the continuous state of the system, x , the forcing function, u , and time, t . The indicator function may differ per discrete mode, α . The indicator function is used to determine whether a zero crossing, zc , occurred based on its sign:

$$zc_\alpha(x, u, t) = \begin{cases} 1 & \text{if } g_\alpha(x, u, t) > 0 \\ 0 & \text{if } g_\alpha(x, u, t) = 0 \\ -1 & \text{if } g_\alpha(x, u, t) < 0 \end{cases} \quad (2)$$

In case zc changes its sign, an event is generated. For example, for the freewheeling diode model in Fig. 4b when the diode is in its *off* state ($I_D = 0$), an indicator function is

$$g_{off}(V_B, p) = V_B - R \frac{p}{L} \quad (3)$$

where p is the continuous state and V_B is the forcing function, while R and L are parameter values for the resistor and inductor, respectively. When this function changes its sign, a zero-crossing event is generated.

Detecting whether a change of sign occurs typically requires the indicator function to be evaluated positively and negatively. Thus, the indicator function fails to guard against invalid operations (e.g., taking the square root of a negative variable). Rather than returning an error, the invalid operation may return a substitute value that enables successful zero-crossing detection (e.g., the square root may return a negative value based on the absolute value), while the substitute value is never considered an actual (accepted) simulation value. Note that there are techniques to prevent invalid operations but at the cost of efficiency (Esposito et al. 2001).

Even though including the equal sign in threshold comparison results in sound mathematical limit behavior, numerically it is less meaningful. It is well-known in computer programming not to check for equality involving a floating point value, for example, as a loop termination condition. Likewise, detecting whether an indicator function equals 0 is challenging. In a common approach, the 0 level is defined as a “numerical 0” by a tolerance, tol .

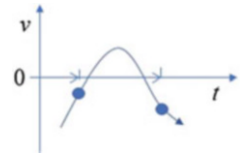
$$zc_{\alpha}(x, u, t) = \begin{cases} 1 & \text{if } g_{\alpha}(x, u, t) > tol \\ 0 & \text{if } -tol \leq g_{\alpha}(x, u, t) \leq tol \\ -1 & \text{if } g_{\alpha}(x, u, t) < -tol \end{cases} \quad (4)$$

Though a numerical zero enables the determination that the value of a behavior is (numerically) 0, it complicates detecting an actual crossing of 0 (i.e., a change in sign of the indicator function). The tolerance around 0 allows a behavior to be numerically 0 for a duration of time even if the gradient is nonzero. And so the indicator function may show a sequence of signs at simulation steps as $- \rightarrow 0 \rightarrow 0 \rightarrow +$. It becomes difficult to determine whether this constitutes an actual 0 crossing or whether the behavior first settled at 0 and then started to deviate.

Another key challenge in zero-crossing detection is when an even number of crossings occur within one simulation step. Figure 9 illustrates the situation where the solid circles indicate values of the indicator function computed at two consecutive simulation steps. The dynamics of the indicator function allow its value to change

Simulation of Hybrid Dynamic Systems,

Fig. 9 Zero crossing with even zeros. © The MathWorks, Inc.



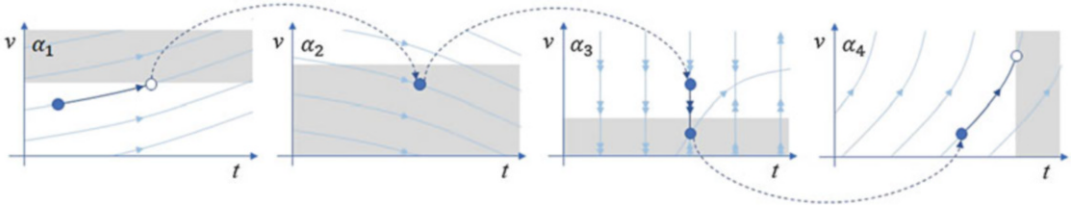
from negative to positive and back without a change in sign being detected at the simulation steps. One approach to mitigating this problem is to include the indicator functions as part of the step-size control that the numerical solver uses for generating continuous-time behavior, which introduces additional computational cost.

After detecting a zero crossing, the point in time at which the indicator function changes sign is located within a given tolerance (different from the tolerance on the value domain). Well-understood root-finding methods such as Newton's method or a bisectional search can be used for this purpose (Park and Barton 1996; Zhang et al. 2008).

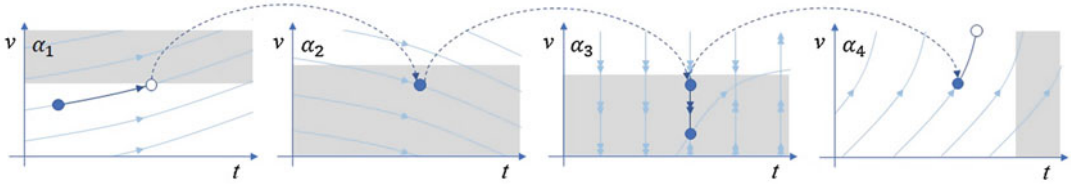
Mode Transition and State Reinitialization

The inequalities that give rise to mode transition events define where on the semantic domain behavior evolution is admitted and where not. Figure 10 depicts a sequence of mode transitions from left to right. In the initial mode, α_1 , behavior evolves in continuous time from an initial state (the solid circle) according to the vector field lines.

The gray area indicates an area of the semantic domain that is inadmissible. Once the behavior value exceeds the lower bound of the area, a zero-crossing event causes a transition to mode α_2 . The state of the behavior immediately prior to the mode transition taking place is used to initialize behavior in mode α_2 . As shown in Fig. 10, this state is in the inadmissible area of mode α_2 causing an immediate further transition to mode α_3 . Mode α_3 consists of a *jump space* as indicated by the double-headed arrows. In this space, a discontinuous change in the jump direction onto the continuous space takes place. The one-dimensional continuous space is shown as a solid curve with single-headed arrow. An example of this behavior is the freewheeling



Simulation of Hybrid Dynamic Systems, Fig. 10 A sequence of mode changes. © The MathWorks, Inc.



Simulation of Hybrid Dynamic Systems, Fig. 11 Alternative scenario across a sequence of mode changes. © The MathWorks, Inc.

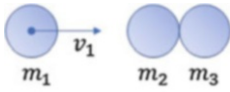
diode circuit in Fig. 4a where the flux of the inductor jumps to 0 when the switch is opened and the diode is still off. Figure 10 shows that in mode α_3 after the discontinuous jump, the state is in the inadmissible space, and a transition to mode α_4 takes place. In mode α_4 , behavior evolves according to the vector field lines with the initial state being the state that resulted from the discontinuous jump in mode α_3 . The behavior evolution in α_4 then reaches an inadmissible area based on time exceeding a threshold value, and further mode transitions may happen.

Figure 11 illustrates an alternative scenario for the sequence of mode changes in Fig. 10. In Fig. 11, the inadmissible space of α_3 is larger such that the initial state after transition from mode α_2 is in the admissible space already. Now, before the discontinuous jump in α_3 takes place, the transition to mode α_4 occurs, and behavior evolution in mode α_4 starts from a different initial value.

Note that the discontinuous change of the behavior state upon entering mode α_3 is part of the behavior specified for that mode instead of a separately specified reinitialization. This is well suited for models where the dynamics are modeled by a system of differential and algebraic equations (DAE). The algebraic part of these

equations defines the jump space, whereas the differential equation part defines the continuous-time behavior evolution space. The continuous behavior evolves on a manifold in the generalized state space, while the jump space corresponds to a projection onto this manifold (Mosterman 2002; van der Schaft and Schumacher 1996; Vergheze et al. 1981).

Alternatively, discontinuous change in the behavior state may be specified when a transition takes place from one mode to the next. For example, the bouncing ball model in Fig. 3 specifies a discontinuous change in velocity when a transition from one mode to the next (albeit the same) mode takes place. From a physics perspective, it may be preferred to have a mode associated with a discontinuity so the configurations of phenomena that cause the discontinuity are clear and to support compositionality. For the bouncing ball model, there is no model of the floor as a source of 0 velocity, while in general in collision models, this is preferred. Figure 12 shows Newton's cradle where in case of a perfectly elastic collision (a coefficient of restitution of 1 for the difference in velocities after and before the collision: $\Delta v^+ = -\Delta v$) with equal masses $m_1 = m_2 = m_3$, the momentum of m_1 first fully transfers to m_2 and



Simulation of Hybrid Dynamic Systems, Fig. 12 A series of reinitializations based on redistribution of momentum. © The MathWorks, Inc.

immediately following to m_3 . By modeling the masses and the collision phenomenon between the colliding masses (m_1 collides with m_2 and m_2 collides with m_3), additional masses are easily added (though great care must be taken when masses are not equal or coefficients of restitution are not 1 Mosterman (2007b)). Note that this model results in a sequence of projections, each of which requires a change in physical state before a new mode of continuous behavior is reached.

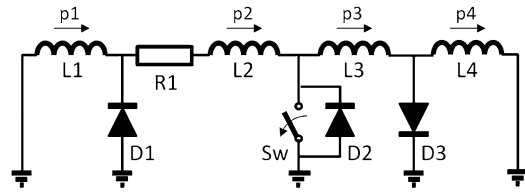
A Mode Transition and Reinitialization

Example

Figure 10 illustrates three classes of mode behavior:

- Continuous evolution with a state event in mode α_1 and a time event in mode α_4
- Immediate consecutive transition in mode α_2
- Transition after an instantaneous projection in mode α_3

The electric circuit in Fig. 13 is an example with physical behavior that can be modeled according to each of these. The four inductors ($L1$ through $L4$) store an initial flux ($p1$ through $p4$) that corresponds to each of their currents ($I1$ through $I4$) by the equation $I = \frac{p}{L}$. The resistor, $R1$, follows Ohm's law and creates a voltage difference (positive from the left to right connection), V , based on the current through $R1$, I , as $V = I \cdot R1$. The three diodes ($D1$ through $D3$) either limit a positive voltage difference to 0 (the *on* state) or a negative current to 0 (the *off* state). Similarly, the switch, Sw , enforces a 0 voltage difference when in its *closed* state or a 0 current when in its *open* state. Initially, the switch is in its open state.

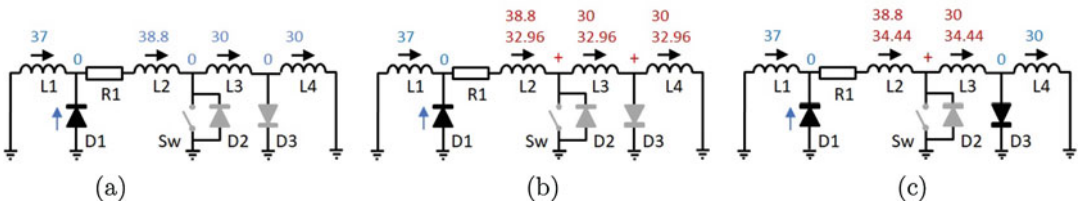


Simulation of Hybrid Dynamic Systems, Fig. 13 Electric circuit with different types of discontinuities. © The MathWorks, Inc.

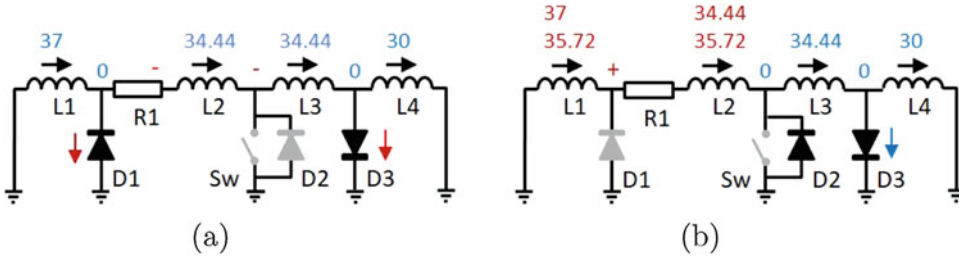
Assume the inductances $L1$, $L2$, $L3$, and $L4$ to be equal to 1.

Consider the scenario that starts with a time event at t_{switch} when the switch is opened while the flux distribution is $p1 = 37$, $p2 = 38.8$, $p3 = 30$, and $p4 = 30$, as shown in Fig. 14a. With the switch closed, the voltages at each of the junctions are 0. The difference in current through $L1$ and $L2$ results in a voltage across $R1$ that is balanced by a corresponding but negative voltage across $L2$. The open switch results in an inconsistent physical state (flux distribution) as opening the branch forces $p2$, $p3$, and $p4$ to be equal (given that $L2 = L3 = L4$). The increase in $p4$ induces a voltage spike as indicated by the + sign in Fig. 14b which causes diode $D3$ to come on and force the junction to a 0 voltage instead as illustrated in Fig. 14c. Because $D3$ comes on as a result of the projected change in flux distribution in Fig. 14b, that flux distribution is never achieved, and redistribution in Fig. 14c is determined based on the original flux distribution in Fig. 14a. Since the intermediate mode in Fig. 14b does not affect the ultimate physical state, it is called a *mythical mode* (Mosterman et al. 1998b; Nishida and Doshita 1987).

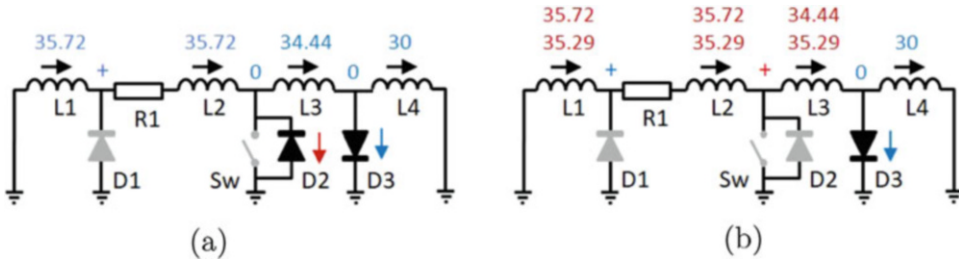
Once a consistent flux distribution is determined, it can be adopted as the current state. Because physical in nature, redistributing fluxes is modeled to require an infinitesimal period of time, ϵ , and the event iteration resumes at this new point in time, $t_{\text{switch}} + \epsilon$. Figure 15a shows the adopted consistent state and the voltages and currents that correspond to the new state. Because $p1$ has become larger than $p2$, a negative current would flow through $D1$ and so the diode turns off. Also, the positive current flow through $R1$ that



Simulation of Hybrid Dynamic Systems, Fig. 14 When the switch opens, a new consistent flux distribution is determined. (a) $\langle t_{\text{switch}}, 0 \rangle$. (b) $\langle t_{\text{switch}}, 1 \rangle$. (c) $\langle t_{\text{switch}}, 2 \rangle$. © The MathWorks, Inc.



Simulation of Hybrid Dynamic Systems, Fig. 15 Time is advanced by ϵ to model the physical state (flux) change and a consecutive change follows. (a) $\langle t_{\text{switch}} + \epsilon, 0 \rangle$. (b) $\langle t_{\text{switch}} + \epsilon, 1 \rangle$. © The MathWorks, Inc.



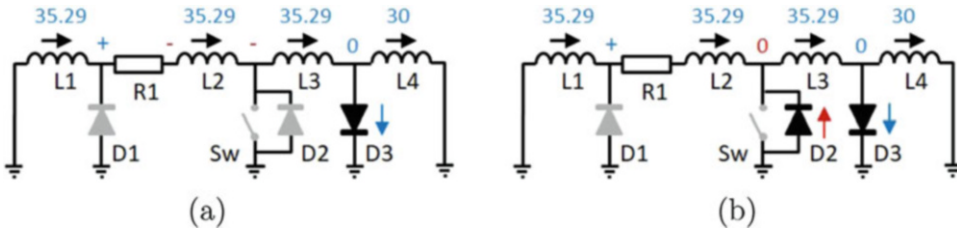
Simulation of Hybrid Dynamic Systems, Fig. 16 Time is advanced by another ϵ to model the physical state (flux) change to introduce a “stutter”. (a) $\langle t_{\text{switch}} + 2\epsilon, 0 \rangle$. (b) $\langle t_{\text{switch}} + 2\epsilon, 1 \rangle$. © The MathWorks, Inc.

is induced by p_2 causes a positive voltage drop. Because the junction to the left of R_1 is at 0 volt, the right-hand port of R_1 is negative. With $L_2 = L_3$ and D_2 off, the voltage drop across both L_2 and L_3 must be equal, which is half the positive voltage drop across R_1 . This negative voltage causes D_2 to come on. Figure 15b shows the new mode with the newly required flux distribution because L_1 and L_2 are now coupled, while L_3 has become decoupled.

Since no further mode changes occur, time is advanced by an infinitesimal amount, and the flux redistribution is adopted as the next physical state. Figure 16a shows this mode and flux distribution as the starting point for the next event iteration. Because the changed flux of L_2

exceeds that of L_3 , D_2 would have a negative current and so D_2 turns off. As a result, L_3 is coupled with L_1 and L_2 , and Fig. 16b shows the corresponding flux redistribution. At this point, no further mode changes occur, time is advanced by an infinitesimal amount, and the redistributed fluxes are adopted.

Figure 17a shows the flux distribution at $t_{\text{switch}} + 3\epsilon$ as well as qualitative values for the voltages and currents. The current through R_1 causes a positive voltage drop that is equally distributed as negative voltage across each of the inductors L_1 , L_2 , and L_3 such that the sum of voltages equals 0. This causes D_2 to turn on, as shown in Fig. 17b. At this point, no further mode changes occur and no flux redistribution



Simulation of Hybrid Dynamic Systems, Fig. 17 A consistent physical state leads to another mode transition with the same consistent physical state. (a) $(t_{\text{switch}} + 3\epsilon, 0)$. (b) $(t_{\text{switch}} + 3\epsilon, 1)$. © The MathWorks, Inc.

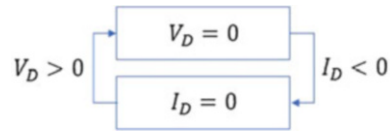
is required. Therefore, the sequence of mode changes, and reinitialization terminates so that continuous-time behavior evolution resumes.

Modeling Diode Behavior

Figure 18 shows the diode model used for simulation of the electrical circuit in Fig. 13. Note that the switching conditions do not include 0 (i.e., the voltage must exceed 0 or the current must fall below 0 before being limited to 0). Otherwise, consider the case where a diode is in its *on* state and enforcing a 0 voltage when it is determined that the current would fall below 0. In this situation, the diode would change its state to *off*, but without any changes in the physical state of the circuit, the corresponding voltage would still be 0. Hence, if the switching conditions would include 0, the diode would immediately switch back to its *off* state, and an infinite loop of state changes from *on* to *off* and back arises.

Mathematically, not including the boundary of a switching area may be complicating since the point in time of switching and the limit value at this point are not well defined. A threshold value that must be exceeded eliminates this concern (e.g., switching conditions $I_D \leq I_{th}$ instead of $I_D \leq 0$ and $V_D \geq V_{th}$ instead of $V_D \geq 0$ in Fig. 4b). In simulation, however, the threshold values are a numerical effect when the switching conditions $I_D < 0$ and $V_D > 0$ are used and a threshold value is not strictly necessary.

Also note that the diode model in Fig. 18 is formulated as a finite state machine, whereas sequential logic is not necessary to capture the desired behavior. For example, in a complementarity formulation (van der Schaft and Schumacher 1996), an ideal diode is modeled



Simulation of Hybrid Dynamic Systems, Fig. 18 Finite state machine model of a diode. © The MathWorks, Inc.

by an equality that requires either I_D or V_D to be 0 while complementing the equation with inequalities that require either the voltage to be less than or equal to 0 or the current to be larger than or equal to 0:

$$\begin{aligned} 0 &= V_D I_D \\ -V_D &\geq 0 \\ I_D &\geq 0 \end{aligned} \quad (5)$$

Complementarity formulations are effective in modeling discontinuous behavior of different forms (e.g., rigid body mechanics (Pfeiffer and Glocker 1996)). The use of sequential logic covers a broader range of phenomena in hybrid dynamic systems, though, and sometimes is required (e.g., a latched bitank system (Mosterman and Biswas 1995)).

A Hyperdense Semantic Domain and Numerical Simulation

As the example in Fig. 13 shows, the semantic domain of piecewise continuous models of physical systems requires three aspects:

- A continuous domain for continuous-time behavior evolution

- An infinitesimal domain for discontinuous changes in physical state
- An ordered domain for mode changes

The ability to separate discontinuous change in physical state from mode changes is illustrated in Fig. 14c. With the flux distribution at t_{switch} , the current through $D1$ keeps $D1$ on. The current at $t_{\text{switch}} + \epsilon$, however, is such that $D1$ turns off. If the infinitesimal time step would not be required to advance the flux distribution, $D1$ would turn off before $D2$ turns on, and a different behavior from what is shown in Fig. 15b would be generated.

The set of hyperreals, ${}^*\mathbb{R}$, of nonstandard analysis (Keisler 2012) embeds infinitesimals in the set of reals, $\mathbb{R}$, and so ${}^*\mathbb{R}$ supports the semantic domain for continuous-time behavior evolution and infinitesimal time steps. In addition, the set of integers support the ordered domain of mode changes. Hence, ${}^*\mathbb{R} \times \mathbb{N}$ provides a *hyperdense* semantic domain that supports the necessary detail to separately evaluate the discontinuous and mode change behavior (Mosterman et al. 2014).

Numerical simulation requires discrete-time steps to evolve behavior on the reals, $\mathbb{R}$, while the infinitesimal time advances can be captured by an integer count of infinitesimal steps during mode transition sequences. The resulting domain for numerical simulation becomes three dimensional as $\mathbb{R} \times \mathbb{N} \times \mathbb{N}$. For example, the mathematical moment of evaluation in Fig. 14a is $\langle t_{\text{switch}}, 0 \rangle$ with the numerical simulation step being identified as $\langle t_{\text{switch}}, 0, 0 \rangle$. Likewise, the mathematical moment of evaluation in Fig. 17b is $\langle t_{\text{switch}} + 3\epsilon, 1 \rangle$ with the numerical simulation step being identified as $\langle t_{\text{switch}}, 3, 1 \rangle$.

The results of a simulation with HYBR-SIM (Mosterman 2002) where $R1 = L1 = L2 = L3 = L4 = 1$ from time 0 to 0.5s are shown in Fig. 19. Before the switch is opened at $t = t_{\text{switch}} = 0.2s$, the flux $p2$ decreases because of the dissipative effect of $R1$. After the switch is opened and the mode transition sequence has converged, the fluxes $p1$ and $p2$ decrease equally (overlapping traces in the graph) because $D1$ is *off*. Figure 20 shows the mode transition

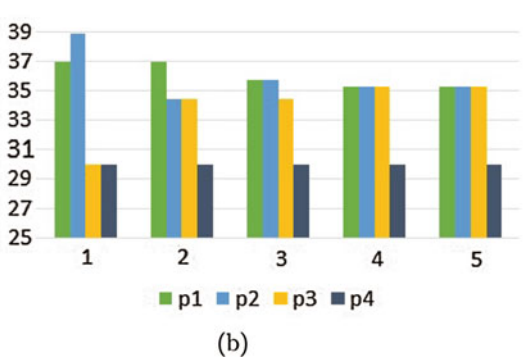
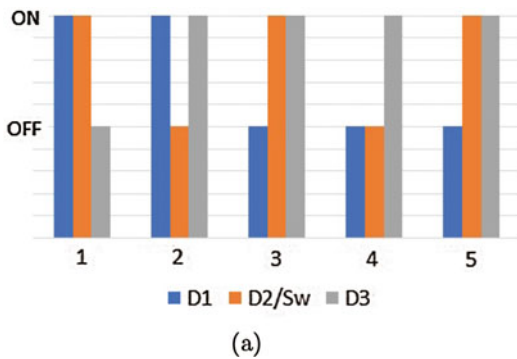
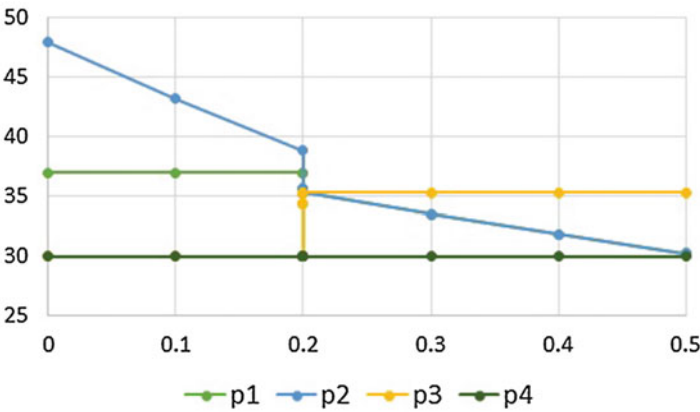
sequence at $t = 0.2s$ in detail where the indices represent the evolution tuples as 1 : $\langle 0.2, 0, 0 \rangle$, 2 : $\langle 0.2, \epsilon, 0 \rangle$, 3 : $\langle 0.2, 2\epsilon, 0 \rangle$, 4 : $\langle 0.2, 3\epsilon, 0 \rangle$, and finally 5 : $\langle 0.2, 3\epsilon, 1 \rangle$. Figure 20a shows the consistent modes for each point in time including the initial mode when the switch is still closed. The overall mode is comprised of a combination of the discrete state of three junctions where the first and last junction are fully determined by the state of the corresponding diodes, $D1$ and $D3$, respectively. The state of the second junction depends on the combination of the switch, Sw , and diode, $D2$. If the switch is closed or the diode is *on*, the junction is *on*, whereas the junction is *off* otherwise. Figure 20b shows the consistent flux distributions for each of the modes in Fig. 20a with the initial distribution of the fluxes immediately before the switch is opened (t_{switch}). Note that the intermediate modes (before an accepted flux distribution is reached) are not logged in simulation; only values in time as simulated hyperreals (i.e., a tuple of time and number of infinitesimal steps from a point in real time) are shown.

Pathological Behavior

Simulation of mode transition behavior may encounter a number of behaviors that are challenging to handle. First, the hybrid dynamic system may exhibit sliding behavior as illustrated in Fig. 21. When the behavior reaches the switching (inadmissible) area (where the boundary is included in the switching condition) in mode α_1 , a transition is made to mode α_2 . The behavior in mode α_2 starts on the boundary of the switching (inadmissible) area where the boundary is not included. As soon as continuous-time simulation restarts, the mode transition back to α_1 occurs. After a similarly small time step, the dynamic field lines push the behavior in mode α_1 back to the switching boundary, and repeated switching occurs with the minimum step size that the numerical simulation allows. Such behavior is, for example, the purview of *sliding mode control* (such as used in antilock braking systems), and special simulation algorithms (Mosterman et al. 1999) can be used to filter out the fast

Simulation of Hybrid Dynamic Systems,

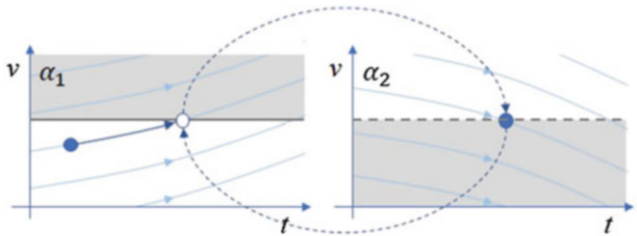
Fig. 19 Simulation of flux distribution over time. © The MathWorks, Inc.



Simulation of Hybrid Dynamic Systems, Fig. 20 Simulation results of the mode transition sequence. (a) Mode transitions. (b) Flux redistributions. © The MathWorks, Inc.

Simulation of Hybrid Dynamic Systems,

Fig. 21 Sliding mode behavior. © The MathWorks, Inc.



switching behavior while retaining the slow dynamic behavior along the switching boundary. Second, in case of overlapping switching areas in Fig. 21, as soon as the switching boundary in mode α_1 is reached, the behavior enters an infinite loop of mode transitions that occur in 0 time. Hence, simulation of the dynamic behavior halts, which is typically undesirable (e.g., because this is inconsistent with observed behavior in the

physical world). This behavior may be difficult to determine as present in a hybrid dynamic system model and often cannot be detected other than when exhibited during simulation. Third, mode transitions may occur at progressively smaller time steps such as described for the bouncing ball model in Fig. 3. In an ideal mathematical model, time converges to a limit point. In a numerical simulation, behavior may be

ill defined when the numerical time step moves past the limit point. Note that the existence of a limit point may be difficult to detect, even during simulation.

Multiparadigm Modeling

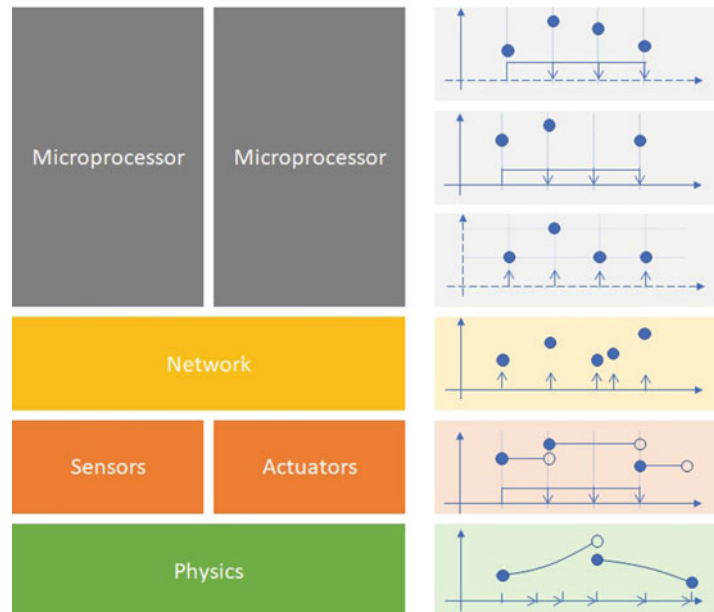
Modern engineered systems increasingly rely on computational elements (e.g., microprocessors, microcontrollers, and field-programmable gate arrays) and a communication infrastructure (e.g., wired or wireless networks). Even if such engineered systems are not intrinsically hybrid dynamic systems, the physics are often best represented at a macroscopic level by models with continuous-time semantics, and the computation is often best represented by models with discrete-time or discrete-event semantics. A structure of such systems with the various different components is shown in Fig. 22. At the bottom is the physical world. Sensors and actuators connect the physical world with the digital world via built-in computational elements. The network connectivity of these sensors and actuators enables communication with other sensors and actuators as well as

powerful microprocessors or other computational elements.

Different formalisms are typically required to model the behavior of the various different components. At the microprocessor level, program code (semantic domain of Fig. 5a), synchronous data flow and difference equations (semantic domain of Fig. 5b), as well as state transition diagrams (semantic domain of Fig. 6b) are often used. Modeling network behavior generally requires the ability to capture different event densities over time for which discrete-event systems are preferred (semantic domain of Fig. 7a) (Cassandras et al. 2006; Cervin et al. 2003; Zeigler et al. 2000). Because of the interaction with the continuous-time models of the physical world, sensors and actuators are often modeled as discrete time with zero-order hold (semantic domain of Fig. 5c). Finally, the macroscopic behavior of physical systems is well represented by differential equations with discontinuous changes that occur at specific points in time (semantic domain of Fig. 7c).

The combination of this scala of modeling formalisms with the variety of semantic domains to be used for simulation is the purview of multiparadigm modeling (Mosterman and Vangheluwe

Simulation of Hybrid Dynamic Systems,
Fig. 22 Multiple semantic domains in engineered system modeling. © The MathWorks, Inc.



2004). Hybrid dynamic systems are thus fundamental to multiparadigm modeling where hybrid means combining different semantic domains.

Behavior Generation Engines

In terms of efficient execution, handling continuous time is the first and foremost challenge. Continuous-time behavior progresses along the gradient of differential equations. When thresholds are exceeded, zero-crossing events are detected and then located, and their effect is evaluated before time progresses further. Because of the progression of state values along time, this is called time-driven execution. In contrast, discrete-event behavior schedules events in the future by keeping an event calendar. The event that is scheduled earliest in time is removed from the calendar, current time is set to the event time, and the event is processed. This processing may cause further events to be scheduled or already scheduled events to be canceled (removed from the calendar). Once processing completes, the current time is set to the time of the earliest scheduled event, and the event is processed. Because time evolves in discrete steps based on scheduled events, this is called an event-driven execution (Mosterman 2007a).

Figure 23 depicts a time-driven execution engine on the left-hand side and an event-driven execution engine on the right-hand side. The bottom traces represent continuous-time evolution on the left and discrete-event evolution on the right. Note that at event times (e.g., zero-crossing events or scheduled discrete events), multiple evaluation steps may take place that may be scheduled as events in turn. Also, in case of piecewise continuous behavior, a series of infinitesimal steps may be scheduled before time-driven execution resumes.

Statically scheduled events because of periodic behavior (e.g., discrete time with a single or multiple rates and synchronous data flow) can either be included in a time-driven or an event-driven execution engine. Figure 23 shows discrete-time behavior as part of the time-driven execution engine because the zero-order hold

behavior closely aligns with the continuous-time behavior and is typically used because of interacting behavior. Without hold behavior, complicated semantic challenges may arise in interacting with minor time steps of numerical solvers (Denckla and Mosterman 2008). Otherwise, the discrete-time behavior schedules a time event, and the numerical solver will evaluate the differential equations at that point in time, then evaluate the discrete-time operations, and resume continuous-time behavior.

Figure 23 shows synchronous behavior as part of the event-driven execution engine because of the discretized time domain. Also, data flow comes in statically scheduled and dynamically scheduled forms and hence is more closely related to discrete-event systems.

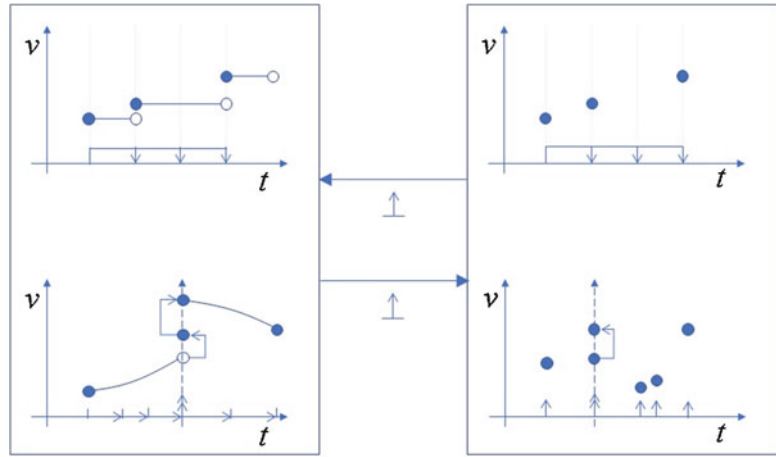
While the architecture in Fig. 23 allows the most efficient simulation mechanism (time driven vs. event driven) for each evolution domain, the moments of interaction must be handled efficiently as well. In one form, the interaction may consist of the time-driven model part scheduling or canceling an event for the event-driven model part (e.g., when the strength of a communicating signal falls below a threshold level, a message may be sent to a transmitter). In another form, the event-driven model part may schedule an event for the time-driven model part (e.g., a network message may set a threshold value for a valve to open). As a result, when an interacting event is scheduled or canceled, the behavior evolution of the receiving model part may change.

Two basic approaches can be employed to handle the interaction effects:

- With *conservative execution*, the engines operate in lock step. The event-driven execution engine may change its current time to be ahead of the current time of the time-driven execution engine but first stores the full state of the event-driven execution engine. The time-driven execution engine then catches up before the event-driven engine makes another step and the process repeats. If the time-driven model part generates an interacting event while in the process of catching up, the event-driven execution engine restores its

Simulation of Hybrid Dynamic Systems,

Fig. 23 Time-driven and event-driven execution engines. © The MathWorks, Inc.



full state, the interacting event is added to the event calendar, and the interacting event is processed next.

- With *optimistic execution*, the engines are allowed to process more than one step ahead of each other. For example, it may be efficient both for the event-driven engine and the time-driven engine if one processes a number of events before the other catches up. In this scheme, it becomes key to determine when the full engine state should be stored to enable quick reconstruction of the state at which an interacting event occurred. In advanced methods, an intermediate state may be reconstructed in approximation by an interpolation polynomial (which may be more efficient than reconstructing a state via simulation).
- The cross product of evolution domains such as a continuous domain combined with an integer domain.
- A combination of different semantic domains such as discrete-time and continuous-time behavior.
- A combination of different execution engines, for example, an event-driven engine combined with a time-driven engine.

Generating an event from a continuous-time behavior via zero-crossing detection and location is both challenging to be sound and to be efficient. Generating a new mode of continuous-time behavior with a consistent continuous-time state from discrete changes is challenging as well. In particular, among the richest of hybrid dynamic system behaviors is the piecewise continuous semantic domain. With a continuous value domain, a three-dimensional hyperdense evolution domain emerges to account for sequences of state reinitialization and mode transition behavior.

Note that modeling mode transitions as either occurring in 0 time or in infinitesimal time has been shown to be an important consideration beyond simulation, namely, in reasoning about discontinuous change, with corresponding direct and approximate reasoning methods, respectively (Nishida and Doshita 1987).

Conclusions

Hybrid dynamic systems may refer to various combinations of behavior and execution semantics:

- A continuous evolution domain with a combination of continuous and discrete behavior in the value domain.

Much progress has been made in hybrid dynamic system simulation. Still, open challenges remain. These include how to:

- Solve or resolve some of the pathological behaviors.
 - Statically (i.e., before simulation) determine whether the simulation may reach an infinite loop of 0 time mode transitions.
 - Find the limit points for converging time if present.
 - Develop robust simulation algorithms for higher-dimensional sliding behavior.
- Simulate impulsive behavior (e.g., support comparison of impulsive quantities such as voltage spikes and impulsive forces).

As a caution in closing, combining continuous behavior with discrete changes results in behavior that is highly sensitive to perturbations in the initial state and parameters of the model but also to the solver parameters. Without a fundamental solution, good practice is to check sensitivity effects by using different solvers and parameters.

Cross-References

- [Control Hierarchy of Large Processing Plants: An Overview](#)
- [Discrete Event Systems and Hybrid Systems, Connections Between](#)
- [Hybrid Dynamical Systems, Feedback Control of](#)
- [Modeling Hybrid Systems](#)
- [Multi-domain Modeling and Simulation](#)
- [Sliding Mode Control: Finite-Time Observation and Regulation](#)
- [Switching Adaptive Control](#)

Bibliography

- Alur R, Dill DL (1994) A theory of timed automata. *Theor Comput Sci* 126:183–235
- Benveniste A, Caspi P, Edwards SA, Halbwachs N, Guernic PL, de Simone R (2003) The synchronous languages twelve years later. *Proc IEEE* 91(1):64–83

- Breedveld PC (1996) The context-dependent trade-off between conceptual and computational complexity illustrated by the modeling and simulation of colliding objects. In: CESA '96 IMACS Multiconference, Ecole Centrale de Lille, Lille
- Cassandras CG, Clune MI, Mosterman PJ (2006) Hybrid system simulation with simevents. In: Cassandras C, Giua A, Seatzu C, Zaytoon J (eds) *Analysis and design of hybrid systems 2006*. Elsevier, Amsterdam, pp 267–269
- Cellier FE (1979) Combined continuous/discrete system simulation by use of digital computers: techniques and tools. Ph.D. dissertation, ETH, Zurich
- Cellier FE, Kofman E (2006) *Continuous System Simulation*. Springer, Berlin, Heidelberg
- Cervin A, Henriksson D, Lincoln B, Eker J, Årzén KE (2003) How does control timing affect performance? analysis and simulation of timing using jitterbug and truetype. *IEEE Control Syst Mag* 23(3):16–30
- Denckla B, Mosterman PJ (2008) Stream- and state-based semantics of hierarchy in block diagrams. In: *Proceedings of the 17th IFAC World Congress, Seoul*, pp 7955–7960
- Espósito JM, Kumar V, Pappas GJ (2001) Accurate event detection for simulating hybrid systems. In: *Proceedings of the 4th International Workshop on Hybrid Systems: Computation and Control, HSCC '01*. Springer, London, UK, pp 204–217
- Keisler HJ (2012) *Elementary calculus: an infinitesimal approach*, 3rd edn. Prindle, Weber and Schmidt, Dover, Mineola
- MathWorks® (2019) *MATLAB® User's Guide*
- Mosterman PJ (2002) HYBRISIM—a modeling and simulation environment for hybrid bond graphs. *J Syst Control Eng* 216(1):35–46
- Mosterman PJ (2007a) Hybrid dynamic systems: modeling and execution. In: Fishwick PA (ed) *Handbook of dynamic system modeling*. CRC Press, Boca Raton, chap 15, pp 15-1–15-26
- Mosterman PJ (2007b) On the normal component of centralized frictionless collision sequences. *ASME J Appl Mech* 74(5):908–915
- Mosterman PJ, Biswas G (1995) Behavior generation using model switching: a hybrid bond graph modeling technique. In: Cellier FE, Granda JJ (eds) 1995 International conference on bond graph modeling and simulation (ICBGM '95). Society for Computer Simulation, Simulation Councils, Inc., Las Vegas, no. 1 in *Simulation*, vol 27, pp 177–182
- Mosterman PJ, Vangheluwe H (2004) Computer automated multi-paradigm modeling: an introduction. *Simul Trans Soc Model Simul* 80:433–450
- Mosterman PJ, Biswas G, Sztipanovits J (1998a) A hybrid modeling and verification paradigm for embedded control systems. *Control Eng Pract* 6:511–521
- Mosterman PJ, Zhao F, Biswas G (1998b) An ontology for transitions in physical dynamic systems. In: AAAI98, pp 219–224
- Mosterman PJ, Zhao F, Biswas G (1999) Sliding mode model semantics and simulation for hybrid systems. In:

- Antsaklis P, Kohn W, Lemmon M, Nerode A, Sastry S (eds) Hybrid systems V. Lecture notes in computer science. Springer, Berlin/Heidelberg, pp 218–237
- Mosterman PJ, Simko G, Zander J, Han Z (2014) A hyperdense semantic domain for hybrid dynamic systems to model different classes of discontinuities. In: Proceedings of the 17th international conference on hybrid systems: computation and control, HSCC '14. ACM, New York, pp 83–92
- Murata T (1989) Petri nets: Properties, analysis and applications. Proc IEEE 77(4):541–580
- Nishida T, Doshita S (1987) Reasoning about discontinuous change. In: Proceedings AAAI-87, Seattle, Washington, pp 643–648
- Park T, Barton PI (1996) State event location in differential-algebraic models. ACM Trans Model Comput Simul 6(2):137–165
- Pfeiffer F, Glocker C (1996) Multibody dynamics with unilateral contacts. John Wiley & Sons, Inc., New York
- van der Schaft AJ, Schumacher JM (1996) The complementary-slackness of hybrid systems. Math Control Signals Syst 9:266–301
- Verghese GC, Lévy BC, Kailath T (1981) A generalized state-space for singular systems. IEEE Trans Autom Control 26(4):811–831
- Zeigler BP, Kim TG, Praehofer H (2000) Theory of modeling and simulation, 2nd edn. Academic Press, Inc., Orlando
- Zhang F, Yeddapanudi M, Mosterman PJ (2008) Zero-crossing location and detection algorithms for hybrid system simulation. In: Proceedings of the 17th IFAC world congress, Seoul, pp 7967–7972

Single-Photon Coherent Feedback Control and Filtering

Guofeng Zhang

Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, China

Abstract

A light field is in a k -photon state if it contains *exactly* k photons. Due to their highly quantum nature, photon states hold promising applications in quantum communication, computing, metrology, and simulations. Recently, there has been rapidly growing interest in the generation and manipulation of various photon states. A new and important problem in the field of control engineering is: How to analyze

and synthesize quantum systems driven by photon states to achieve pre-specified control performance? In this survey, we introduce single-photon states and show how a quantum linear system processes a single-photon input and how to use linear coherent feedback networks to shape the temporal pulse of a single photon. A single-photon filter is also presented. (An extended version of this survey can be found at [arXiv:1902.10961](https://arxiv.org/abs/1902.10961).)

Keywords

Quantum control systems · Filtering · Coherent feedback control

Notation $|0\rangle$ denotes the vacuum field state. Given a column vector of operators or complex numbers $X = [x_1, \dots, x_n]^\top$, the adjoint operator or complex conjugate of X is denoted by $X^\# = [x_1^*, \dots, x_n^*]^\top$. Let $X^\dagger = (X^\#)^\top$ and $\check{X} = [X^\top \ X^\dagger]^\top$. The commutator between operators A and B is defined to be $[A, B] = AB - BA$. Given two constant matrices $U, V \in \mathbb{C}^{r \times k}$, a doubled-up matrix $\Delta(U, V)$ is defined as $\Delta(U, V) = [U \ V; V^\# \ U^\#]$. Finally, $\delta(t - r)$ is the Dirac delta function, and $\xi[i\omega]$ is the complex domain function obtained after Fourier transforming a time-domain function $\xi(t)$.

Single-Photon States

Let $b(t)$ denote the annihilation operator of a light field. Then a continuous-mode single-photon state $|1_\xi\rangle$ of the field can be defined as $|1_\xi\rangle \equiv \mathbf{B}^*(\xi) \triangleq \int_{-\infty}^{\infty} b^*(t)\xi(t)dt |0\rangle$, where $\xi(t)$ is an L^2 integrable function satisfying $\int_{-\infty}^{\infty} |\xi(t)|^2 dt = 1$ and $b^*(t)$ (the adjoint of $b(t)$) is the creation operator of the field. Under the state $|1_\xi\rangle$, the quantum stochastic process $b(t)$ has zero mean, and the covariance function, defined by $R(t, r) \triangleq \langle 1_\xi | \check{b}(t)\check{b}^\dagger(r) | 1_\xi \rangle$, satisfies

$$R(t, r) = \delta(t - r) \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \xi(r)^* \xi(t) & 0 \\ 0 & \xi(r) \xi(t)^* \end{bmatrix}.$$

For example, the Gaussian pulse shape of a single-photon state $|1_\xi\rangle$ can be given by

$$\xi(t) = \left(\frac{\Omega^2}{2\pi}\right)^{\frac{1}{4}} \exp\left(-\frac{\Omega^2}{4}(t - \tau)^2\right), \quad (1)$$

where τ is the photon peak arrival time. Applying Fourier transform to $\xi(t)$, we get

$$|\xi[i\omega]|^2 = \frac{1}{\sqrt{2\pi}(\Omega/2)} \exp\left(-\frac{\omega^2}{2(\Omega/2)^2}\right).$$

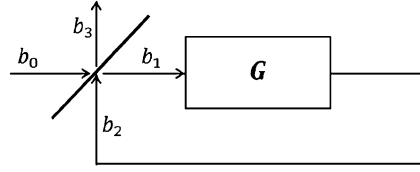
Hence, Ω is the frequency bandwidth of the single-photon wavepacket.

Linear Response to Single-Photon States

In this section, we discuss how a quantum linear system G processes single-photon states. Let G be an optical cavity having a single internal mode (a quantum harmonic oscillator represented by its annihilation operator a) and interacting with an external light field represented by its annihilation operator b . Because a is an internal mode, it and its adjoint operator a^* satisfy the canonical commutation relation $[a, a^*] = 1$, in contrast to the singular commutation relation $[b(t), b(r)^*] = \delta(t - r)$ for the free propagating field. Let ω_c be the detuned frequency between the resonant frequency of the internal mode a and the central frequency of the external light field b . Let κ be the half linewidth of the cavity. Then the dynamics of this system can be described by $da(t) = -(i\omega_c + \frac{\kappa}{2})a(t)dt - \sqrt{\kappa}dB(t)$ and $dB_{\text{out}}(t) = \sqrt{\kappa}a(t)dt + dB(t)$, where $B(t) = \int_0^t b(\tau)d\tau$ is the integrated field input operator (a quantum Wiener process). More discussions on open quantum systems can be found in, e.g., Gardiner and Zoller (2004), Wiseman and Milburn (2009), and Zhang et al. (2018).

Theorem 1 (Zhang and James 2013, Proposition 2)

Assume there is one input field which is in the single photon state $|1_\xi\rangle$. Then the steady-state output field state for the linear quantum system



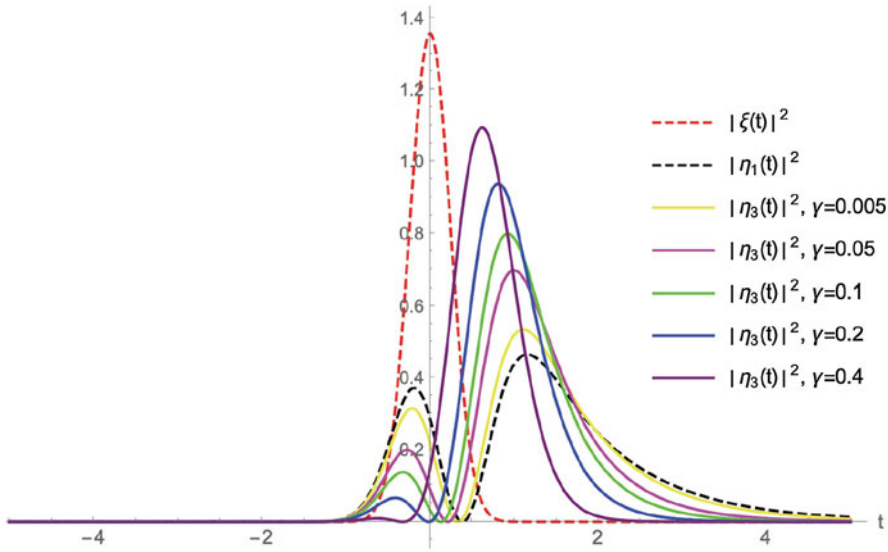
Single-Photon Coherent Feedback Control and Filtering, Fig. 1 A linear quantum feedback network composed of an optical cavity and a beam splitter

G is $\rho_{\text{out}} = (\mathbf{B}^*(\xi_{\text{out}}^-) - \mathbf{B}(\xi_{\text{out}}^+))\rho_{\text{field},g}(\mathbf{B}^*(\xi_{\text{out}}^-) - \mathbf{B}(\xi_{\text{out}}^+))^*$, where $\Xi_G[s]$ is the transfer function of the system G , $\Delta(\xi_{\text{out}}^-[s], \xi_{\text{out}}^+[s]) = \Xi_G[s]\Delta(\xi[s], 0)$, and $\rho_{\text{field},g}$ is the density operator for the output field with zero mean and covariance function $R_{\text{out}}[i\omega] = \Xi_G[i\omega]R_{\text{in}}[i\omega]\Xi_G[i\omega]^\dagger$ with $R_{\text{in}}[i\omega] = [1 \ 0; 0 \ 1]$. In particular, for an empty optical cavity, we have $\xi_{\text{out}}^+[s] \equiv 0$ and $R_{\text{out}}[i\omega] \equiv R_{\text{in}}[i\omega]$; in other words, the steady-state output is a single-photon state $|1_{\xi_{\text{out}}^-}\rangle$.

Single-Photon Pulse Shaping

In this section, we show how a coherent feedback network can be constructed to manipulate the temporal pulse shape of a single-photon state.

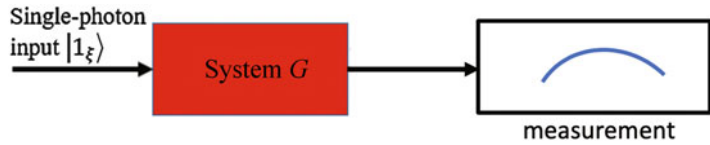
If an optical cavity is driven by a single-photon state $|1_\xi\rangle$, by Theorem 1, the output pulse shape in the frequency domain is $\eta_1[i\omega] = \frac{i(\omega + \omega_c) - \frac{\kappa}{2}}{i(\omega + \omega_c) + \frac{\kappa}{2}}\xi[i\omega]$. Now we put the cavity in a feedback network closed by a beam splitter, as shown in Fig. 1. Let the beam splitter be defined by a two-by-two matrix $S_{\text{BS}} = [\sqrt{\gamma} \ e^{-i\phi}\sqrt{1-\gamma}; -e^{i\phi}\sqrt{1-\gamma} \ \sqrt{\gamma}]$ with $0 < \gamma < 1$. The whole system from b_0 to b_3 in Fig. 1 is still a quantum linear system that is driven by the single-photon state $|1_\xi\rangle$ for the input b_0 . By Theorem 1, the pulse shape for the output field b_3 is $\eta_3[i\omega] = (-\frac{1-\sqrt{\gamma}}{1+\sqrt{\gamma}}(\omega + \omega_c)i + \frac{\kappa}{2})/(\frac{1-\sqrt{\gamma}}{1+\sqrt{\gamma}}(\omega + \omega_c)i + \frac{\kappa}{2})\xi[i\omega]$. Fix $\tau = 0$ and $\Omega = 2.92$ for the Gaussian single-photon state (1) and $\omega_c = 0$ and $\kappa = 1$ for the optical cavity (Fig. 2). The temporal pulse shapes $\xi(t)$, $\eta_1(t)$ and $\eta_3(t)$ are plotted in Fig. 2.



Single-Photon Coherent Feedback Control and Filtering, Fig. 2 $|\xi(t)|^2$ denotes the detection probability of input photon, $|\eta_1(t)|^2$ denotes the detection probability of output photon in the case of the optical cavity alone,

$|\eta_3(t)|^2$ are the detection probabilities of output photon in the coherent feedback network (Fig. 1) with different beamsplitter parameters γ

Single-Photon Coherent Feedback Control and Filtering, Fig. 3 Single-photon filtering



Single-Photon Filtering

As discussed above, a single-photon light field has statistical properties. Hence, it makes sense to study the filtering problem of a quantum system driven by a single-photon field. Single-photon filters were first derived in Gough et al. (2012), and their multiphoton version was developed in Song et al. (2016). The basic setup is given in Fig. 3. In Fig. 3, G is a quantum system which is driven by a single photon of pulse shape ξ . After interaction, the output field is measured, and the measurement outcome is $Y(t) = B_{\text{out}}(t) + B_{\text{out}}^*(t)$. Y_t enjoys the self-non-demolition property $[Y(t), Y(r)] = 0$ and the non-demolition property $[X(t), Y(r)] = 0$ for all $t_0 \leq r \leq t$, where X is an arbitrary system operator and t_0 is the time when the system and field start interaction. The quantum conditional expectation is defined as $\pi_t(X) \equiv \mathbb{E}[X(t)|\mathcal{Y}_t]$, where $\mathbb{E}$ denotes the expectation and the commutative von

Neumann algebra $\mathcal{Y}_t$ is generated by the past measurement observations $\{Y(s) : t_0 \leq s \leq t\}$. The *conditioned* system density operator $\rho(t)$ can be obtained by means of $\pi_t(X) = \text{Tr}[\rho(t)^\dagger X]$. It turns out that $\rho(t)$ is a solution to a system of stochastic differential equations, which is called the *quantum filter* whose exact form can be found in Gough et al. (2012, Eq. (42)).

Summary and Future Directions

In this section, we discuss three possible future research directions.

In this survey, the single photons are characterized in terms of their temporal pulse shapes, which are L^2 integrable functions. In this sense, these photon states are continuous-variable (CV) states. In addition to CV photonic states, there are discrete-variable (DV) photonic states, for example, number states and polarization states.

In Zhang and James (2013), the linear systems theory, summarized in section “[Linear Response to Single-Photon States](#),” was applied to derive the output of a quantum linear system driven by a coherent state and a single-photon state. If the pulse information is ignored and only the number of photons is counted, the results reduce to those in the physics literature; see, e.g. Bartley et al. (2012). It can be easily shown that the general framework still works if the coherent state is replaced by a squeezed-vacuum state. Nonetheless, it is unclear to what extent that the mathematical methods proposed for photonic CV states also work for DV photonic states.

In the section “[Single-Photon Pulse Shaping](#),” we have discussed the problem of single-photon pulse shaping by using a very simple example (see Fig. 1), where the system G is an optical cavity. Clearly, a passive quantum linear controller K can be added into the network in Fig. 1. Adding a passive quantum linear controller increases flexibility of the pulse shaping of the input single photon. There may be one of the future research directions.

Finally, in measurement-based feedback control of quantum systems, *real-time* implementation of a quantum filter is essential. In the case of a two-level system driven by a vacuum field, a system of three stochastic differential equations (SDEs) is sufficient for filtering. However, for a two-level system driven by a single photon, the single-photon filter consists of a system of 9 SDEs (Gough et al. 2012, Eq. (42)), and a two-photon filter consists of 30 SDEs (Song et al. 2016). Thus, numerically efficient and reliable implementation of single- or multi-photon filters is an important problem in the measurement-based feedback control of quantum systems driven by photons. There is presently very few pieces of work in the direction; a recent attempt using quantum information geometry theory can be found in Gao et al. (2019)

Cross-References

- [Conditioning of Quantum Open Systems](#)
- [Quantum Networks](#)

Acknowledgments I wish to thank financial support from the Hong Kong Research Grant Council under grants 15206915, 15208418, and 15203619.

References

- Bartley TJ, Donati G, Spring JB, Jin X-M, Barbieri M, Datta A, Smith B.J. and Walmsley IA (2012) Multiphoton state engineering by heralded interference between single photons and coherent states. *Phys Rev A* 86(4):043820
- Gao Q, Zhang GF, Petersen IR (2019) An exponential quantum projection filter for open quantum systems. *Automatica* 99:59–68
- Gardiner C, Zoller P (2004) *Quantum noise: a handbook of Markovian and non-Markovian Quantum stochastic methods with applications to quantum optics*. Springer, Berlin
- Gough JE, James MR, Nurdin HI and Combes J (2012) Quantum filtering for systems driven by fields in single-photon states or superposition of coherent states. *Phys Rev A* 86(4):043819
- Song HT, Zhang GF, Xi ZR (2016) Continuous-mode multiphoton filtering. *SIAM J Control Optim* 54(3):1602–1632
- Wiseman HM, Milburn GJ (2009) *Quantum measurement and control*. Cambridge University Press, Cambridge
- Zhang GF, Grivopoulos S, Petersen IR, Gough JE (2018) The Kalman decomposition for linear quantum systems. *IEEE Trans Autom Control* 63(2):331–346
- Zhang GF, James MR (2013) On the response of quantum linear systems to single photon input fields. *IEEE Trans Autom Control* 58(5):1221–1235

Singular Trajectories in Optimal Control

Bernard Bonnard¹ and Monique Chyba²

¹Institute of Mathematics, University of Burgundy, Dijon, France

²University of Hawaii-Manoa, Manoa, HI, USA

Abstract

Singular trajectories arise in optimal control as singularities of the end-point mapping. Their importance has long been recognized, at first in the Lagrange problem in the calculus of variations where they are lifted into abnormal extremals. Singular trajectories are candidates as minimizers for the time-optimal control

problem, and they are parameterized by the maximum principle via a pseudo-Hamiltonian function. Moreover, besides their importance in optimal control theory, these trajectories play an important role in the classification of systems for the action of the feedback group.

Keywords

Abnormal extremals · End-point mapping · Martinet flat case in sub-Riemannian geometry · Pseudo-hamiltonian

Introduction

The concept of singular trajectories in optimal control corresponds to *abnormal extrema* in optimization. Suppose that a point $x^* \in X \simeq \mathbb{R}^n$ is a point of extremum for a smooth function $\mathcal{L} : \mathbb{R}^n \rightarrow \mathbb{R}$ under the equality constraints $F(x) = 0$ where $F : X \rightarrow Y$ is a smooth mapping into $Y \simeq \mathbb{R}^p$, $p < n$. The *Lagrange multiplier rule* (Agrachev et al. 1997) asserts the existence of nonzero pairs (λ_0, λ^*) of Lagrange multipliers such that $\lambda_0 \mathcal{L}'(x^*) + \lambda^* F'(x^*) = 0$. The *normality condition* is given by $\lambda_0 \neq 0$, and the abnormal case corresponds to the situation when the rank of $F'(x^*)$ is strictly less than p .

Abnormal extremals have played an important role in the standard calculus of variations (Bliss 1946). Indeed, consider a classical Lagrange problem:

$$\frac{dx}{dt}(t) = F(x(t), u(t)), \quad \min_{u(\cdot)} \int_0^T L(x(t), u(t)) dt \\ x(0) = x_0, x(T) = x_1,$$

where $x(t) \in X \simeq \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, F and L are smooth. Using an infinite dimensional framework, the Lagrange multiplier rule still holds and an abnormal extremum corresponds to a singularity of the set of constraints.

Definition

Consider a system of $\mathbb{R}^n$: $\frac{dx}{dt}(t) = F(x(t), u(t))$ where F is a smooth mapping from $\mathbb{R}^n \times \mathbb{R}^m$

into $\mathbb{R}^n$. Fix $x_0 \in \mathbb{R}^n$ and $T > 0$. The *end-point mapping* is the mapping $E^{x_0, T} : u(\cdot) \in \mathcal{U} \rightarrow x(T, x_0, u)$ where $\mathcal{U} \subset L^\infty[0, T]$ is the set of admissible controls such that the corresponding trajectory $x(\cdot, x_0, u)$ is defined on $[0, T]$. A control $u(\cdot)$ and its corresponding trajectory are called *singular* on $[0, T]$ if $u(\cdot) \in \mathcal{U}$ is such that the Fréchet derivative $E'^{x_0, T}$ of the end-point mapping is not of full rank n at $u(\cdot)$.

Fréchet Derivative and Linearized System

Given a reference trajectory $x(\cdot)$, $t \in [0, T]$, associated to $u(\cdot)$ with $x(0) = x_0$, and solution of $\frac{dx}{dt}(t) = F(x(t), u(t))$, the system

$$\delta \dot{x}(t) = A(t)\delta x(t) + B(t)\delta u(t)$$

with

$$A(t) = \frac{\partial F}{\partial x}(x(t), u(t)), \quad B(t) = \frac{\partial F}{\partial u}(x(t), u(t))$$

is called the *linearized system* along the control-trajectory pair $(u(\cdot), x(\cdot))$.

Let $M(t)$ be the fundamental matrix, $t \in [0, T]$ solution of

$$\dot{M}(t) = A(t)M(t), \quad M(0) = I_n.$$

Integrating the linearized system with $\delta x(0) = 0$, one gets the following proposition.

Proposition 1 *The Fréchet derivative of $E^{x_0, T}$ at $u(\cdot)$ is given by*

$$E_u'^{x_0, T}(v) = M(T) \int_0^T M^{-1}(t)B(t)v(t)dt.$$

Computation of the Singular Trajectories and Pontryagin Maximum Principle

According to the previous computations, a control $u(\cdot)$ with corresponding trajectory $x(\cdot)$ is singular on $[0, T]$ if the Fréchet derivative $E'^{x_0, T}$

is not of full rank at $u(\cdot)$. This is equivalent to the condition that the linearized system is *not controllable* (Lee and Markus 1967).

Such a condition is difficult to verify directly since the linearized system is time-dependent and the computation is associated to the Maximum Principle (Pontryagin et al. 1962).

Let p^* be a nonzero vector such that p^* is orthogonal to $\text{Im}(E^{x_0, T})$ and let $p(t) = p^* M(T) M^{-1}(t)$; then $p(\cdot)$ is solution of the *adjoint system*

$$\dot{p}(t) = -p(t) \frac{\partial F}{\partial u}(x(t), u(t))$$

and satisfies almost everywhere the equality

$$p(t) \frac{\partial F}{\partial u}(x(t), u(t)) = 0.$$

Introduce the *pseudo-Hamiltonian* $H(x, p, u) = \langle p, F(x, u) \rangle$, where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product, one gets the following characterization.

Proposition 2 *If (x, u) is a singular control-trajectory pair on $[0, T]$, then there exists a nonzero adjoint vector $p(\cdot)$ defined on $[0, T]$ such that (x, p, u) is solution a.e. of the following equations:*

$$\begin{aligned} \frac{dx}{dt} &= \frac{\partial H}{\partial p}(x, p, u), \quad \frac{dp}{dt} = -\frac{\partial H}{\partial x}(x, p, u) \\ \frac{\partial H}{\partial u}(x, p, u) &= 0. \end{aligned}$$

Application to the Lagrange Problem

Consider the problem

$$\frac{dx}{dt}(t) = F(x(t), u(t)), \min \int_0^T L(x(t), u(t)) dt$$

with $x(0) = x_0, x(T) = x_1$.

Introduce the *cost-extended pseudo-Hamiltonian*: $\tilde{H}(x, p, u) = \langle p, F(x, u) \rangle + p_0 L(x, u)$; it follows that the maximum principle is equivalent to the Lagrange multiplier rule presented in the introduction:

$$\begin{aligned} \frac{d\tilde{x}}{dt} &= \frac{\partial \tilde{H}}{\partial \tilde{p}}(\tilde{x}, \tilde{p}, u), \quad \frac{d\tilde{p}}{dt} = -\frac{\partial \tilde{H}}{\partial \tilde{x}}(\tilde{x}, \tilde{p}, u) \\ \frac{\partial \tilde{H}}{\partial u}(\tilde{x}, \tilde{p}, u) &= 0 \end{aligned}$$

where $\tilde{x} = (x, x^0)$ is the extended state variable solution of $\frac{dx}{dt} = F(x, u)$, $\frac{dx^0}{dt} = L(x, u)$ and $\tilde{p} = (p, p_0)$ is the extended adjoint vector. One has the condition $\langle \tilde{p}, \tilde{E}_u^{x_0, T}(v) \rangle = 0$ where $\tilde{E}_u^{x_0, T}$ is the cost-extended end-point mapping.

The Role of Singular Extremals in Optimal Control

While the traditional treatment in optimization of singular extremals is to consider them as a pathology, in modern optimal control, they play an important role which is illustrated by two examples from *geometric optimal control*.

Singular Trajectories in Quantum Control

Up to a normalization (Lapert et al. 2010), the time minimization *saturation problem* is to steer in minimum time the magnetization vector $M = (x, y, z)$ from the north pole of the Bloch Ball $N = (0, 0, 1)$ to its center $O = (0, 0, 0)$. The evolution of the system is described by the *Bloch equation* in nuclear magnetic resonance (Levitt 2008)

$$\frac{dx}{dt} = -\Gamma x + u_2 z$$

$$\frac{dy}{dt} = -\Gamma y - u_1 z$$

$$\frac{dz}{dt} = \gamma(1 - z) + u_1 y - u_2 x$$

where (Γ, γ) are proportional to the inverse of the relaxation times and $u = (u_1, u_2)$ is the control radio frequency-magnetic field bounded according to $|u| \leq M$. Due to the z -symmetry of revolution, one can restrict the problem to the 2D single-input case

$$\frac{dy}{dt} = -\Gamma y - uz, \quad \frac{dz}{dt} = \gamma(1 - z) + uy$$

that can be written as $\frac{dq}{dt} = F(q) + uG(q)$.

According to the maximum principle, the time-optimal solutions are the concatenations of *regular extremals* for which $u(t) = M \text{sign}\langle p(t), G(q(t)) \rangle$ and singular arcs where $\langle p(t), G(q(t)) \rangle = 0, \forall t$, and $p(t)$ is solution of the adjoint system. Differentiating with respect of time and using the *Lie bracket* notation $[X, Y](q) = \frac{\partial X}{\partial q}(q)Y(q) - \frac{\partial Y}{\partial q}(q)X(q)$, we get

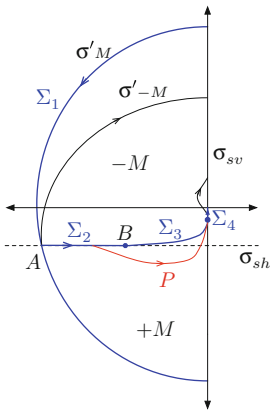
$$\langle p, [G, F](q) \rangle = 0,$$

$$\langle p, [[G, F], G](q) \rangle + u \langle p, [[G, F], F](q) \rangle = 0.$$

This leads to two singular arcs:

- The vertical line $y = 0$, corresponding to the z -axis of revolution
- The horizontal line $z = \frac{\gamma}{2(\gamma - \Gamma)}$

The interesting physical case is when $2\Gamma > 3\gamma$ where the vertical singular line is such that $-1 < \frac{\gamma}{2(\gamma - \Gamma)} < 0$. In this case, the time minimum solution is represented on Fig. 1. On Fig. 2 we draw the experimental solution in the deoxygenated blood case, compared with the standard inversion recovery sequence.



Singular Trajectories in Optimal Control, Fig. 1 The computed optimal solution is the following concatenation: bang arc σ'_M with the horizontal singular arc σ_{sh} followed by a bang arc P and finally the singular vertical arc σ_{sv}

Abnormal Extremals in SR Geometry

Sub-Riemannian geometry was introduced by R.W. Brockett as a generalization of Riemannian geometry (Brockett 1982; Montgomery 2002) with many applications in control (for instance, in motion planning (Bellaïche et al. 1998; Gauthier and Zakalyukin 2006) and quantum control). Its formulation in the framework of control theory is

$$\dot{q}(t) = \sum_{i=1}^m u_i(t) F_i(q(t)), \quad \min_{u(\cdot)} \int_0^T \left(\sum_{i=1}^m u_i^2(t) \right) dt$$

where $q \in U$ open set in $\mathbb{R}^n$, $m < n$ and $F_1, \dots, F_m$ are smooth vector fields which forms an orthonormal basis of the distribution they generate.

According to the maximum principle, normal extremals are solutions of the Hamiltonian vector field $\mathbf{H}_n$, $H_n = \frac{1}{2}(\sum_{i=1}^m H_i(q, p)^2)$, $H_i = \langle p, F_i(q) \rangle$ for $i = 1, \dots, m$. Again abnormal extremals can be computed by differentiating the constraint $H_i = 0$ along the extremals. Their first occurrence takes place in the so-called Martinet flat case: $n = 3, m = 2$, F_1, F_2 are given by

$$F_1 = \frac{\partial}{\partial x} + \frac{y^2}{2} \frac{\partial}{\partial z}, \quad F_2 = \frac{\partial}{\partial y}$$

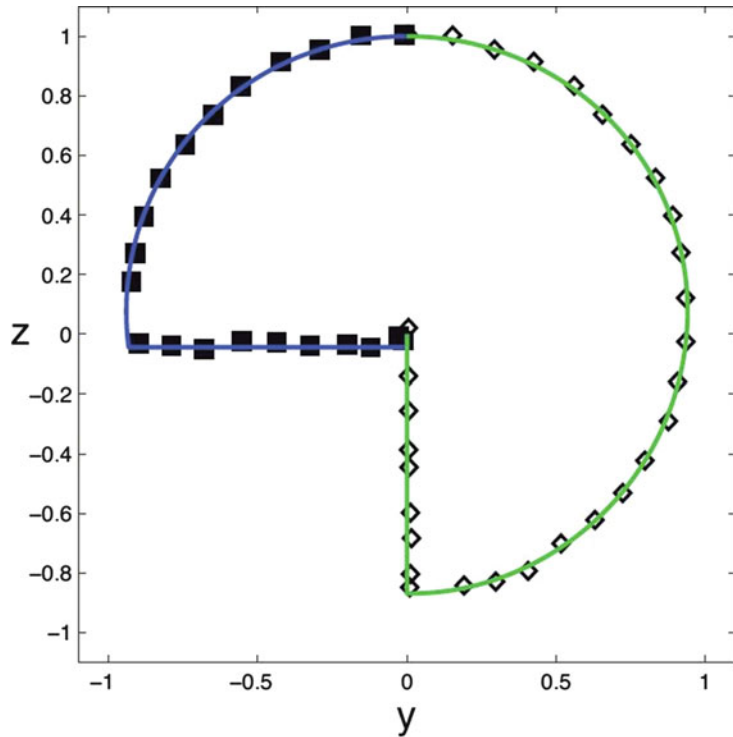
where $q = (x, y, z) \in U$ neighborhood of the origin, and the metric is given by $ds^2 = dx^2 + dy^2$. The singular trajectories are contained in the Martinet plane $M : y = 0$ and are the lines $z = z_0$. An easy computation shows that they are optimal for the problem. We represent below the role of the singular trajectories when computing the sphere of small radius, from the origin, intersected with the Martinet plane (Fig. 3).

Summary and Future Directions

Singular trajectories play an important role in many optimal control problem such as in

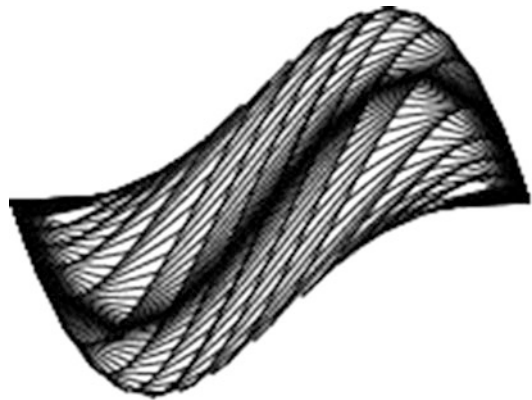
Singular Trajectories in Optimal Control,

Fig. 2 Experimental result. Usual inversion sequence in *green*, optimal computed sequence in *blue*



quantum control and cancer therapy (Schättler and Ledzewicz 2012). They have to be carefully analyzed in any applications; in particular in Boscain and Piccoli (2006) the authors provide for single-input systems in two dimensions a classification of optimal synthesis with singular arcs.

Additionally, from a theoretical point of view, singular trajectories can be used to compute feedback invariants for nonlinear systems (Bonnard and Chyba 2003). In relation, a purely mathematical problem is the classification of distributions describing the nonholonomic constraints in sub-Riemannian geometry (Montgomery 2002).



Singular Trajectories in Optimal Control, Fig. 3 Projection of the SR sphere on the xz -plane. The singular line is $x = t$ and the picture shows the pinching of the SR sphere in the singular direction

Cross-References

- Differential Geometric Methods in Nonlinear Control
- Feedback Stabilization of Nonlinear Systems
- Optimal Control and Pontryagin's Maximum Principle
- Robustness Issues in Quantum Control
- Sub-Riemannian Optimization

References

- Agrachev A, Sarychev AV (1998) On abnormal extremals for lagrange variational problems. *J Math Syst Estim Control* 8(1):87–118
- Agrachev A, Bonnard B, Chyba M, Kupka I (1997) Sub-Riemannian sphere in Martinet flat case. *ESAIM Control Optim Calc Var* 2:377–448
- Bellaïche A, Jean F, Risler JJ (1998) Geometry of nonholonomic systems. In: Laumond JP (ed) *Robot motion planning and control. Lecture notes in control and information sciences*, vol 229. Springer, London, pp 55–91
- Bliss G (1946) *Lectures on the calculus of variations*. University of Chicago Press, Chicago
- Bloch A (2003) *Nonholonomic mechanics and control. Interdisciplinary applied mathematics*, vol 24. Springer, New York
- Bonnard B, Chyba M (2003) Singular trajectories and their role in control theory. *Mathématiques & applications*, vol 40. Springer, Berlin
- Bonnard B, Cots O, Glaser S, Lapert M, Sugny D, Zhang Y (2012) Geometric optimal control of the contrast imaging problem in nuclear magnetic resonance. *IEEE Trans Autom Control* 57(8):1957–1969
- Boscain U, Piccoli B (2004) Optimal syntheses for control systems on 2-D manifolds. *Mathématiques & applications*, vol 43. Springer, Berlin
- Brockett RW, (1982) Control theory and singular Riemannian geometry. *New directions in applied mathematics*. Springer, New York/Berlin, pp 11–27
- Gauthier JP, Zakalyukin V (2006) On the motion planning problem, complexity, entropy, and nonholonomic interpolation. *J Dyn Control Syst* 12(3):371–404
- Lapert M, Zhang Y, Braun M, Glaser SJ, Sugny D (2010) Singular extremals for the time-optimal control of dissipative spin 1/2 particles. *Phys Rev Lett* 104: 083001
- Lapert M, Zhang Y, Janich M, Glaser SJ, Sugny D (2012) Exploring the physical limits of saturation contrast in magnetic resonance imaging. *Nat Sci Rep* 2: 589
- Lee EB, Markus L (1967) *Foundations of optimal control theory*. Wiley, New York/London/ Sydney
- Levitt MH (2008) *Spin dynamics: basics of nuclear magnetic resonance*, 2nd edn. Wiley, Chichester/Hoboken
- Montgomery R (2002) *A tour of subriemannian geometries, their geodesics and applications. Mathematical surveys and monographs*, vol 91. American Mathematical Society, Providence
- Schättler H, Ledzewicz U (2012) *Geometric optimal control: theory, methods and examples. Interdisciplinary applied mathematics*, vol 38. Springer, New York
- Pontryagin LS, Boltyanskii VG, Gamkrelidze RV, Mishchenko EF (1962) *The mathematical theory of optimal processes* (Translated from the Russian by Tririgoff KN; edited by Neustadt LW). Wiley Interscience, New York/London

Sliding Mode Control: Finite-Time Observation and Regulation

Arie Levant

Department of Applied Mathematics, Tel-Aviv University, Ramat-Aviv, Israel

Abstract

Sliding Modes (SMs) are used to control uncertain systems by properly choosing and exactly keeping constraints involving system outputs and their derivatives. The constraint relative degree turns out to be the main approach parameter. Modern SM control (SMC) applies real-time precise differentiators and establishes the constraint exactly and in finite time. SMC systems are robust to noisy discrete sampling, unaccounted-for additional model dynamics, actuators, and sensors.

Keywords

Variable structure systems · Relative degree · Discontinuous control · Robustness · Uncertainty · Filtering · Differentiation · Sampling · Noise · Homogeneity

Notation A binary operation $\diamond$ of two sets is defined as $A \diamond B = \{a \diamond b \mid a \in A, b \in B\}$, $a \diamond B = \{a\} \diamond B$. A function of a set is the set of function values on this set. $\mathbb{R}_+ = [0, \infty)$; $\lfloor a \rfloor^b = |a|^b \text{sign } a$, $\lfloor a \rfloor^0 = \text{sign } a$.

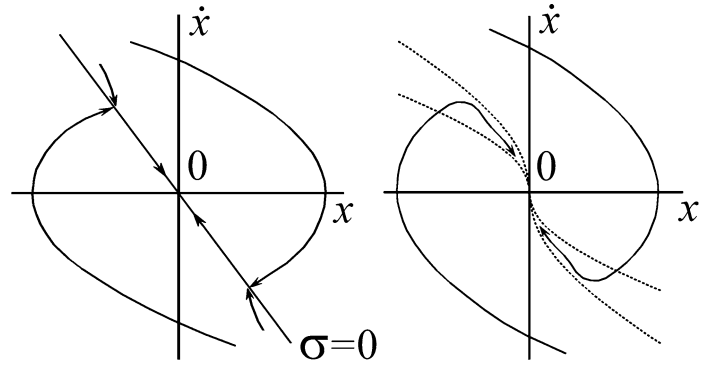
Introduction

Sliding mode control (SMC) systems, also called variable-structure systems, counteract uncertainties by keeping constraints in the permanent-switching mode called SM (Utkin 1992; Utkin et al. 2009; Slotine and Li 1991). For example, consider stabilizing the system

$$\ddot{x} = A(t, x, \dot{x}) + B(t, x, \dot{x})u, \quad x, u \in \mathbb{R}, \\ |A(t, x, \dot{x})| \leq 1, \quad B(t, x, \dot{x}) \in [1, 2].$$

**Sliding Mode Control:
Finite-Time Observation
and Regulation, Fig. 1**

Trajectories of (a) classic SMC, (b) quasi-continuous second-order SMC



a. Classic SMC

b. 2nd-order SMC

Any line in the phase plane $x, \dot{x}$ also has the meaning of a differential equation. Thus, keeping the trajectory on the line $\sigma = \dot{x} + x = 0$ in the SM $\sigma = 0$ asymptotically stabilizes the system. The corresponding control $u = -(2 + |\dot{x}|) \text{sign } \sigma$ is the classical SMC (Fig. 1a, Utkin 1992). Another option is to apply the control $u = -2 \frac{|\dot{x}|^2 + x}{\dot{x}^2 + |x|}$. It stabilizes x in finite time (FT) and is discontinuous only at $x = \dot{x} = 0$ keeping $x \equiv 0$ in the *second-order SM* (Fig. 1b, Ding et al. 2016).

Thus, the uncertainty has been completely removed by discontinuous control. Both controls utilize estimations of $\dot{x}$ for stabilizing x .

Basic Notions

Filippov definition Consider a differential equation (DE) $\dot{x} = v(t, x)$, $x \in \mathbb{R}^{n_x}$, where v is a locally essentially bounded Lebesgue-measurable function. Its Filippov solution is any locally absolutely continuous function $x(t)$ which almost everywhere satisfies $\dot{x} \in K_F[v](t, x)$, $K_F[v](t, x) = \bigcap_{\delta > 0} \bigcap_{\mu_L N = 0} \overline{\text{co}} v(t, O_\delta(x) \setminus N)$. Here μ_L is the Lebesgue measure, $O_\delta(x)$ is the δ -vicinity of x , and $\overline{\text{co}}(\cdot)$ denotes the convex closure operation.

Usually v is almost everywhere continuous. Then $K_F[v](t, x)$ is the convex closure of all possible limits of $v(t, y)$, obtained as continuity points (t, y) approach (t, x) . Filippov solutions can be considered as limit solutions produced

by various switching regularizations (Filippov 2013). All DEs are further understood in the Filippov sense.

Relative degree The relative-degree notion (Isidori 1995) is extended to the non-autonomous case by adding the fictitious equation $\dot{t} = 1$. For the simplicity in the following, we restrict ourselves to the single-input single-output (SISO) case.

Consider a smooth SISO system

$$\dot{x} = a(t, x) + b(t, x)u, \quad \sigma = \sigma(t, x), \quad (1)$$

where $x \in \mathbb{R}^{n_x}$, a, b , and σ are smooth functions, $u, \sigma(t, x) \in \mathbb{R}$. If r is the relative degree at the point (t_0, x_0) , then

$$\sigma^{(r)} = h(t, x) + g(t, x)u \quad (2)$$

holds for some smooth h, g , where the function $g(t, x)$ does not vanish in a vicinity of (t_0, x_0) . Relative-degree identification is often reduced to finding the shortest chain-rule differentiation way from σ to u and does not require the exact model knowledge.

In practice almost always $r = 2, 3, 4$, for engineers need a simple model. Significant parts of a real system are voluntarily moved to actuators and sensors or are simply ignored as insignificant functional and singular perturbations. Note that SMC systems are robust to such model perturbations (Levant and Livne 2016; Levant 2010).

Sliding mode Any Filippov solution lying on the discontinuity surface/set of a differential equation is said to be in SM. If a constraint $\sigma = 0$ is kept, the notation SM $\sigma = 0$ is used, and σ is called *the sliding variable* (Utkin 1992).

SM order Consider a SM $\sigma = 0$ in a closed-loop system. Let σ be a continuous scalar function. *The sliding order* k is the lowest k , such that the k th-order total time derivative $\sigma^{(k)}$ is not a continuous function of the state and time (Levant 2003; Shtessel et al. 2014). The corresponding motion $\sigma \equiv 0$ is called the k th-order SM or k -sliding mode (k -SM). In the case of a vector sliding variable σ also the sliding order is a vector.

Connection to the relative degree Consider system (1) with a scalar sliding variable σ of the relative degree r . Then, $\sigma, \dot{\sigma}, \dots, \sigma^{(r-1)}$ are continuous functions of t, x , i.e., the sliding order is never less than r .

In the control discontinuity case, the relative and the sliding order, the SM, and the zero dynamics coincide. In that case the function $u_{eq} = -h/g$, found from (2) and $\sigma^{(r)} = 0$, is traditionally called the *equivalent control* (Utkin 1992). The classic SMs (Slotine and Li 1991; Utkin 1992) (Fig. 1a) are based on the 1-SM $\sigma = 0$.

SMC completely removes *matched disturbances*. Indeed, let (1) have the form $\dot{x} = a(t, x) + b(t, x)(u + \xi(t, x))$. Then SM (zero) dynamics do not depend on ξ .

Chattering attenuation SMC can generate undesired system vibrations, called chattering (Boiko and Fridman 2005), which often are considered the main drawback of SMC. Replacing the relay function $\text{sign } \sigma$ with a “sigmoid” function, like $\sigma/(|\sigma| + \epsilon)$, $\epsilon > 0$, (Slotine and Li 1991), yields a system sensitive to uncertainties for finite ϵ and causes strong chattering due to small sampling noises for $\epsilon \ll 1$ (Levant 2010).

Another chattering-reduction method consists in inserting a number of integrators in the feedback (Bartolini et al. 1998; Shtessel et al. 2014). Suppose the relative degree is r . Then the virtual discontinuous control $u^{(l)}$ is applied to establish the $(r + l)$ -SM $\sigma = 0$. Correspondingly, $u, \dot{u}, \dots, u^{(l-1)}$ are formally included in the system state (Dorel and Levant 2008). Note that the chattering reduction is not due to the continuity of the resulting actual control $u(t)$, but due to simultaneously keeping $\sigma, \dot{\sigma}, \dots, \sigma^{(r+l-1)}$ at zero, while only $\sigma, \dot{\sigma}, \dots, \sigma^{(r-1)}$ are the physical plant coordinates (Levant 2010).

Finite-Time Output Regulation

SMC problem Consider system (1), (2) of the relative degree r , and assume that

$$|h(t, x)| \leq C, \quad 0 < K_m \leq g(t, x) \leq K_M. \quad (3)$$

Such assumption holds at least for any compact operational region. Each solution of (1) is assumed infinitely extendable in time, provided σ , its derivatives $\dot{\sigma}, \dots, \sigma^{(r-1)}$, and u remain bounded along the solution.

Due to the uncertainty of g, h , any feedback control $u = u(\sigma)$, $\sigma = (\sigma, \dot{\sigma}, \dots, \sigma^{(r-1)})$ is to be discontinuous. Such a control is called quasi-continuous (QC) if it is continuous at each $\sigma \neq 0$.

Consider an output-regulation/tracking problem. Let σ be the tracking error, and consider the problem of establishing the r -SM $\sigma = 0$ by a feedback $u(\sigma)$ in FT.

Homogeneous SMC

Uncertain dynamics (2) satisfy the concrete differential inclusion $\sigma^{(r)} \in [-C, C] + [K_m, K_M]u$. Most r -SM controllers are built as its FT stabilizers. Probably the simplest QC r -SMC has the form (Cruz-Zavala and Moreno 2017; Ding et al. 2016)

$$u = -\alpha \Psi_r(\sigma) = -\alpha \frac{|\sigma^{(r-1)}|^{\frac{\omega}{r}} + \beta_{r-2} |\sigma^{(r-2)}|^{\frac{\omega}{2}} + \dots + \beta_0 |\sigma|^{\frac{\omega}{r}}}{|\sigma^{(r-1)}|^{\frac{\omega}{r}} + \beta_{r-2} |\sigma^{(r-2)}|^{\frac{\omega}{2}} + \dots + \beta_0 |\sigma|^{\frac{\omega}{r}}}, \quad \omega > 0. \quad (4)$$

Theorem by Ding et al. (2016) states that for any $\omega > 0$, there exist such $\beta_0, \dots, \beta_{r-2} > 0$ that controller (4) stabilizes σ in FT for any sufficiently large $\alpha > 0$ only depending on K_m, K_M, C .

Functions $\Psi_r(\sigma)$ are invariant with respect to the transformation $\sigma^{(i)} \mapsto \kappa^{r-i} \sigma^{(i)}$,

$$\begin{aligned} r=1: u &= -\alpha \operatorname{sign} \sigma; & r=2: u &= -\alpha \frac{|\dot{\sigma}|^2 + \sigma}{\dot{\sigma}^2 + |\sigma|}; \\ r=3: u &= -\alpha \frac{\ddot{\sigma}^3 + 2|\dot{\sigma}|^{\frac{3}{2}} + \sigma}{|\ddot{\sigma}|^3 + 2|\dot{\sigma}|^{\frac{3}{2}} + |\sigma|}; & r=4: u &= -\alpha \frac{|\ddot{\sigma}|^4 + 2|\dot{\sigma}|^2 + 2|\dot{\sigma}|^{\frac{4}{3}} + \sigma}{\ddot{\sigma}^4 + 2\ddot{\sigma}^2 + 2\dot{\sigma}^{\frac{4}{3}} + |\sigma|}; \\ r=5: u &= -\alpha \frac{|\sigma^{(4)}|^5 + 6|\ddot{\sigma}|^{\frac{5}{2}} + 5|\ddot{\sigma}|^{\frac{5}{3}} + 3|\dot{\sigma}|^{\frac{5}{4}} + \sigma}{|\sigma^{(4)}|^5 + 6|\ddot{\sigma}|^{\frac{5}{2}} + 5|\ddot{\sigma}|^{\frac{5}{3}} + 3|\dot{\sigma}|^{\frac{5}{4}} + |\sigma|}. \end{aligned} \quad (5)$$

Differentiation and Filtering

Let $\operatorname{Lip}_n(L)$ be the set of all functions $\mathbb{R}_+ \rightarrow \mathbb{R}$, whose n th derivative has the Lipschitz constant $L > 0$. Assume that the sampled input signal $f(t)$, $f(t) = f_0(t) + \eta(t)$, consist of a Lebesgue-measurable noise $\eta(t)$ and an unknown basic signal $f_0(t)$, $f_0 \in \operatorname{Lip}_n(L)$. The noise η is bounded, $|\eta| \leq \varepsilon_0$. The number $\varepsilon_0 \geq 0$ is unknown.

Differentiation problem (Levant 2003) The problem is to estimate the derivatives $f_0^{(i)}(t)$, $i = 0, 1, \dots, n$, in real time by differentiator outputs $z_i(t)$. That estimation is to be exact in the absence of noises after some FT transient. The steady-state accuracies $\sup |z_i - f_0^{(i)}|$ are to continuously depend on ε_0 .

Asymptotically optimal differentiation Any differentiator exact on the input signals $f_0, f \in \operatorname{Lip}_n(L)$ has the worst-case steady-state accuracy $\sup |z_i - f_0^{(i)}| = 2^{\frac{i}{n+1}} K_{n,i} L^{\frac{i}{n+1}} \varepsilon_0^{\frac{n+1-i}{n+1}}$ for some f_0, f and $|f(t) - f_0(t)| \leq \varepsilon_0$ (Levant et al. 2017). Here $K_{n,i} \in [1, \pi/2]$ are the Kolmogorov constants (Kolmogoroff 1962), e.g., $K_{1,1} = \sqrt{2}$.

A differentiator is called *asymptotically optimal* (Levant et al. 2017; Levant and Yu 2018) if for some $\mu_i > 0$ the steady-state accuracy $\sup |z_i - f_0^{(i)}| \leq \mu_i L^{\frac{i}{n+1}} \varepsilon_0^{\frac{n+1-i}{n+1}}$ holds for any signal $f_0 \in \operatorname{Lip}_n(L)$ and any noise η , $|\eta(t)| \leq \varepsilon_0$, $i = 0, 1, \dots, n$.

$i = 0, 1, \dots, r-1$, $\kappa > 0$. Such controllers are called r -SM homogeneous (Levant and Livne 2016; Shtessel et al. 2014). Obviously, $|\Psi_r(\sigma)| \leq 1$, i.e., $|u| \leq \alpha$.

The following are valid controllers for $r = 1, 2, 3, 4, 5$, $\omega = r$:

Introduce the number $n_f \geq 0$ which is further called *the differentiator filtering order*. The **filtering differentiator** (Levant and Yu 2018; Levant and Livne 2019) has the form

$$\begin{aligned} \dot{w}_1 &= -\tilde{\lambda}_{n+n_f} L^{\frac{1}{n+n_f+1}} [w_1]^{\frac{n+n_f}{n+n_f+1}} + w_2, \\ &\dots \\ \dot{w}_{n_f-1} &= -\tilde{\lambda}_{n+2} L^{\frac{n_f-1}{n+n_f+1}} [w_1]^{\frac{n+2}{n+n_f+1}} + w_{n_f}, \\ \dot{w}_{n_f} &= -\tilde{\lambda}_{n+1} L^{\frac{n_f}{n+n_f+1}} [w_1]^{\frac{n+1}{n+n_f+1}} + z_0 - f(t), \\ \dot{z}_0 &= -\tilde{\lambda}_n L^{\frac{n_f+1}{n+n_f+1}} [w_1]^{\frac{n}{n+n_f+1}} + z_1, \\ &\dots \\ \dot{z}_{n-1} &= -\tilde{\lambda}_1 L^{\frac{n+n_f}{n+n_f+1}} [w_1]^{\frac{1}{n+n_f+1}} + z_n, \\ \dot{z}_n &= -\tilde{\lambda}_0 L \operatorname{sign}(w_1). \end{aligned} \quad (6)$$

In the case $n_f = 0$ the first n_f equations (6) disappear, and $w_1 = z_0 - f(t)$ is formally substituted for w_1 yielding the well-known differentiator by Levant (2003). In the case $n = 0$ only the equation for z_0 remains in the lower part (7).

Parameters $\tilde{\lambda}_i$ are most easily found using the parameters $\lambda_0, \dots, \lambda_p$, $p = n + n_f$, of the differentiator recursive form (Levant et al. 2017; Levant 2003): $\tilde{\lambda}_0 = \lambda_0$, $\tilde{\lambda}_p = \lambda_p$, and $\tilde{\lambda}_j = \lambda_j \tilde{\lambda}_{j+1}^{j/(j+1)}$, $j = p-1, p-2, \dots, 1$. An infinite sequence of parameters $\lambda = \{\lambda_0, \lambda_1, \dots\}$ can be

Sliding Mode Control: Finite-Time Observation and Regulation, Table 1 Parameters $\tilde{\lambda}_0, \tilde{\lambda}_1, \dots, \tilde{\lambda}_{n+n_f}$ of differentiator (6), (7) for $n + n_f = 0, 1, \dots, 7$

0	1.1							
1	1.1	1.5						
2	1.1	2.12	2					
3	1.1	3.06	4.16	3				
4	1.1	4.57	9.30	10.03	5			
5	1.1	6.75	20.26	32.24	23.72	7		
6	1.1	9.91	43.65	101.96	110.08	47.69	10	
7	1.1	14.13	88.78	295.74	455.40	281.37	84.14	12

built (Levant 2003), providing coefficients $\tilde{\lambda}_i$ of (6), (7) for all $n, n_f \geq 0$. In particular, $\lambda = \{1.1, 1.5, 2, 3, 5, 7, 10, 12, \dots\}$ suffice for $n + n_f \leq 7$ (Levant et al. 2017). The corresponding parameters $\tilde{\lambda}_i$ are listed in Table 1.

For brevity denote (6), (7) by

$$\dot{w} = \Omega_{n,n_f}(w, z_0 - f, L), \quad \dot{z} = D_{n,n_f}(w_1, z, L), \quad (8)$$

with the tracking difference $z_0(t) - f(t)$ singled out as a separate argument.

Extend the above conditions on the input by letting the noise have the form $\eta(t) = \eta_0(t) + \eta_1(t) + \dots + \eta_{n_f}(t)$, where each η_k , $k = 0, \dots, n_f$, is a Lebesgue-measurable locally-integrable signal. For each k there exists a uniformly bounded solution $\xi_k(t)$ of the equation $\dot{\xi}^{(k)} = \eta_k$, $|\xi(t)| \leq \varepsilon_k$. Though $\eta_1, \dots, \eta_{n_f}$ are possibly unbounded, they can be described as being small in the average.

The differentiator in FT provides the accuracy

$$|z_i(t) - f_0^{(i)}(t)| \leq \mu_i L \rho^{n+1-i}, \quad i = 0, 1, \dots, n, \\ \rho = \max \left[\left(\frac{\varepsilon_0}{L} \right)^{1/(n+1)}, \dots, \left(\frac{\varepsilon_{n_f}}{L} \right)^{1/(n+n_f+1)} \right] \quad (9)$$

for some $\mu_i > 0$ only depending on the parameters $\tilde{\lambda}_0, \dots, \tilde{\lambda}_{n+n_f}$ (Levant and Livne 2019).

Taking $\eta_1, \dots, \eta_{n_f} = 0$, obtain that differentiator (6), (7) is asymptotically optimal. Moreover, it is also applicable in the case when the noise components η_k are only small in average on any *finite* time interval not exceeding some $T_k > 0$ in its length (Levant and Livne 2019). The error dynamics of the differentiator is homogeneous (Levant and Livne 2019).

Homogeneous Output Feedback SMC

The stated SMC problem of the FT exact stabilization of σ is solved by the output feedback SMC

$$\begin{aligned} \dot{w} &= \Omega_{r-1,n_f}(w, z_0 - \sigma, L), \\ \dot{z} &= D_{r-1,n_f}(w_1, z, L), \\ u &= -\alpha \Psi(z), \quad L \geq C + K_M \alpha \end{aligned} \quad (10)$$

for any filtering order $n_f \geq 0$. The proof is trivial, since $\sigma \in \text{Lip}_{r-1}(L)$.

Let the sliding variable be sampled with the same noise $\eta(t) = \eta_0(t) + \eta_1(t) + \dots + \eta_{n_f}(t)$ as above. Then for any sufficiently large $\alpha > 0$, control (10) in FT provides for the accuracy

$$|\sigma_0^{(i)}(t)| \leq \tilde{\mu}_i \rho^{r-i}, \quad i = 0, 1, \dots, r-1, \\ \rho = \max \left[\left(\frac{\varepsilon_0}{L} \right)^{1/r}, \dots, \left(\frac{\varepsilon_{n_f}}{L} \right)^{1/(r+n_f)} \right] \quad (11)$$

for some $\tilde{\mu}_i > 0$ only depending on the parameters $\tilde{\lambda}_0, \dots, \tilde{\lambda}_{n+n_f}, L, \alpha, C, K_m, K_M$.

Note that the bound L can be very rough (50–100 times larger than needed), and the knowledge of C, K_m, K_M is not really required, since the control parameter α is usually found by simulation.

Discretization

In reality the system evolves in the continuous time, whereas the sampling and the control injection are performed at discrete times. Correspondingly, the differentiator dynamics are replaced with some numeric integration of (8).

Let the input be sampled at the times $t_0, t_1, \dots, t_k \rightarrow \infty, \tau_k = t_{k+1} - t_k > 0, \tau_k \leq \tau$.

Discretization of the output-feedback control (10) is done by the one-step Euler method with u and z kept constant over each sampling interval. In that case the accuracy (11) is preserved for $\rho = \max[(\varepsilon/L)^{1/r}, \tau]$ with possibly changed coefficients $\tilde{\mu}_i$, provided $\eta_1(t) + \dots + \eta_{n_f}(t) \equiv 0$, i.e., only η_0 is present (Levant and Livne 2015). The general-case formula is more complicated and is omitted here (Levant and Livne 2019).

The stand-alone discretization of the differentiator (8) takes the form (Levant and Livne 2019)

$$\begin{aligned} w(t_{k+1}) &= w(t_k) + \mathcal{Q}_{n,n_f}(w(t_k), z_0(t_k) - f(t_k), L)\tau_k, \\ z(t_{k+1}) &= z(t_k) + D_{n,n_f}(w_1(t_k), z(t_k), L)\tau_k + H_n(z(t_k), \tau_k), \\ H_n(z(t_k), \tau_k) &= (H_{n,0}, \dots, H_{n,n})^T, \quad H_{n,n-1} = H_{n,n} = 0, \\ H_{n,i} &= \frac{1}{2!}z_{i+2}(t_k)\tau_k^2 + \dots + \frac{1}{(n-i)!}z_n(t_k)\tau_k^{n-i}, \quad i = 0, 1, \dots, n-2. \end{aligned} \quad (12)$$

The Taylor-like terms H_n are required only to improve the accuracy with respect to τ in the presence of very small noises (Livne and Levant 2014). Also here formula (9) remains true when only the noise η_0 is present and $\rho = \max[(\varepsilon/L)^{1/n+1}, \tau]$. The general formula is more complicated (Levant and Livne 2019).

Examples

Numeric differentiation Consider the sampled signal

$$\begin{aligned} f(t) &= f_0(t) + \eta(t), \\ f_0(t) &= 0.5 \sin t + 0.8 \cos(0.8t), \end{aligned} \quad (13)$$

where $\eta(t)$ is the noise. Let the sampling interval be constant, $\tau_k = \tau$. The filtering differentiator (12) of the order $n = 5$ and the filtering order 2 is applied with $L = 1$, $\tau = 10^{-4}$, and $z(0) = 0$, $w(0) = 0$. Obviously $|f_0^{(6)}| \leq L$. The coefficients are taken from Table 1 from row 7, $7 = 5 + 2$.

Performance of the differentiator for $\eta = 0$ is demonstrated in Fig. 2a. Denote $|\Sigma|_{5,2} = (|w_1|, |w_2|, |z_0 - f_0|, \dots, |z_5 - f_0^{(5)}|)$. The differentiation accuracy for $t \in [10, 20]$ is provided by the component-wise inequality $|\Sigma|_{5,2} \leq (3.0 \cdot 10^{-23}, 2.4 \cdot 10^{-19}, 1.3 \cdot 10^{-15}, 1.4 \cdot 10^{-12}, 1.2 \cdot 10^{-9}, 5.1 \cdot 10^{-7}, 1.1 \cdot 10^{-4}, 0.012)$. This accuracy is practically the best one which can be obtained in the presence of the digital round-up errors due to the standard double-precision number format (Levant et al. 2017).

Consider the noise $\eta(t) = 3 \cos(10000t) - 6 \sin(20000t) - 4 \cos(70000t) + \eta_G(t)$, $\eta_G \in N(0, 0.1^2)$, where $\eta_G(t)$ is a Gaussian signal with the standard deviation 0.1. The corresponding performance of the same differentiator appears in Fig. 2b. The accuracy is $|\Sigma|_{5,2} \leq (1.2 \cdot 10^{-5}, 1.8 \cdot 10^{-3}, 0.015, 0.14, 0.60, 1.4, 1.9, 1.1)$.

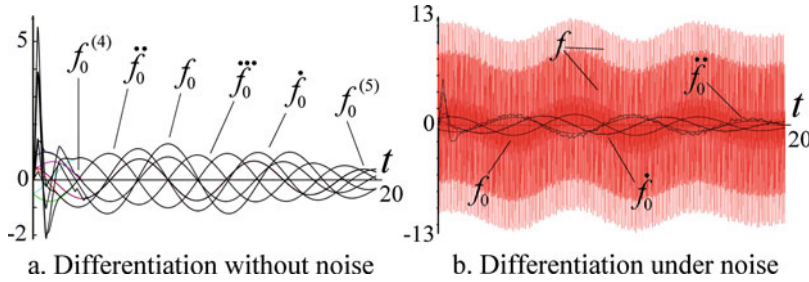
Car control Consider a simple “bicycle” kinematic car model (Rajamani 2005)

$$\dot{x} = V \cos(\varphi), \quad \dot{y} = V \sin(\varphi), \quad \dot{\varphi} = \frac{V}{\Delta} \tan \theta, \quad \dot{\theta} = u, \quad (14)$$

where x and y are the Cartesian coordinates of the middle point of the rear axle (Fig. 3a), $\Delta = 5m$ is the distance between the two axles, φ is the orientation angle, $V = 10m/s$ is the constant longitudinal velocity, θ is the steering angle (i.e., the actual input), and $u = \dot{\theta}$ is the control.

The goal is to move along the trajectory $(x(t), y(t)) = (x(t), g(t))$, whereas $g(t), y(t)$ are sampled in real time. Correspondingly, introduce the sliding variable $\sigma = y(t) - g(t)$ sampled with the time step τ and a noise $\eta(t)$. Let $g(t) = 10 \sin(0.05x(t)) + 5$.

Obviously, $\dot{x}, \dot{y}$ contain φ , $\dot{\varphi}$ contains θ , and $\dot{\theta} = u$. Thus, the relative degree is 3. Apply the standard 3-SM controller (10), (5) for $r = 3$, $n_f = 2$, $L = 50$, $\alpha = 0.5$. Parameters α, L are found by simulation. The second-order differentiator starts at $t = 0$; the control is applied from $t = 1$ to $t = 40$. The integration is

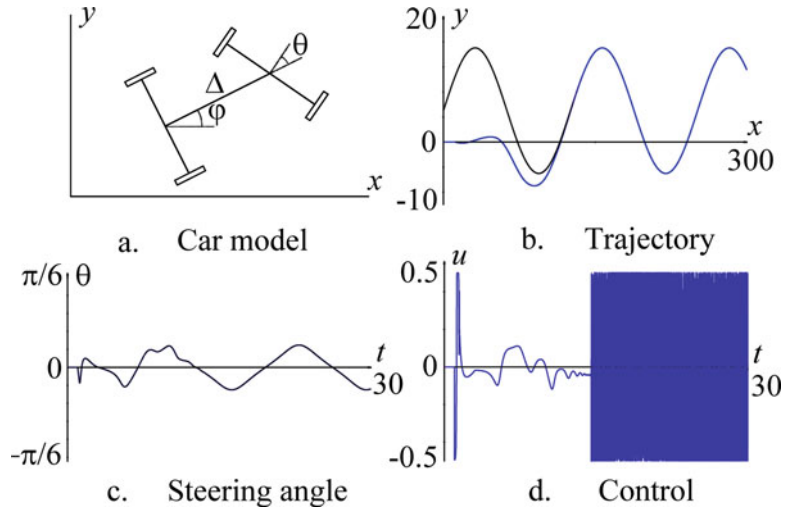


Sliding Mode Control: Finite-Time Observation and Regulation, Fig. 2 Performance of differentiator (12) with $n = 5$, $n_f = 2$, $L = 1$ for $\tau = 10^{-5}$ and input (13):

(a) there is no noise; (b) performance for noisy sampling (only estimations of f_0 , $\dot{f}_0$, $\ddot{f}_0$ are shown)

Sliding Mode Control: Finite-Time Observation and Regulation, Fig. 3

Performance of the 3-SM car control (10) for $r = 3$, $n_f = 2$, $L = 50$, $\alpha = 0.5$, and the integration/sampling step $\tau = 10^{-5}$: (a) car model; (b) the required and the resulting trajectories; (c) steering angle; (d) control



performed by the Euler method with the time step $10^{-5}s$ (MatLab ODE solvers are not applicable).

First consider the case of the exact measurements, $\eta = 0$, $\tau = 10^{-5}s$. The corresponding performance is shown in Fig. 3. The SM accuracy $|\sigma| \leq 1.1 \cdot 10^{-12}m$, $|\dot{\sigma}| \leq 1.3 \cdot 10^{-5}m/s$, $|\ddot{\sigma}| \leq 0.002m/s^2$ is maintained.

Now introduce the sampling noise $\eta(t) = 2\cos(10000t) + \eta_G(t)$, $\eta_G \in N(0, 0.5^2)$ (Fig. 4c). The corresponding performance is shown in Fig. 4. The SM accuracy $|\sigma| \leq 0.041m$, $|\dot{\sigma}| \leq 0.67m/s$, $|\ddot{\sigma}| \leq 5.2m/s^2$ is maintained for the sampling step $\tau = 10^{-5}$ (Fig. 4a). The accuracy deteriorates for the sampling step $\tau = 0.01s$ to $|\sigma| \leq 2.8m$, $|\dot{\sigma}| \leq 2.7m/s$, $|\ddot{\sigma}| \leq 6.8m/s^2$ (Fig. 4b,d). The performance is still quite acceptable.

Summary and Future Directions

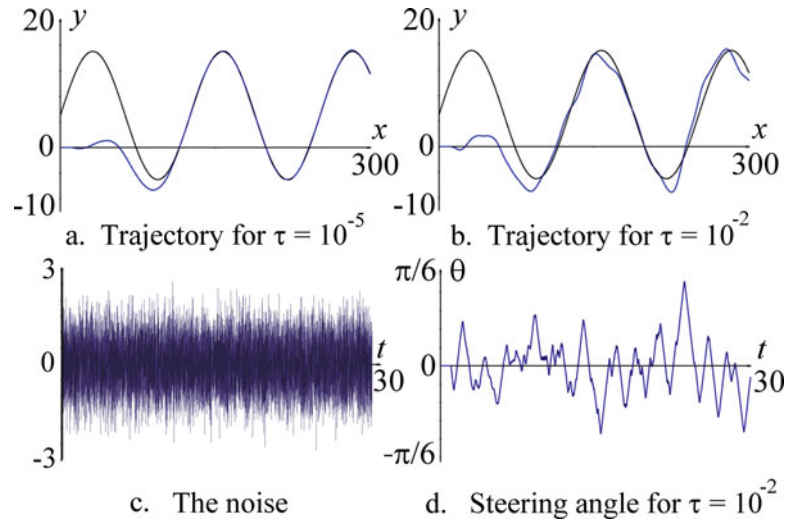
Multi-input multi-output regulation problem is solvable provided some rough approximation of the high-frequency gain matrix (Isidori 1995) is available (Defoort et al. 2009; Levant and Shustin 2018). Non-minimum phase SMC remains a long-standing challenge.

SMC contours are easily incorporated in control systems due to their accuracy and robustness, FT convergence, and the control smoothness option. In particular, SM observers and filters are capable of replacing expensive sensors in practical applications and linear filters in signal processing. Parameter L can be variable (Levant et al. 2017).

SMC systems are robust to noises, delays, singular and regular perturbations, and even to

Sliding Mode Control: Finite-Time Observation and Regulation, Fig. 4

Performance of the 3-SM car control (10) for $r = 3$, $n_f = 2$, $L = 50$, $\alpha = 0.5$, for the noisy sampling with the step τ : (a) trajectory for $\tau = 10^{-5}$ s; (b) trajectory for $\tau = 0.01$ s; (c) the noise; (d) steering angle for $\tau = 0.01$ s



relative-degree fluctuations (Levant 2010; Levant and Livne 2016). The practical relative degree approach (Shtessel et al. 2014) is expected to solve numerous problems with non-existing or large relative degrees or even without a reliable model.

Integral SMs (Utkin et al. 2009) remove the transient to SM. The Lyapunov-based homogeneous SM control and observation (Bernuau et al. 2014; Cruz-Zavala and Moreno 2016, Cruz-Zavala and Moreno 2017) is capable of fixed-time convergence (Angulo et al. 2013; Polyakov et al. 2015). Alternative SMC discretization strategies by Acary et al. (2012) reduce the chattering and improve the accuracy. Discrete SMs are considered by Furuta (1990), Bartoszewicz (1998) and Gao et al. (1995). SMC of infinite-dimensional systems is considered by Orlov (2008).

Cross-References

- [Differential Geometric Methods in Nonlinear Control](#)
- [Feedback Stabilization of Nonlinear Systems](#)
- [Nonlinear Zero Dynamics](#)
- [Regulation and Tracking of Nonlinear Systems](#)

Bibliography

- Acary V, Brogliato B, Orlov Y (2012) Chattering-free digital sliding-mode control with state observer and disturbance rejection. *IEEE Trans Autom Control* 57(5):1087–1101
- Angulo M, Moreno J, Fridman L (2013) Robust exact uniformly convergent arbitrary order differentiator. *Automatica* 49(8):2489–2495
- Bartolini G, Ferrara A, Usai E (1998) Chattering avoidance by second-order sliding mode control. *IEEE Trans Autom Control* 43(2):241–246
- Bartoszewicz A (1998) Discrete-time quasi-sliding-mode control strategies. *IEEE Trans Ind Electron* 45(4):633–637
- Bernuau E, Efimov D, Perruquetti W, Polyakov A (2014) On homogeneity and its application in sliding mode control. *J Frankl Inst* 351(4):1866–1901
- Boiko I, Fridman L (2005) Analysis of chattering in continuous sliding-mode controllers. *IEEE Trans Autom Control* 50(9):1442–1446
- Cruz-Zavala E, Moreno J (2016) Lyapunov functions for continuous and discontinuous differentiators. *IFAC-PapersOnLine* 49(18):660–665
- Cruz-Zavala E, Moreno J (2017) Homogeneous high order sliding mode design: a Lyapunov approach. *Automatica* 80:232–238
- Defoort M, Floquet T, Kokosy A, Perruquetti W (2009) A novel higher order sliding mode control scheme. *Syst Control Lett* 58(2):102–108
- Ding S, Levant A, Li S (2016) Simple homogeneous sliding-mode controller. *Automatica* 67(5):22–32
- Dorel L, Levant A (2008) On chattering-free sliding-mode control. In: 47th IEEE conference on decision and control, 2008, pp 2196–2201

- Filippov A (2013) Differential equations with discontinuous righthand sides. Springer Science & Business Media, Dordrecht
- Furuta K (1990) Sliding mode control of a discrete system. *Syst Control Lett* 14(2):145–152
- Gao W, Wang Y, Homaiifa A (1995) Discrete-time variable structure control systems. *IEEE Trans Ind Electron* 42(2):117–122
- Isidori A (1995) Nonlinear control systems I. Springer, New York
- Kolmogoroff AN (1962) On inequalities between upper bounds of consecutive derivatives of an arbitrary function defined on an infinite interval. *Amer Math Soc Transl Ser 1* 2:233–242
- Levant A (2003) Higher order sliding modes, differentiation and output-feedback control. *Int J Control* 76(9/10):924–941
- Levant A (2010) Chattering analysis. *IEEE Trans Autom Control* 55(6):1380–1389
- Levant A, Livne M (2015) Uncertain disturbances' attenuation by homogeneous MIMO sliding mode control and its discretization. *IET Control Theory Appl* 9(4):515–525
- Levant A, Livne M (2016) Weighted homogeneity and robustness of sliding mode control. *Automatica* 72(10):186–193
- Levant A, Livne M (2019) Robust exact filtering differentiators. *European Journal of Control*, available online <https://doi.org/10.1016/j.ejcon.2019.08.006>
- Levant A, Shustin B (2018) Quasi-continuous MIMO sliding-mode control. *IEEE Trans Autom Control* 63(9):3068–3074
- Levant A, Yu X (2018) Sliding-mode-based differentiation and filtering. *IEEE Trans Autom Control* 63(9):3061–3067
- Levant A, Livne M, Yu X (2017) Sliding-mode-based differentiation and its application. In: *Proceedings of the 20th IFAC world congress*, Toulouse, July 9–14, 2017
- Livne M, Levant A (2014) Proper discretization of homogeneous differentiators. *Automatica* 50:2007–2014
- Orlov Y (2008) Discontinuous systems: Lyapunov analysis and robust synthesis under uncertainty conditions. Springer Science & Business Media, London
- Polyakov A, Efimov D, Perruquetti W (2015) Finite-time and fixed-time stabilization: implicit Lyapunov function approach. *Automatica* 51(1):332–340
- Rajamani R (2005) Vehicle dynamics and control. Springer, New York
- Shtessel Y, Edwards C, Fridman L, Levant A (2014) Sliding mode control and observation. Birkhauser, New York
- Slotine JJ, Li W (1991) Applied nonlinear control. Prentice Hall, Englewood Cliffs
- Utkin V (1992) Sliding modes in control and optimization. Springer, Berlin
- Utkin V, Guldner J, Shi J (2009) Sliding mode control in electro-mechanical systems. CRC Press, Boca Raton

Small Phase Theorem

Wei Chen, Dan Wang, and Li Qiu

Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong, China

Abstract

The small gain theorem is one of the most important results in the theory of robust control. It lays the foundation for the traditional gain-based analysis and synthesis, especially within the $\mathcal{H}_\infty$ control paradigm. This entry is concerned with the small phase theorem, which can be regarded as a fitting counterpart to the small gain theorem. With the aid of a suitable definition of complex matrix phases, small phase results for matrices and linear time-invariant (LTI) systems are described. Together, they pave the way for a comprehensive theory of phases of complex systems.

Keywords

Matrix phases · Phase response · Matrix small phase theorem · LTI small phase theorem

Introduction

In classical frequency domain analysis of single-input-single-output (SISO) systems, the magnitude (gain) response and phase response go hand in hand. The broad-ranging research into the multi-input-multi-output (MIMO) systems theory over the past several decades has elevated the concept of magnitude to a great height. In particular, the small gain theorem has come about as one of the most significant milestones in robust control theory, and it underpins the development of the celebrated $\mathcal{H}_\infty$ control. In contrast, a widely accepted parallel small phase theorem has been all but lacking for more than half a century. One main reason is that while the magnitudes of a complex matrix are commonly characterized by

its singular values, it is not until recently that a suitable definition of phase(s) of a complex matrix has been discovered.

This entry describes small phase theorems for matrices and LTI systems, based on the definition of phases of a complex matrix in Furtado and Johnson (2001) and Wang et al (2020). They open the door to a complete phase theory of complex systems.

Phases of a Complex Matrix

It is well understood that an $n \times n$ complex matrix C has n magnitudes, served by the n singular values $\sigma(C) = [\sigma_1(C) \ \sigma_2(C) \ \dots \ \sigma_n(C)]$ with $\bar{\sigma}(C) = \sigma_1(C) \geq \sigma_2(C) \geq \dots \geq \sigma_n(C) = \underline{\sigma}(C)$. The magnitudes of a matrix possess plentiful nice properties, among which the following majorization inequality is of particular interest to the control community.

Given $x, y \in \mathbb{R}^n$, denote by $x^\downarrow$ and $y^\downarrow$ the rearranged versions of x and y so that their elements are sorted in a non-increasing order. Then, x is said to be *majorized* by y , denoted by $x \prec y$, if $\sum_{i=1}^k x_i^\downarrow \leq \sum_{i=1}^k y_i^\downarrow$, $k = 1, \dots, n-1$, and $\sum_{i=1}^n x_i^\downarrow = \sum_{i=1}^n y_i^\downarrow$. When x and y are nonnegative, x is said to be *log-majorized* by y , denoted by $x \prec_{\log} y$, if $\prod_{i=1}^k x_i^\downarrow \leq \prod_{i=1}^k y_i^\downarrow$, $k = 1, \dots, n-1$, and $\prod_{i=1}^n x_i^\downarrow = \prod_{i=1}^n y_i^\downarrow$. The magnitudes of a matrix product satisfy (Marshall et al 2011)

$$\sigma(AB) \prec_{\log} \sigma(A) \odot \sigma(B), \quad (1)$$

where $\odot$ denotes the Hadamard product, i.e., the elementwise product.

An early attempt in Postlethwaite et al (1981) defined the phases of C as the complex arguments of the eigenvalues of the unitary part of its polar decomposition. However, phases defined this way do not enjoy certain desired properties.

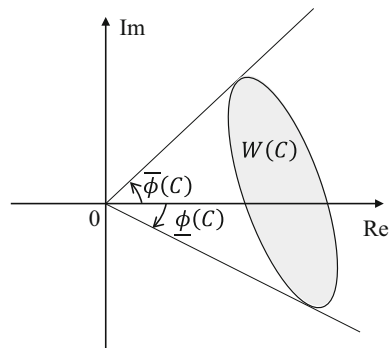
A more appropriate definition of matrix phases that is naturally amenable to small phase type robustness analysis is one that is based on numerical range (Furtado and Johnson 2001; Wang et al 2020). The numerical range, also called field of

values, of a matrix $C \in \mathbb{C}^{n \times n}$ is defined as $W(C) = \{x^* C x : x \in \mathbb{C}^n \text{ with } \|x\| = 1\}$, which, as a subset of $\mathbb{C}$, is compact and convex and contains the spectrum of C .

If $0 \notin W(C)$, then $W(C)$ is in an open half complex plane. In this case, C is said to be a sectorial matrix which admits a sectorial decomposition $C = T^* D T$, where T is nonsingular and D is a diagonal unitary matrix that is unique up to a permutation (Horn and Steinberg 1959; Zhang 2015). The phases of C , denoted by $\bar{\phi}(C) = \phi_1(C) \geq \phi_2(C) \geq \dots \geq \phi_n(C) = \underline{\phi}(C)$, are defined to be the complex arguments of the eigenvalues of D so that $\bar{\phi}(C) - \underline{\phi}(C) < \pi$ and $\gamma(C) = \frac{\bar{\phi}(C) + \underline{\phi}(C)}{2}$, called the phase center of C , lies in $(-\pi, \pi]$. Define $\phi(C) = [\phi_1(C) \ \phi_2(C) \ \dots \ \phi_n(C)]$.

A graphical interpretation of the phases of a matrix C is provided in Fig. 1. The two angles from the positive real axis to each of the two supporting rays of $W(C)$ extending from the origin are $\bar{\phi}(C)$ and $\underline{\phi}(C)$, respectively. The other phases of C lie in between.

A naive analogy of the inequality (1) would hint at $\phi(AB) \prec \phi(A) + \phi(B)$. This, unfortunately, fails to hold even for positive definite A and B . Notwithstanding, if we consider instead $\lambda(AB) = [\lambda_1(AB) \ \dots \ \lambda_n(AB)]$, i.e., the vector of eigenvalues of AB , the following weaker but useful result can be derived (Ballantine and Johnson 1975; Wang et al 2020).



Small Phase Theorem, Fig. 1 A geometric interpretation of $\bar{\phi}(C)$ and $\underline{\phi}(C)$

Theorem 1 Let $A, B \in \mathbb{C}^{n \times n}$ be sectorial, and $\angle \lambda(AB)$ take values in $(\gamma(A) + \gamma(B) - \pi, \gamma(A) + \gamma(B) + \pi)$. Then $\angle \lambda(AB) \prec \phi(A) + \phi(B)$.

Matrix Small Phase Theorem

The singularity of the matrix $I + AB$ plays an important role in the stability analysis of feedback systems. The gain majorization relation (1) underpins the well-known matrix small gain theorem: If $\bar{\sigma}(A)\bar{\sigma}(B) < 1$, then $I + AB$ is nonsingular (Stewart and Sun 1990).

As for a counterpart, we have the following matrix small phase theorem (Wang et al 2020) that is underlied by the phase majorization relation in Theorem 1.

Theorem 2 (Matrix small phase theorem) If $\bar{\phi}(A) + \bar{\phi}(B) < \pi$ and $\underline{\phi}(A) + \underline{\phi}(B) > -\pi$, then $I + AB$ is nonsingular.

It is also known that the matrix small gain theorem is necessary in the following sense: For a given matrix A , there exists a matrix B satisfying $\bar{\sigma}(A)\bar{\sigma}(B) = 1$ such that $I + AB$ loses rank. In the same spirit, the above matrix small phase theorem is necessary in the following sense: For a given matrix A , there exists a matrix B satisfying $\bar{\phi}(A) + \bar{\phi}(B) = \pi$ or $\underline{\phi}(A) + \underline{\phi}(B) = -\pi$ such that $I + AB$ loses rank.

Phase Response of a MIMO LTI System

Let G be an $m \times m$ real rational proper stable transfer matrix, i.e., $G \in \mathcal{RH}_\infty^{m \times m}$. Then $\sigma(G(j\omega))$, the vector of singular values of $G(j\omega)$, is an $\mathbb{R}^m$ -valued function of the frequency, which we call the *magnitude response* of G . The $\mathcal{H}_\infty$ norm of G , defined by $\|G\|_\infty = \sup_{\omega \in \mathbb{R}} \bar{\sigma}(G(j\omega))$, is of particular importance.

Suppose $G(j\omega)$ is sectorial for all $\omega \in [0, \infty]$. Such a system is called a frequency-wise sectorial system. Also, assume for simplicity that $W(G(j\omega))$ does not intersect the negative real axis for all $\omega \in [0, \infty]$. Then $\phi(G(j\omega))$, the vector of phases of $G(j\omega)$ with each element taking values in $(-\pi, \pi)$, is well defined as an

$\mathbb{R}^m$ -valued function of the frequency, which we call the *phase response* of G . We define the $\mathcal{H}_\infty$ phase of G , as the counterpart to its $\mathcal{H}_\infty$ norm, to be $\Phi_\infty(G) = \sup_{\omega \in \mathbb{R}} \bar{\phi}(G(j\omega))$. Clearly, $\Phi_\infty(G) \leq \pi$. By plotting $\sigma(G(j\omega))$ and $\phi(G(j\omega))$ side by side, we have the MIMO Bode plot of G .

The notion of phase of MIMO LTI systems provides a means to unify the concepts of positive real systems and negative imaginary systems, in addition to generalizing the SISO system phase and much more. For instance, $G \in \mathcal{RH}_\infty^{m \times m}$ is said to be strongly positive real if $G(j\omega) + G^*(j\omega) > 0$ for all $\omega \in [0, \infty]$ (Liu and Yao 2016). In the language of phase, this is equivalent to $\Phi_\infty(G) < \frac{\pi}{2}$.

LTI Small Phase Theorem

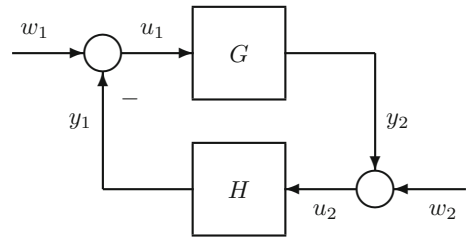
Suppose $G, H \in \mathcal{RH}_\infty^{m \times m}$. The feedback interconnection of G and H , as depicted in Fig. 2, is said to be stable if the gang of four matrix

$$G\#H = \begin{bmatrix} (I + HG)^{-1} & (I + HG)^{-1}H \\ G(I + HG)^{-1} & G(I + HG)^{-1}H \end{bmatrix}$$

is stable, i.e., $G\#H \in \mathcal{RH}_\infty^{2m \times 2m}$.

A version of the famous small gain theorem states that if $\bar{\sigma}(G(j\omega))\bar{\sigma}(H(j\omega)) < 1$ for all $\omega \in \mathbb{R}$, then $G\#H$ is stable (Zhou et al 1996). As its counterpart, the small phase theorem (Chen et al 2019) is in order.

Theorem 3 (LTI small phase theorem) If $\bar{\phi}(G(j\omega)) + \bar{\phi}(H(j\omega)) < \pi$ for all $\omega \in \mathbb{R}$, then $G\#H$ is stable.



Small Phase Theorem, Fig. 2 A standard feedback system

The small phase theorem generalizes a stronger version of the passivity theorem, which states that $G\#H$ is stable if both G and H are strongly positive real (Liu and Yao 2016).

Note that the small gain theorem provides a quantifiable tradeoff between the gains of G and H , while the above small phase theorem does the same with respect to the phases of G and H . In early literature, the notions of input feedforward passivity index and output feedback passivity index (Bao and Lee 2007; Kottenstette et al 2014; Vidyasagar 1981; Wen 1988) have been used to characterize the tradeoff between the surplus and deficit of passivity in open-loop systems. The concept of MIMO system phases provides an alternative that may be even more suited to this task. Specifically, $\frac{\pi}{2} - \Phi_{\infty}(G)$ gives a natural measure of passivity of system G , which we call the *angular passivity index*. The small phase theorem above indicates that if the sum of the angular passivity indexes of G and H are positive, then $G\#H$ is stable. In addition, $\pi - \Phi_{\infty}(GH)$ yields a natural phase margin for a stable $G\#H$.

It is known that the LTI small gain theorem is necessary in the following sense. Let $r \in \mathcal{RH}_{\infty}$ be a scalar transfer function and define a ball of systems

$$\mathcal{B}(r) = \left\{ H \in \mathcal{RH}_{\infty} : \overline{\sigma}(H(j\omega)) < |r(j\omega)| \text{ for all } \omega \in \overline{\mathbb{R}} \right\}.$$

Then, $G\#H$ is stable for all $H \in \mathcal{B}(r)$ if and only if $\overline{\sigma}(G(j\omega)) \leq \frac{1}{|r(j\omega)|}$ for all $\omega \in \mathbb{R}$. Analogously, the above LTI small phase theorem is necessary in the following sense. Let $h \in \mathcal{RH}_{\infty}$ be a scalar strongly positive real function and define a cone of systems

$$\mathcal{C}(h) = \left\{ H \in \mathcal{RH}_{\infty} : \overline{\phi}(H(j\omega)) < \pi/2 + \angle h(j\omega) \text{ for all } \omega \in \overline{\mathbb{R}} \right\}.$$

Then, $G\#H$ is stable for all $H \in \mathcal{C}(h)$ if and only if $\overline{\phi}(G(j\omega)) \leq \pi/2 - \angle h(j\omega)$ for all $\omega \in \mathbb{R}$.

Summary and Future Directions

Small phase theorem kicks off an exciting journey of developing a comprehensive phase theory of complex systems, complementing the existing gain-based system and control theory. A sectorized real lemma (a phasic counterpart to the bounded real lemma) can be found in Chen et al (2019), which gives a state space characterization of phase-bounded MIMO LTI systems in terms of linear matrix inequalities. The implications of small phase theorem in large-scale dynamical networks are worth the investigation. Similar to the fact that the small gain theorem nurtured the celebrated $\mathcal{H}_{\infty}$ norm optimal control, the small phase theorem is expected to cultivate an $\mathcal{H}_{\infty}$ phase optimal control theory, which opens up a vast space for future research.

Some preliminary studies indicate that extending the phase concept to nonlinear systems is a promising direction. How does one define the phase of a nonlinear system in a meaningful manner? How is it related to the classical Lyapunov approach? Can one formulate a nonlinear small phase theorem? Studying these questions and many others will lead to a powerful nonlinear system phase theory.

Gain and phase not only complement each other, but they also interact intimately. When they are integrated, miracles ensue. Understanding their exact relations and eventually coming up with a combined gain/phase theory will have a profound impact in the evolution of system and control theory.

Cross-References

- [H-Infinity Control](#)
- [KYP Lemma and Generalizations/Applications](#)

Recommended Reading

For more details on the newest developments of matrix phases, one can refer to Wang et al (2020), where a collection of properties of matrix phases are studied and the matrix small phase theorem

is provided. Earlier explorations of matrix phases and sectorial matrices can be found in Ballantine and Johnson (1975); Furtado and Johnson (2001); Arlinskiĭ and Popov (2003); Zhang (2015); Horn and Steinberg (1959); Horn and Johnson (1991).

The phase response of a MIMO system and the LTI small phase theorem are detailed in Chen et al (2019). The concept of phase has a great potential for unifying the notions of positive real systems, passive systems, negative imaginary systems, and counterclockwise dynamics, on which the representative works include Bao and Lee (2007), Wen (1988), Kottenstette et al (2014), Lanzon and Petersen (2008), Mabrok et al (2015), and Angeli (2006). The results described in this article are closely related to the theory of integral quadratic constraints and the (generalized) KYP lemma. See the seminal papers Iwasaki and Hara (1998, 2005) and the recent work in Khong and Kao (2019).

Bibliography

- Angeli D (2006) Systems with counterclockwise input-output dynamics. *IEEE Trans Autom Control* 51(7):1130–1143
- Arlinskiĭ YM, Popov AB (2003) On sectorial matrices. *Linear Algebra Appl* 370:133–146
- Ballantine CS, Johnson CR (1975) Accretive matrix products. *Linear Multilinear Algebra* 3(3):169–185
- Bao J, Lee PL (2007) *Process Control: the Passive Systems Approach*. Springer, London
- Chen W, Wang D, Khong SZ, Qiu L (2019) Phase analysis of MIMO LTI systems. In: 58th IEEE Conference on Decision and Control, pp 6062–6067
- Furtado S, Johnson CR (2001) Spectral variation under congruence. *Linear Multilinear Algebra* 49(3):243–259
- Horn A, Steinberg R (1959) Eigenvalues of the unitary part of a matrix. *Pac J Math* 9(2):541–550
- Horn RA, Johnson CR (1991) *Topics in Matrix Analysis*. Cambridge University Press, Cambridge
- Iwasaki T, Hara S (1998) Well-posedness of feedback systems: insights into exact robustness analysis and approximate computations. *IEEE Trans Autom Control* 43(5):619–630
- Iwasaki T, Hara S (2005) Generalized KYP lemma: unified frequency domain inequalities with design applications. *IEEE Trans Autom Control* 50(1):41–59
- Khong SZ, Kao CY (2019) Converse theorems for integral quadratic constraints. *arXiv preprint arXiv:190510600*
- Kottenstette N, McCourt MJ, Xia M, Gupta V, Antsaklis PJ (2014) On relationships among passivity, positive realness, and dissipativity in linear systems. *Automatica* 50(4):1003–1016
- Lanzon A, Petersen IR (2008) Stability robustness of a feedback interconnection of systems with negative imaginary frequency response. *IEEE Trans Autom Control* 53(4):1042–1046
- Liu KZ, Yao Y (2016) *Robust Control: Theory and Applications*. John Wiley & Sons, Singapore
- Mabrok M, Kallapur AG, Petersen IR, Lanzon A (2015) A generalized negative imaginary lemma and Riccati-based static state-feedback negative imaginary synthesis. *Syst Control Lett* 77:63–68
- Marshall AW, Olkin I, Arnold BC (2011) *Inequalities: Theory of Majorization and Its Applications*. Springer, London
- Postlethwaite I, Edmunds J, MacFarlane A (1981) Principal gains and principal phases in the analysis of linear multivariable feedback systems. *IEEE Trans Autom Control* 26(1):32–46
- Stewart G, Sun JG (1990) *Matrix Perturbation Theory*. Academic Press, Boston
- Vidyasagar M (1981) Input-Output Analysis of Large-Scale Interconnected Systems: Decomposition, Well-Posedness and Stability. Springer, Berlin
- Wang D, Chen W, Khong SZ, Qiu L (2020) On the phases of a complex matrix. *Linear Algebra Appl* 593:152–179
- Wen JT (1988) Robustness analysis based on passivity. In: 1988 American control conference, pp 1207–1213
- Zhang F (2015) A matrix decomposition and its applications. *Linear Multilinear Algebra* 63(10):2033–2042
- Zhou K, Doyle JC, Glover K (1996) *Robust and Optimal Control*. Prentice-Hall, Inc., Upper Saddle River

Small Signal Stability in Electric Power Systems

Vijay Vittal

Arizona State University, Tempe, AZ, USA

Abstract

Small signal rotor angle stability analysis in power systems is associated with insufficient damping of oscillations under small disturbances. Rotor angle oscillations due to insufficient damping have been observed in many power systems around the world. This entry overviews the predominant approach to examine small signal rotor angle stability in large power systems using eigenvalue analysis.

Keywords

Eigenvalues · Eigenvectors · Low-frequency oscillations · Mode shape · Oscillatory modes · Participation factors · Small signal rotor angle stability

Small Signal Rotor Angle Stability in Power Systems

As power system interconnections grew in number and size, automatic controls such as voltage regulators played critical roles in enhancing reliability by increasing the synchronizing capability between the interconnected systems. As technology evolved the capabilities of voltage regulators to provide synchronizing torque following disturbances were significantly enhanced. It was, however, observed that voltage regulators tended to reduce damping torque, as a result of which the system was susceptible to rotor angle oscillatory instability. An excellent exposition of the mechanism and the underlying analysis is provided in the textbooks (Anderson and Fouad 2003; Sauer and Pai 1998; Kundur 1993), and a number of practical aspects of the analysis are detailed in Eigenanalysis and Frequency Domain Methods for System Dynamic Performance (1989) and Rogers (2000). Two types of rotor angle oscillations are commonly observed. Low-frequency oscillations involving synchronous machines in different operating areas are commonly referred to as inter-area oscillations. These oscillations are typically in the 0.1–2 Hz frequency range. Oscillations between local machines or a group of machines at a power plant are referred to as plant mode oscillations. These oscillations are typically above the 2 Hz frequency range. The modes associated with rotor angle oscillations are also termed inertial modes of oscillation. Other modes of oscillations associated with the various controls also exist. With the integration of significant new wind and photovoltaic generation which are interconnected to the grid using converters, new modes of oscillation involving the converter controls and conventional synchronous generator states are being observed.

The basis for small signal rotor angle stability analysis is that the disturbances considered are small enough to justify the use of linear analysis to examine stability (Kundur et al. 2004). As a result, Lyapunov's first method Vidyasagar (1993) provides the analytical underpinning to analyze small signal stability. Eigenvalue analysis is the predominant approach to analyze small signal rotor angle stability in power systems. Commercial software packages that utilize sophisticated algorithms to analyze large-scale power systems with the ability to handle detailed models of power system components exist.

The power system representation is described by a set of nonlinear differential algebraic equations shown in (1)

$$\begin{aligned}\dot{x} &= f(x, z) \\ 0 &= g(x, z)\end{aligned}\quad (1)$$

where x is the state vector and z is a vector of algebraic variables. Small signal stability analysis involves the linearization of (1) around a system operating point which is typically determined by conducting a power flow analysis:

$$\begin{bmatrix} \Delta \dot{x} \\ 0 \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \end{bmatrix}\quad (2)$$

The power system state matrix can be obtained by eliminating the vector of algebraic variables Δz in (2)

$$\Delta \dot{x} = (J_1 - J_2 J_4^{-1} J_3) \Delta x = A \Delta x \quad (3)$$

where A represents the system state matrix. Based on Lyapunov's first method, the eigenvalues of A characterize the small signal stability behavior of the nonlinear system in a neighborhood of the operating point around which the system is linearized. The eigenvectors corresponding to the eigenvalues also provide significant qualitative information. For each eigenvalue λ_i , there exists a vector u_i known as the right eigenvector of A which satisfies the equation

$$A u_i = \lambda_i u_i \quad (4)$$

There also exists a row vector v_i known as the left eigenvector of A which satisfies

$$v_i A = \lambda_i v_i \quad (5)$$

For a system which has distinct eigenvalues, the right and left eigenvectors form an orthogonal set governed by

$$\begin{aligned} v_i u_j &= k_{ij} \\ \text{where} \\ k_{ij} &\neq 0 \quad i = j \\ k_{ij} &= 0 \quad i \neq j \end{aligned} \quad (6)$$

One set (either right or left) of eigenvectors are usually scaled to unity and the other set obtained by solving (6) with $k_{ij} = 1$. The right eigenvectors can be assembled together as columns of a square matrix U , and the corresponding left eigenvectors can be assembled as rows of a matrix V ; then

$$V = U^{-1} \quad (7)$$

and

$$V A U = \Lambda \quad (8)$$

where Λ is a diagonal matrix with the distinct eigenvalues as the diagonal entries. The relationship in (8) is a similarity transformation and in the case of distinct eigenvalues provides a pathway to obtain solutions to the linear system of equations (3). Applying the following similarity transformation to (3)

$$\Delta x = U z \rightarrow \Delta x_i(t) = \sum_{j=1}^n u_{ij} z_j e^{\lambda_j t} \quad (9)$$

$$U \dot{z} = A U z \quad (10)$$

$$\dot{z} = U^{-1} A U z = V A U z = \Lambda z \quad (11)$$

$$\dot{z}_i(t) = \lambda_i z_i \Rightarrow z_i(t) = z_i(0) e^{\lambda_i t} \quad (12)$$

$$z_i(0) = v_i^T \Delta x(0) \quad (13)$$

$$z_i(t) = v_i^T \Delta x(0) e^{\lambda_i t} \quad (14)$$

From (9) and (14), it can be observed that the right eigenvector describes how each mode of the system is distributed throughout the state vector (and is referred to as the mode shape), and the left eigenvector in conjunction with the initial conditions of the system state vector determines the magnitude of the mode. The right eigenvector or the mode shape has been often used to identify dynamic patterns in small signal dynamics. One problem with the mode shape is that it is dependent on the units and scaling of the state variables as a result of which it is difficult to compare the magnitudes of entries that are disparate and correspond to states that impact the dynamics differently. This resulted in the development of the participation factors (Pérez-Arriaga et al. 1982) which are dimensionless and independent of the choice of units. The participation factor is expressed as

$$p_{ik} = v_{ik} u_{ik} \quad (15)$$

The magnitude of the participation factor measures the relative participation of the i th state variable in the k th mode and vice versa.

Small Signal Stability Analysis Tools for Large Power Systems

Efficient software tools exist that facilitate the application of the methods in section “[Small Signal Rotor Angle Stability in Power Systems](#)” to large power systems (Powertech 2012; Martins 1989). These tools incorporate detailed models of power system components and also leverage the sparsity in power systems. The building of the A matrix is a complex task for large power systems with a multitude of dynamic components. The approach in Powertech (2012) utilizes a technique where state space equations are developed for each dynamic component in the system using a solved power flow solution and the dynamic data description for a given system. These state space equations are then coupled based on the system topology, and the system A matrix is derived as in (3). Reference Martins (1989) takes advantage of the sparsity of the Jacobian matrix

in (2) and develops efficient algorithms to determine the eigenvalues and eigenvectors. The software tools also provide the flexibility of a number of different options with regard to eigenvalue computations:

1. Calculation of a specific eigenvalue at a specified frequency or with a specified damping ratio
2. Simultaneous calculation of a group of relevant eigenvalues in a specified frequency range or in specified damping ratio range

In addition to the features described above, commercial software packages also provide features to evaluate:

1. Frequency response plots
2. Participation factors
3. Transfer functions, residues, controllability, and observability factors
4. Linear time response to step changes
5. Eigenvalue sensitivities to changes in specified parameters

Applications of Small Signal Stability Analysis in Power Systems

Small signal stability analysis tools are used for a range of applications in power systems. These applications include:

Analysis of local stability problems – These types of stability problems are primarily associated with the tuning of control associated with the synchronous generator, converter interconnected renewable resources, and HVDC link current control. In certain cases analysis of local stability problems could also involve design of supplementary controllers which enhance the stability region. Since the stability problem pertains to a local portion of the power system, there is significant flexibility in modeling the system. In many instances local stability problems facilitate the use of a simple representation of a power system which could include the particular machine or

a local group of machines in question together with a highly equivalenced representation of the rest of the system. In cases where controls other than generator controls influence stability, e.g., static VAR compensators or HVDC links, the system representation would need to be extended to include portions of the system where these devices are located. Typical small signal stability problems that are analyzed include:

1. Power system stabilizer design
2. Automatic voltage regulator tuning
3. Governor tuning
4. DC link current control
5. Small signal stability analysis for subsynchronous resonance
6. Load modeling effects on small signal stability

References Eigenanalysis and Frequency Domain Methods for System Dynamic Performance (1989) and Rogers (2000) provide comprehensive examples of the analysis conducted for each of the problems listed above.

Analysis of global stability problems – These types of stability problems are associated with controls that impact generators located in different areas of the power systems. The analysis of these inter-area problems requires a more systematic approach and involves representation of the power system in greater detail. The problems that are analyzed under this category include:

1. Power system stabilizer design
2. HVDC link modulation
3. Static VAR compensator controls

References Eigenanalysis and Frequency Domain Methods for System Dynamic Performance (1989) and Rogers (2000) again provide details of the analysis conducted for each of the problems listed under this category.

Cross-References

- [Lyapunov Methods in Power System Stability](#)
- [Lyapunov's Stability Theory](#)

- [Power System Voltage Stability](#)
- [Stability: Lyapunov, Linear Systems](#)

Bibliography

- Anderson PM, Fouad AA (2003) Power system control and stability, 2nd edn. Wiley Interscience, Hoboken
- Eigenanalysis and Frequency Domain Methods for System Dynamic Performance (1989) IEEE Special Publication, 90TH0292-3-PWR
- Kundur P (1993) Power system stability and control. McGraw Hill, San Francisco
- Kundur P, Paserba J, Ajarapu V, Andersson G, Bose A, Canizares C, Hatziaargyriou N, Hill D, Stankovic A, Taylor C, Van Cutsem T, Vittal V (2004) Definition and classification of power system stability. IEEE/CIGRE joint task force on stability terms and definitions report. IEEE Trans Power Syst 19:1387–1401
- Martins N (1989) Efficient eigenvalue and frequency response methods applied to power system small signal stability studies. IEEE Trans Power Syst 1:74–82
- Pérez-Arriaga JJ, Verghese GC, Schweppe FC (1982) Selective modal analysis with applications to electric power systems, part 1: Heuristic introduction. IEEE Trans Power Appar Syst 101:3117–3125
- Powertech (2012) Small signal analysis tool (SSAT) user manual. Powertech Labs Inc, Surrey
- Rogers G (2000) Power system oscillations. Kluwer Academic, Dordrecht
- Sauer PW, Pai MA (1998) Power system dynamics and stability. Prentice Hall, Upper Saddle River
- Vidyasagar M (1993) Nonlinear systems analysis, 2nd edn. Prentice Hall, Englewood Cliffs

Space Robotics

Marcello Romano
Department of Mechanical and Aerospace
Engineering, Naval Postgraduate School,
Monterey, CA, USA

Abstract

Space robotics enables humankind to pioneer the exploration and utilization of space. This article gives a general introduction to the science and technology of space robotics. First,

the article defines space robotics, introduces some fundamental concepts and lists its foundational disciplines. Then, a key mathematical model is presented. Finally, a survey of the state of the art is given, together with a perspective on future developments and pointers to additional information. This article was written as a self-contained introduction. It is directed at space science and engineering students, new space researchers and practitioners, as well as at a broader audience interested in astronautics.

Keywords

Astronautics · Orbital robotics · Planetary robotics

Introduction

Space robotics is defined as the science concerned with the study of engineering and operations of spacecraft systems that can *physically interact with another object in space* by using actively controlled onboard mechanisms that enable *manipulation* or *mobility*. In this definition, a spacecraft stands for a space vehicle that is either manned or unmanned. That space vehicle can be either orbiting a celestial body (Earth, Moon, Sun, or other Solar System planets, moons, asteroids, and comets) – and in this case is also referred to as an artificial satellite – or operating on the surface of an extraterrestrial celestial body.

Space robotics can be further divided into two large subcategories:

Orbital robotics, a.k.a. Microgravity Robotics, is concerned with orbiting space robotic systems: a paradigmatic example is an artificial satellite equipped with an onboard robotic manipulator that enables grasping and manipulating other orbiting objects for servicing, assembly of space stations or other large structures, and removal of orbital debris.

Planetary robotics is concerned with space robotic systems operating on the surface of an extraterrestrial body: a paradigmatic example is

a space vehicle landing on the surface of the Moon or Mars and releasing a robotic rover for exploring nearby the landing site. The rover can be equipped with a robotic manipulator for delivering instruments to the surface.

Space robotics is a challenging endeavor with one of the highest ratios of cost per mission over rate of success, among science and engineering fields. Primary difficulties in the design and operation of space robotic systems include mass, volume and power limitations, reduced gravity (which affects many phenomena, including mechanical, thermal, fluidic, and biological ones), harsh environments with extreme temperature fluctuations, radiation, remoteness of operation, need of radio communication with a ground station over either a direct path or relayed by other spacecraft, and presence of a delay in communication due to distance (e.g., quantifiable as a fraction of a second around the Earth, a few seconds at the Moon, and a few minutes at Mars). Additional difficulties for planetary robotics include a priori unknown or uncertain terrain characteristics and the need to avoid cross-planetary contamination.

Space robotics is an essential aspect of many of the current and future space exploration and utilization missions, and it will play an increasingly important role as technological progress continues in computing, control, space systems engineering, and robotic autonomy. For a historical perspective on space robotics, also see Gao and Chien (2017).

Fundamentals

This section introduces the basic functional principles, technologies, and engineering practices that are utilized in space robotics.

A spacecraft is a self-contained vehicle system designed to operate in space. It uses electrical energy, commonly produced by onboard solar panels and stored in batteries, to power its components. The main subsystems of a spacecraft include structure, electrical, computing, communication, thermal control, sensors and actuators, propulsion, payload instruments, mechanisms,

and software. A spacecraft is launched into space from the surface of the Earth as a payload of a multistage rocket launch vehicle or launcher. After orbit insertion the spacecraft is separated from the launcher and is in a natural (i.e., when control actions are off) dynamic state of free falling and free tumbling, in a quasi-vacuum environment. The center of mass of a spacecraft moves along an orbit, which – in most cases – can be modeled as either a circle or an ellipse with the center of the attracting body (e.g., the Earth) at one of its foci. Minute environmental actions (forces and torques) on an orbiting spacecraft typically have a negligible effect over a short time (relative to a conventionally defined “time constant” of the system, such as the orbital period) but a substantial effect over a longer time. The most important environmental actions might include residual aerodynamic forces and torques; solar pressure forces and torques; magnetic torques; and gravity gradient forces and torques.

After orbit insertion, a spacecraft is maneuvered by control actions in order to reach and maintain the orbit and orientation needed to fulfill its mission. The most common actuators to maneuver a spacecraft along its orbit, i.e., to generate an external force that changes its orbital velocity, are jet thrusters. A jet thruster works on the basis of the conservation of linear momentum. It generates a force on the spacecraft by expelling propellant in the opposite direction. The most common actuators to reorient a spacecraft, i.e., to generate a torque that changes its angular velocity and orientation, are again jet thrusters but also magnetic actuators and momentum-exchange devices.

Sensors commonly used for space robotic systems include onboard attitude navigation sensors, such as Sun sensors, star sensors, magnetometers, and rate gyroscopes; onboard orbital navigation sensors, such as optical sensors, Global Position System (GPS) receivers (for some of the orbits around Earth), and celestial navigation sensors; Earth-based navigation sensors, such as radar and electromagnetic range and velocity sensors; onboard configuration sensors, such as displacement sensors at the joints of robotic manipulators; and onboard relative navigation sensors like

artificial vision systems and radar for determining the relative position and velocity of a co-orbiting target object.

An interplanetary spacecraft usually requires considerably more complicated orbital maneuvers than a spacecraft orbiting around the Earth. In fact, an interplanetary flight involves orbiting different central bodies in sequence, including, for instance, Earth, Sun, and Mars. For planetary space robotic systems, those maneuvers include also a descent and landing on the target body.

Unmanned space robotic systems can be either tele-operated or tele-commanded from a ground station: in the latter case, a sequence of operations is commanded for later execution. Furthermore, unmanned spacecraft can rely – at least partially – on autonomous onboard command and operation scheduling and execution. Autonomous execution and command capability become necessary when the tasks are either of a repetitive nature or impossible to be commanded and executed remotely, because of a communication unreachability or a communication delay that is long with respect to the duration of the task itself. Finally, manned space robotic systems can also be operated by onboard crew members.

Core Disciplines

The core disciplines of space robotics are listed below together with selected references to specialized literature.

Engineering Mechanics: is the science concerned with the study of the physical behavior of material bodies, with interest generally focused on the mathematical description of their configuration and motion. Engineering mechanics is rooted in Newtonian classical physics and mathematical analysis and encompasses the two subtopics of kinematics (position, orientation, and their evolutions in time) and dynamics (effect of forces and torques). Among the many bibliographic references on this subject, Likins (1973) gives a rigorous and complete introduction from the astronautical engineering perspective. Stemming from engineering mechanics are the subjects of multibody

mechanics and mechanisms that are concerned with the modeling of systems consisting of multiple interconnected parts which can move relatively to each other. See, for instance, Wittenburg (2007) and Conley (1998).

Space Flight Mechanics (a.k.a. Astrodynamics or Astronautics): is the science concerned with the study of modeling, designing, and directing the translational and rotational trajectories of space vehicles, including rocket launchers, orbiting artificial satellites, and reentry vehicles. Comprehensive textbooks are here listed from the most introductory to the most advanced: Bate et al. (1971), Vallado and McClain (2013), Walter (2012), Schaub and Junkins (2018), Hughes (2012), and Gurfil and Seidelmann (2016).

Robotics: is the science concerned with the study of systems that can replace human beings in the execution of physical tasks. It is rooted in multibody mechanics and control system engineering. Primary topics include the kinematic, dynamic, motion planning, navigation, and control of robotic systems. Most commonly studied robotic systems are fixed-base robotic manipulators and wheeled rovers. Excellent references on general robotics include Siciliano et al. (2010) and Siciliano and Khatib (2008).

Guidance, Navigation, and Control: guidance is concerned with determining a reference trajectory for the evolution in time of the state of a physical system (i.e., of the configuration and its rate of change). Navigation is concerned with estimating the current state of the system. Finally, control is concerned with using the method of “feedback” in order to command the actuators to act on the system to follow the desired reference trajectory while rejecting disturbances due to un-modeled dynamic effects, uncertainties, and disturbances. Textbooks on these subjects include Kirk (2012) and Wie (2015). Also see the articles by Crassidis and Zanetti and D’Sousa in this book.

Space System Engineering: is concerned with the analysis, design, integration, and testing of spacecraft and their missions. Space System Engineering involves the concurrent and iterative design of all of the subsystems into an integrated spacecraft system. The design of a spacecraft

aims essentially at satisfying specific mission goals while minimizing the overall cost, mass, and volume, the last two factors being upper-bounded by launch constraints. Furthermore, system reliability needs to be maximized. In fact, spacecraft are typically unreachable and unserviceable during their operations. The mentioned factors in spacecraft design result in a general conservative approach and in the use of well-proven technologies that often lag years behind the latest commercially available technologies used in other fields. Standard references include in the area of Space System Engineering Wertz et al. (2011) and Fortescue et al. (2003).

Mathematical Model of an Orbiting Space Robot

This section introduces – for the sake of offering a high-level physical insight – the equations of motion that are used to model a typical space robotic system.

Let us consider the example of a space robotic system consisting of an Earth-orbiting spacecraft equipped with a robotic manipulator. The robotic manipulator consists of a series of rotational joints and rigid links, terminating with an end effector. A formally straightforward extension of the analysis is possible for more complicated systems, including multiple manipulators. Details on the subject are reported, for instance, in Xu and Kanade (1992) and Wilde et al. (2018).

The governing equations of motion of the spacecraft-manipulator system can be written – in compact form – as follows:

$$\mathbf{H}\dot{\mathbf{u}} + \mathbf{C}\mathbf{u} = \boldsymbol{\tau}, \quad (1)$$

where $\mathbf{H} \in \mathcal{R}^{(n+6) \times (n+6)}$ is the *Generalized Inertia Matrix*, which is positive definite and symmetric, n is the number of links of the manipulator (while the base spacecraft is conventionally indicated by the index 0, each successive joint and link of the manipulator are indicated by indexes from 1 to n), $\mathbf{C} \in \mathcal{R}^{(n+6) \times (n+6)}$ is the *Convective Inertia Matrix*, $\mathbf{u} \in \mathcal{R}^{(n+6)}$ is the

Generalized Velocity Matrix, and $\boldsymbol{\tau} \in \mathcal{R}^{(n+6)}$ is the *Generalized Force Matrix*.

The Generalized Inertia Matrix depends on the system geometry, i.e., on the mass and inertia tensor of each of the bodies composing the space robot, and on its current configuration, i.e., on the absolute orientation and position of the base and the relative angular displacement of each of the manipulator links. The Generalized Velocity Matrix contains the angular velocity and linear velocity of the base spacecraft, each having three scalar components on a chosen reference Cartesian coordinate system (e.g., fixed with respect to the inertial frame) and the n relative rotational speeds of the manipulator joints. Furthermore, the Convective Inertia Matrix depends on the system geometry, on the system current configuration, and on its current velocities (elements of the Generalized Velocity Matrix). Finally, the Generalized Force Matrix contains all of the external and internal forces and torques acting on the system, including base-spacecraft actuation forces and torques, manipulator joint torques, forces and torques due to the interaction with another orbiting body, and orbital disturbance force and torques.

Given specific space robotic systems, the above equations of motions are equivalently derived by using either the Lagrangian approach or the Newton-Euler approach. While the former starts with the expression of the kinetic and potential energy of the whole system and then apply the Lagrangian equations formality, the latter builds up the equation of motion by successively taking into account each of the rigid bodies composing the system and applying the Newton's second law of translation and the Euler's law of rotation. The Newton-Euler approach is often preferred in literature because of its iterative nature which offers more scalability up to systems with many degrees of freedom.

In order to better illustrate the mechanics of motion of a spacecraft-manipulator system and, in particular, the occurring dynamic coupling between the base spacecraft and the manipulator, it is useful to rewrite Eq.(1) in the following form:

$$\begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_{0m} \\ \mathbf{H}_{0m}^T & \mathbf{H}_m \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}}_0 \\ \dot{\mathbf{u}}_m \end{bmatrix} + \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_{0m} \\ \mathbf{C}_{m0} & \mathbf{C}_m \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_0 \\ \boldsymbol{\tau}_m \end{bmatrix}, \quad (2)$$

where $\mathbf{H}_0$ is the 6×6 inertia matrix of the base spacecraft, $\mathbf{H}_m$ is the $n \times n$ inertia matrix of the robotic manipulator, and $\mathbf{H}_{0m}$ is the $6 \times n$ dynamic-coupling inertia matrix that expresses the contribution of the dynamic coupling between the manipulator and the base spacecraft to the kinetic energy of the spacecraft-manipulator system. Analogously, for the sub-matrices of the Convective Inertia Matrix $\mathbf{C}_0$, $\mathbf{C}_m$, $\mathbf{C}_{0m}$, and $\mathbf{C}_{m0}$. Furthermore, $\mathbf{u}_0$ contains the six generalized velocities of the base spacecraft, and $\mathbf{u}_m$ contains the n generalized velocities pertaining to the robotic manipulator. Finally, $\boldsymbol{\tau}_0$ contains the six generalized forces on the base spacecraft, and $\boldsymbol{\tau}_m$ contains the n generalized forces at the manipulator joints.

As it is clear from Eq. (2), the motion of the spacecraft base and the robotic manipulator are dynamically coupled. Notably, the coupling effect becomes more important as the ratios between mass and inertia of the manipulator vs mass and inertia of the base increase. At the limit when the mass and inertia of the base are infinitely larger than the mass and inertias of the manipulator, Eq. (2) becomes equivalent to the equations of motion of a fixed-base robotic manipulator.

The above equations of motion of a spacecraft-manipulator system can be used to solve either the *forward dynamics* problem of mapping a history of generalized forces onto the resulting history of motion in terms of base-spacecraft and manipulator motion or the *inverse dynamics* problem of mapping a desired history of motion onto the needed history of generalized forces to generate that motion. Furthermore, the equations of motion can be used to design and simulate the effect of guidance and control methods.

The most common maneuver strategies for a spacecraft-manipulator system are *free-floating* and *free-flying* maneuvers. The former refers to a maneuver during which the base spacecraft is

not controlled by thrusters while the manipulator motion is controlled through the actions at its joints, while the latter refers to a maneuver during which the spacecraft is controlled by thrusters during the manipulator motion. Other type of maneuvers are possible: see Wilde et al. (2018) for more details. In particular, the *free-floating* approach is preferred, whenever the related constrained work-space size is acceptable for the task at hand. In fact, the *free-floating* approach elegantly enables a propellant-less maneuvering by exploiting the dynamic coupling between the base spacecraft and the onboard robotic manipulator.

By assuming that the spacecraft-manipulator system is not subjected to external forces or torques, and by integrating in time the first six equations of Eq. (2), related to the base-link, yields

$$\mathbf{H}_0 \mathbf{u}_0 + \mathbf{H}_{0m} \mathbf{u}_m = \begin{bmatrix} \mathcal{L} \\ \mathcal{P} \end{bmatrix}, \quad (3)$$

with $\mathcal{P}$ denoting the three components of the total absolute linear momentum and $\mathcal{L}$ the three components of the total absolute angular momentum. Both of these quantities are conserved, based on the stated hypotheses. These momenta equations constrain the motion of the spacecraft-manipulator system as an under-actuated system in a nonholonomic fashion. They enable to design *free-floating* maneuvers and to compute the reaction forces between the manipulator and the base spacecraft. See Xu and Kanade (1992) for more details.

Current Efforts and Perspective on Future Developments

Space robotics is listed among the key technologies and aerospace developmental priorities for the decades to come in both the 2015 NASA Technology Roadmap (Miller 2015) and in the recently published 2020 NASA Technology Taxonomy (Terrier 2020).

Notably, space robotics is one of the enabling technologies for the following missions that are considered with high priority worldwide:

On-orbit Servicing, Assembly, and Manufacturing (OSAM). This definition refers to the use

of spacecraft equipped with robotic manipulators or other robotic mechanisms for conducting servicing operations such as inspection, refueling, or maintenance of orbiting satellites; to assemble large structures in space made from smaller constituent modules, such are needed for future space stations, space telescopes, and possibly space solar power generators; to manufacture structure in space, e.g., via 3D printing; or to manufacture engineering or biological items for later use on Earth. Examples of recently flown missions of this kind include the Northrop Grumman's Mission Extension Vehicle (2019) and the NASA Seeker CubeSat (2019). Planned missions include the Defense Advanced Research Projects Agency (DARPA) Robotic Servicing of Geosynchronous Satellites (RSGS), the NASA Restore-L, the Effective Space Solutions' Space Drone, and the Made in Space's Archinaut One.

Space Debris Removal. This refers to the use of spacecraft-manipulator systems for grappling and deorbiting large space debris. A planned mission of this kind is the European Space Agency (ESA) e-Deorbit.

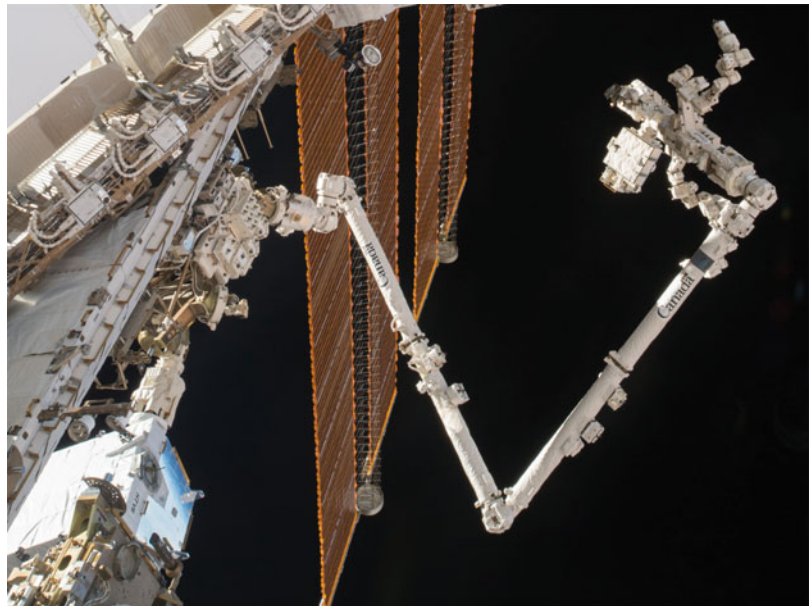
Solar System Exploration and in Situ Utilization. Space robotics will continue to play an essential role in the exploration and

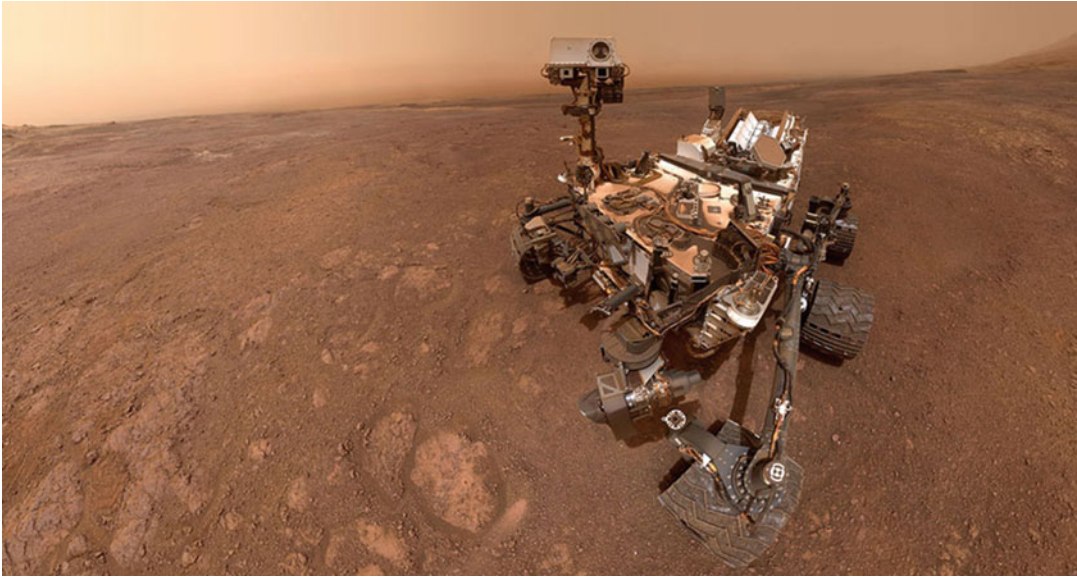
utilization of orbital space and of the solar system's bodies, supporting both unmanned and manned missions (also see Roesler 2019). Examples of recently flown missions of this kind include the Japan Aerospace Exploration Agency (JAXA) Hayabusa-2 spacecraft that performed subsurface sample-gathering touchdown in 2019 on the asteroid Ryugu and the China National Space Administration (CNSA) Chang'e 4, first Spacecraft Lander and Rover deployed on the far side of the Moon (2019), the NASA OSIRIS-REx, launched in 2018 and planned to collect and return a sample from Asteroid Bennu in 2020. Planned missions of this kind include the ESA European Robotic Arm (ERA) to be attached to the Russian segment of the International Space Station, the ESA ExoMars Mars Rover mission, the CNSA Chang'e 5 lunar sample return mission, the NASA Mars 2020 Mars rover mission, the NASA VIPER lunar rover planned to explore permanently shadowed areas in the lunar south pole region, and the NASA Gateway equipped with the Canadarm 3 robotic arm. Also see Figs. 1, 2, and 3.

Space Defense. During the last decade, space has risen to become a contested environment and space defense a key component of the Great

Space Robotics, Fig. 1

The CSA/NASA Canadarm 2, a 17-m long robotic arm and the Special Purpose Dexterous Manipulator (DEXTRE) mounted as its end effector, permanently on-board the International Space Station. (Picture credit: NASA–National Aeronautics and Space Administration and CSA–Canadian Space Agency)





Space Robotics, Fig. 2 A self-portrait of the NASA *Curiosity* Rover on the surface of Mars in January 2019. (Picture Credit: JPL–Jet Propulsion Laboratory,

CALTECH–California Institute of Technology, NASA–National Aeronautics and Space Administration)



Space Robotics, Fig. 3 NASA Astrobee Intra-Vehicular free flyer robot (on the right hand-side of the picture), named Bumble, during a free flight inside the Japanese

Kibo laboratory of the International Space Station in July 2019. (Picture Credit: NASA–National Aeronautics and Space Administration).

Power Competition involving China, Russia, and the United States. This is reflected in the establishment of the US Space Force in December 2019. It is foreseeable that space robotics

technology could play a critical role in both defensive and offensive military capabilities in space. However, it is the author's hope that international policy and diplomacy, under the auspices

of the United Nations Office of Outer Space Affairs (UNOOSA), will be able to avoid an escalation of conflict in space and to contribute to set the main focus of space engineering and science toward peaceful uses for mankind's benefit.

Important areas of active research aiming at meeting the future needs and challenges of space robotics include autonomous rendezvous, proximity operations, docking, and capture; hardware and algorithms for improved sensing and perception; high-level autonomy and failure management and robustness; human-robot interaction; Guidance, Navigation and Control; manipulation and mobility; modeling, simulation, and on-the-ground experimentation (also see Wilde et al. 2019); modularity; standardization of technology and practices, as fostered – for instance – by the Consortium for Execution of Rendezvous and Servicing Operations (CONFERS); system engineering methods; legal aspects; and national and international policy.

Dedication

The author would like to dedicate this article to the memory of astronaut-select Prof. Franco Rossitto, space flight passionate, mentor, and friend.

Cross-References

- [Inertial Navigation](#)
- [Satellite Control](#)
- [Spacecraft Attitude Determination](#)

Bibliography

- Bate R, Mueller D, White J (1971) Fundamentals of astrodynamics. Dover Publications, New York
- Conley P (1998) Space vehicle mechanisms: elements of successful design. Wiley, New York
- Fortescue P, Stark J, Swinerd G (2003) Spacecraft systems engineering. Wiley, New York
- Gao Y, Chien S (2017) Review on space robotics: toward top-level science through space exploration. *Sci Robot* 2:eaan5074
- Gurfil P, Seidelmann P (2016) Celestial mechanics and astrodynamics: theory and practice. Springer, Berlin
- Hughes P (2012) Spacecraft attitude dynamics. Dover Publications, New York
- Kirk D (2012) Optimal control theory: an introduction. Dover Publications, New York
- Likins PW (1973) Elements of engineering mechanics. McGraw-Hill, New York
- Miller D (ed) (2015) NASA Technology Roadmaps, National Aeronautics and Space Administration
- Roesler G (2019) Don't forget the robots. *Aerosp Am* 42–46
- Schaub H, Junkins J (2018) Analytical mechanics of space systems, 4 ed. AIAA, Reston
- Siciliano B, Khatib O (2008) Springer handbook of robotics. Springer, Berlin
- Siciliano B, Sciavicco L, Villani L, Oriolo G (2010) Robotics: modelling, planning and control. Springer, Berlin
- Terrier D (ed) (2020) NASA Technology Taxonomy, National Aeronautics and Space Administration
- Vallado D, McClain W (2013) Fundamentals of astrodynamics and applications, 4 edn. Microcosm Press, Hawthorne
- Walter U (2012) Astronautics, 2 edn. Wiley-VCH, Weinheim
- Wertz J, Everett D, Puschell J (2011) Space mission engineering: the new SMAD. Microcosm Press, Hawthorne
- Wie B (2015) Space vehicle guidance, control, and astrodynamics. AIAA, Reston
- Wilde M, Clark C, Romano M (2019) Historical survey of kinematic and dynamic spacecraft simulators for laboratory experimentation of on-orbit proximity maneuvers. *Progress in Aerospace Sciences*, 110
- Wilde M, Kwok Choon S, Grompone A, Romano M (2018) Equations of motion of free-floating spacecraft-manipulator systems: an engineer's tutorial. *Front Robot AI* 5:41
- Wittenburg J (2007) Dynamics of multibody systems, 2 edn. Springer, Berlin
- Xu Y, Kanade T (1992) Space robotics: dynamics and control. Springer, Berlin

Spacecraft Attitude Determination

John L. Crassidis

Department of Mechanical and Aerospace Engineering, University at Buffalo, The State University of New York, Buffalo, NY, USA

Abstract

Spacecraft attitude determination refers to process of determining the orientation of a vehicle in orbit from onboard attitude sensor

measurements. Knowledge of the attitude is crucial to point a spacecraft to a desired orientation using an attitude control system. The sensor measurements are usually put into vector form, as well as their associated reference vectors. If all the measured and reference vectors are error-free, then the rotation (attitude) matrix is the same for all sets of observations. However, if measurement errors exist, such as noise, then a least-squares type method must be used. Filtering algorithms are typically used to filter noisy measurements, and estimate other quantities such as calibration parameters. This chapter provides an overview of attitude determination and filtering for the purpose of attitude determination of spacecraft in orbit.

Keywords

Wahba's problem · q Method ·
Multiplicative extended Kalman filter

Introduction

Most spacecraft have pointing requirements in order to meet their mission objectives. For example, the Geostationary Operational Environment Satellite (GOES) series, which supply the weather pictures seen on televised weather programs, are required to point their onboard weather sensing devices at the Earth to within a high accuracy in order to obtain detailed images. Other applications are not as straightforward as GOES. For example, the pointing requirements for the International Space Station (ISS) are typically defined to mitigate the effects of external torque disturbances, such as gravity gradient torques and atmospheric drag torques, so that actuator usage can be minimized. All spacecraft have both *knowledge* and *control* requirements that makeup the overall pointing requirements. For example, for the ISS the knowledge requirement is 0.5 degrees per axis (3σ), and the control requirement maintains the attitude ± 5 degrees per axis. Obviously, the control requirement cannot be larger than

the knowledge requirement. The purpose of this chapter is to introduce the basics of spacecraft attitude determination, which is used to fulfill spacecraft knowledge requirements.

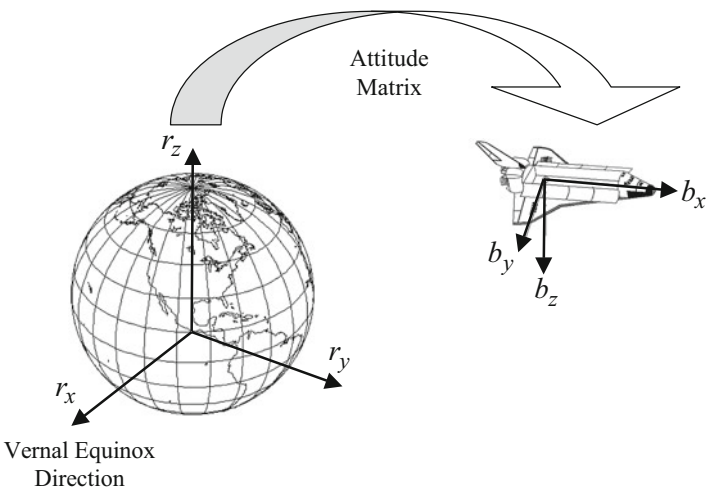
When one uses the term “attitude” to describe the orientation of a spacecraft, one must first describe the frames required to provide the proper context behind the meaning of the attitude. The attitude of a spacecraft is defined as its orientation with respect to some reference frame. If the reference frame is non-moving, then it is commonly referred to as an *inertial* frame. To describe the attitude, two coordinate systems are usually defined: one on the spacecraft body and one on the reference frame. For most dynamical applications, these coordinate systems have orthogonal unit vectors that follow the right-hand rule. The *attitude matrix* (A), often referred to as the direction cosine matrix or rotation matrix, maps one frame to another (for spacecraft kinematics this mapping is usually from the reference frame to the vehicle body frame). A geometrical representation of this concept is shown in Fig. 1, where the r_i ($i = x, y, z$) quantities correspond to the inertial frame and the b_i quantities correspond to the spacecraft body frame.

There are two types of spacecraft attitude determination. One uses any number of measurement observations at a specified time point without other information (such as motion models) to determine the attitude. This type requires at least two non-collinear vectors to determine a solution. Many algorithms are available to solve this type of problem; see Markley and Mortari (2000) for a review of several algorithms. The other type incorporates the measurements with kinematic, or possibly dynamic, models of motion to provide a “filtered” attitude solution. In general, filtered processes provide better accuracy than the first form of spacecraft attitude determination. Many algorithms are available to solve this type of problem as well; see Crassidis et al. (2007) for a survey of several algorithms.

Body frame observations can be obtained from various attitude sensors. Star sensors are used to provide line-of-sight (LOS) vectors to pre-identified stars. These are generally the most accurate attitude sensors, which can provide ≈ 1

Spacecraft Attitude Determination, Fig. 1

Relationship between reference and body frames



arc seconds off boresight and ≈ 100 arc seconds around the boresight information. Also, modern-day star sensors can track multiple stars, thus providing a mechanism to determine an attitude without other sensors. Sun sensors essentially provide the same LOS information as star sensors, but are not as accurate. Earth-horizon sensors provide a direction to the geocenter with accuracies from 0.1° in near-Earth orbit to 0.03° in geostationary orbit using a fixed-head sensor. Three-axis magnetometers measure the three components of the magnetic field, but only two degrees of freedom are actually provided. Obtainable accuracies depend more on the errors in the magnetic field model than on sensor noise in the magnetometer; typical accuracies range from 0.5° to 3° . Sun sensors, Earth-horizon sensors, and three-axis magnetometers cannot be used alone to determine the attitude completely; however, these sensors are often used simultaneously to determine the attitude of a spacecraft. Another attitude sensor relies on signals obtained from the Global Positioning System (GPS). This sensor measures carrier differences using multiple antennae to determine angles to various GPS satellites, which can be used to completely determine the attitude, to about a few degrees, if enough angles are given. Gyros are used to propagate a kinematics model, which, if an accurate initial attitude estimate is given, provide an attitude estimate without external sensors. How-

Spacecraft Attitude Determination, Table 1 Typical attitude sensor performance ranges

Sensor	Typical performance range
Star sensors	0.0003° to 0.01°
Sun sensors	0.005° to 3°
Horizon sensors	0.1° to 1° (Scanner/Pipper) $<0.1^\circ$ to 0.25° (Fixed head)
Magnetometers	0.5° to 3°
GPS	0.5° to 3°
Gyros	0.003 to 1 deg/h

ever, all gyros inherently drift over time, which degrades the attitude accuracy. A summary of the typical sensor performance ranges is shown in Table 1.

Attitude Determination

This section provides a review of the fundamentals of attitude determination. First, the mathematics of the underlying observation equations are summarized. Then, the popular q method is shown, which solves Wahba's classic problem. Finally, maximum likelihood estimation is shown.

Attitude sensors are generally modeled using the following (noiseless) observation model:

$$\mathbf{b} = \mathbf{A} \mathbf{r} \tag{1}$$

where A is the 3×3 attitude matrix, $\mathbf{r}$ is a 3×1 assumed-known *reference vector* of length one, and $\mathbf{b}$ is a 3×1 *body-observation vector*, also of length one. For attitude determination, right-handed orthogonal coordinates frames are always assumed. The attitude matrix maps coordinates in the reference frame to coordinates in the body frame. The body coordinate frame is usually defined through the initial vehicle design procedure. Many reference coordinate frames can be employed, which are usually dictated by the mission objectives. For example, if a stellar pointing requirement is given, then an inertially fixed frame is often used, where the motion of the stars is zero with respect to this frame. The star locations with respect to this frame are stored in a star catalogue.

The parameterization of choice for attitude determination is the quaternion, defined by $\mathbf{q} \equiv [q_1 \ q_2 \ q_3 \ q_4]^T = [q_1 \ q_2 \ q_3 \ q_4]^T$. The corresponding attitude matrix in Eq. (1) using the quaternion description of the attitude is given by $A(\mathbf{q}) = \mathcal{E}^T(\mathbf{q})\Psi(\mathbf{q})$, with

$$\mathcal{E}(\mathbf{q}) \equiv \begin{bmatrix} q_4 I_{3 \times 3} + [\mathbf{q}_{1:3} \times] \\ -\mathbf{q}_{1:3}^T \end{bmatrix} = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \quad (2a)$$

$$\Psi(\mathbf{q}) \equiv \begin{bmatrix} q_4 I_{3 \times 3} - [\mathbf{q}_{1:3} \times] \\ -\mathbf{q}_{1:3}^T \end{bmatrix} = \begin{bmatrix} q_4 & q_3 & -q_2 \\ -q_3 & q_4 & q_1 \\ q_2 & -q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \quad (2b)$$

where $I_{3 \times 3}$ is a 3×3 identity matrix and $[\mathbf{q}_{1:3} \times]$ is the standard “cross product matrix” Markley and Crassidis (2014). Successive rotations can be accomplished using quaternion multiplication: $A(\mathbf{q}')A(\mathbf{q}) = A(\mathbf{q}' \otimes \mathbf{q})$. The composition of the quaternions is bilinear, with

$$\mathbf{q}' \otimes \mathbf{q} = [\Psi(\mathbf{q}') : \mathbf{q}'] \mathbf{q} = [\mathcal{E}(\mathbf{q}) : \mathbf{q}] \mathbf{q}' \quad (3)$$

Note that here we adopt the convention of Lefferts et al. (1982) who multiply the quaternions in the same order as the attitude matrix

multiplication. Also, the inverse quaternion is defined by $\mathbf{q}^{-1} \equiv [-\mathbf{q}_{1:3}^T \ q_4]^T$. Since a four-dimensional vector is used to describe three dimensions, the quaternion components cannot be independent of each other. The quaternion satisfies a single constraint given by $\mathbf{q}^T \mathbf{q} = 1$.

There is no such thing as a “perfect measurement,” because every sensor contains errors, either in the form of random noise or systematic (nonrandom) errors. Oftentimes, systematic errors are reduced through pre-mission calibration procedures, although sensors may still require on-orbit calibration due to environmental effects or other sources of errors, such as aging of components. Standard attitude determination approaches assume that the systematic errors are zero, or negligible, and model the body measurements as

$$\tilde{\mathbf{b}} = A \mathbf{r} + \mathbf{v} \quad (4)$$

where $\mathbf{v}$ is a zero-mean Gaussian noise vector with assumed known covariance. For unit vector observations, the problem is even more complicated because a unit vector is never “observed.” For example, the body measurement from a star tracker is modeled by

$$\tilde{\mathbf{b}} \equiv \frac{1}{\sqrt{f^2 + (x + v_x)^2 + (y + v_y)^2}} \begin{bmatrix} -(x + v_x) \\ -(y + v_y) \\ f \end{bmatrix} \quad (5)$$

where f is the known focal length, x and y are the true focal plane observations, and v_x and v_y are zero-mean Gaussian noise processes with the same variance, σ^2 . The desired unit vector measurement model is given by Eq. (4). Clearly, using the measurement model in Eq. (5) does not produce a Gaussian process for $\mathbf{v}$, even though v_x and v_y are Gaussian. Fortunately, Shuster and Oh (1981) have shown that nearly all the probability of the errors is concentrated on a very small area about the direction of $A \mathbf{r}$, so the sphere containing that point can be approximated by a tangent plane, characterized by $\mathbf{v}^T A \mathbf{r} = 0$,

$E\{\mathbf{v}\} = \mathbf{0}$, and

$$E\{\mathbf{v}\mathbf{v}^T\} = \sigma^2 [I_{3 \times 3} - (A\mathbf{r})(A\mathbf{r})^T] \quad (6) \quad \text{with}$$

where $E\{\}$ denotes expectation. Note that this matrix is always singular. For large signal-to-noise ratios, which is valid for most spacecraft attitude sensors, the noise vector $\mathbf{v}$ is approximately Gaussian with covariance given by Eq.(6). This is highly important because Gaussian assumptions tend to greatly simplify the mathematical formulations used to develop the attitude determination algorithms.

Wahba's Problem

In 1965 Grace Wahba (1965) posed a problem that involves finding the proper orthogonal matrix A that minimizes the loss function

$$J(A) = \frac{1}{2} \sum_{i=1}^N a_i \|\tilde{\mathbf{b}}_i - A\mathbf{r}_i\|^2, \quad \text{subject to } A^T A = I_{3 \times 3} \quad (7)$$

where a_i are nonnegative weights and N is the total number of measurements. The loss function can be rewritten as $J(A) = \lambda_0 - \text{trace}(A B^T)$, where trace is the matrix trace operator and $\lambda_0 \equiv \sum_{i=1}^N a_i$. The matrix B is defined by

$$B \equiv \sum_{i=1}^N a_i \tilde{\mathbf{b}}_i \mathbf{r}_i^T = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \quad (8)$$

Clearly, $J(A)$ is minimized when $\text{trace}(A B^T)$ is maximized.

At first glance, the solution to this problem seems to follow a standard least-squares approach. However, the constraint $A^T A = I_{3 \times 3}$ makes the problem more difficult. Since the time this problem was first posed, many solutions have been proposed. A useful solution was given by Davenport who parameterized the attitude by a unit quaternion Markley and Crassidis (2014). Parameterizing the attitude matrix using a quaternion leads to maximizing

$$J'(\mathbf{q}) = \mathbf{q}^T K \mathbf{q}, \quad \text{subject to } \mathbf{q}^T \mathbf{q} = 1 \quad (9)$$

$$K \equiv \begin{bmatrix} S - \text{trace}(B) I_{3 \times 3} & \mathbf{z} \\ \mathbf{z}^T & \text{trace}(B) \end{bmatrix} \quad (10)$$

where $S \equiv B + B^T$ and $\mathbf{z} \equiv [B_{23} - B_{32} \ B_{31} - B_{13} \ B_{12} - B_{21}]^T = \sum_{i=1}^N a_i \tilde{\mathbf{b}}_i \times \mathbf{r}_i$. The optimal quaternion solution, denoted by $\hat{\mathbf{q}}$, is given by computing the normalized eigenvector of K with the largest eigenvalue, with

$$K \hat{\mathbf{q}} = \lambda_{\max} \hat{\mathbf{q}} \quad (11)$$

where $\lambda_{\max}$ is the maximum eigenvalue and $\hat{\mathbf{q}}$ is its corresponding unit eigenvector, which is also the optimal estimate of the quaternion.

Although Davenport's solution is useful, it was not practical for spacecraft applications in the 1970s and early 1980s because the eigenvector decomposition was too computationally burdensome. The problem was overcome by Shuster's QUaternion ESTimator (QUEST) algorithm Shuster and Oh (1981), which was first applied to the Magnetic Field Satellite mission in 1979, and to date has been the most widely used attitude determination algorithm.

Maximum Likelihood Estimation

Shuster noted that choosing the weights to be inverse variances, $a_i = \sigma_i^{-2}$ makes Wahba's problem a maximum likelihood estimation problem Shuster (1989). The "Fisher information matrix" (FIM) for a parameter vector $\mathbf{x}$ is given by

$$F = E \left\{ \frac{\partial}{\partial \mathbf{x} \partial \mathbf{x}^T} J(\mathbf{x}) \right\} \quad (12)$$

where $J(\mathbf{x})$ is the negative log-likelihood function, which is the loss function in this case (neglecting terms independent of A). Asymptotically, the FIM tends to the inverse of the estimate error covariance, denoted by P . This matrix is useful because it can be used to develop 3σ bounds between the true attitude and the estimated attitude. The FIM for the attitude is expressed in terms of incremental error angles,

$\delta\boldsymbol{\vartheta}$, defined according to

$$A \hat{A}^T = e^{-[\delta\boldsymbol{\vartheta} \times]} \approx (I_{3 \times 3} - [\delta\boldsymbol{\vartheta} \times]) \quad (13)$$

where A is treated as the *true* attitude here and $\hat{A}$ is its *estimate*. The parameter vector is now given by $\mathbf{x} = \delta\boldsymbol{\vartheta}$, and the error covariance is defined by $P = E\{\mathbf{x}\mathbf{x}^T\} - E\{\mathbf{x}\}E^T\{\mathbf{x}\}$, which is given by Shuster (1989)

$$P = \left(-\sum_{i=1}^N \sigma_i^{-2} [A \mathbf{r}_i \times]^2 \right)^{-1} \quad (14)$$

The attitude A is evaluated at its respective *true* value. In practice, though, $A \mathbf{r}_i$ is often replaced with the measurement $\hat{\mathbf{b}}_i$, which allows a calculation of the error covariance without computing an attitude.

Attitude Filtering

The workhorse of attitude filtering is the extended Kalman filter (EKF) Crassidis and Junkins (2012). The EKF uses a continuous-time model with discrete-time measurements to provide optimal state estimates. The state estimate model is propagated forward in time until a measurement occurs, given at time t_1 . Then a discrete-time state update occurs, which updates the final value of the propagated state $\hat{\mathbf{x}}_1^-$ to the new state $\hat{\mathbf{x}}_1^+$. Finally, this state is then used as the initial condition to propagate the state estimate model to time t_2 . The scheme continues forward in time, updating the state when a measurement occurs.

The continuous-time gyro sensors are typically modeled as

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{\beta} + \boldsymbol{\eta}_v \quad (15a)$$

$$\dot{\boldsymbol{\beta}} = \boldsymbol{\eta}_u \quad (15b)$$

where $\tilde{\boldsymbol{\omega}}$ is the measured rate, $\boldsymbol{\omega}$ is the true rate, $\boldsymbol{\beta}$ is the drift, and $\boldsymbol{\eta}_v$ and $\boldsymbol{\eta}_u$ are independent zero-mean Gaussian white-noise processes with

$$E\{\boldsymbol{\eta}_v(t)\boldsymbol{\eta}_v^T(\tau)\} = \sigma_v^2 \delta(t - \tau) I_{3 \times 3} \quad (16a)$$

$$E\{\boldsymbol{\eta}_u(t)\boldsymbol{\eta}_u^T(\tau)\} = \sigma_u^2 \delta(t - \tau) I_{3 \times 3} \quad (16b)$$

where $\delta(t - \tau)$ is the Dirac delta function. A more general gyro sensor model includes scale factors and misalignments, which can also be estimated in real time Markley and Crassidis (2014).

The EKF works on the premise of using a Taylor series expansion of the nonlinear model. Here the model contains the quaternion kinematics, as well as the drift in the gyro:

$$\dot{\mathbf{q}} = \frac{1}{2} \boldsymbol{\varepsilon}(\mathbf{q})(\tilde{\boldsymbol{\omega}} - \boldsymbol{\beta} - \boldsymbol{\eta}_v) \quad (17a)$$

$$\dot{\boldsymbol{\beta}} = \boldsymbol{\eta}_u \quad (17b)$$

Note that the gyro measurements are used in the kinematics model directly. Linearizing Eq. (17) using a straightforward Taylor series expansion has a number of problems. The most problematic issue is the fact that the quaternion normalization will be destroyed in the linearization. To overcome this problem, a “multiplicative EKF” (MEKF) Markley and Crassidis (2014) is used in the linearization process. The quaternion equivalent of Eq. (13) is given by

$$\delta\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \approx \begin{bmatrix} 0.5 \delta\boldsymbol{\vartheta} \\ 1 \end{bmatrix} \quad (18)$$

The MEKF uses $\delta\boldsymbol{\vartheta}$ as the “local parameterization,” while the quaternion is used as the “global parameterization.” This allows the state vector to be reduced by one state. The six-state MEKF attitude filter is shown in Table 2, where $F(t)$, $G(t)$, and $Q(t)$ are given by

$$\begin{aligned} F(t) &= \begin{bmatrix} -[\hat{\boldsymbol{\omega}}(t) \times] & -I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}, \\ G(t) &= \begin{bmatrix} -I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}, \\ Q(t) &= \begin{bmatrix} \sigma_v^2 I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \sigma_u^2 I_{3 \times 3} \end{bmatrix} \end{aligned} \quad (19)$$

Spacecraft Attitude Determination, Table 2
MEKF for attitude filtering

Initialize	$\hat{\mathbf{q}}(t_0) = \hat{\mathbf{q}}_0, \quad \hat{\boldsymbol{\beta}}(t_0) = \hat{\boldsymbol{\beta}}_0$ $P(t_0) = P_0$
Gain	$K_k = P_k^- H_k^T (\hat{\mathbf{x}}_k^-) [H_k (\hat{\mathbf{x}}_k^-) P_k^- H_k^T (\hat{\mathbf{x}}_k^-) + R_k]^{-1}$ $H_k (\hat{\mathbf{x}}_k^-) = \begin{bmatrix} [A(\hat{\mathbf{q}}^-) \mathbf{r}_1 \times] & 0_{3 \times 3} \\ \vdots & \vdots \\ [A(\hat{\mathbf{q}}^-) \mathbf{r}_N \times] & 0_{3 \times 3} \end{bmatrix}_{I_k}$
Update	$P_k^+ = [I - K_k H_k (\hat{\mathbf{x}}_k^-)] P_k^-$ $\Delta \hat{\mathbf{x}}_k^+ = K_k [\tilde{\mathbf{y}}_k - \mathbf{h}_k (\hat{\mathbf{x}}_k^-)]$ $\Delta \hat{\mathbf{x}}_k^+ \equiv \begin{bmatrix} \delta \hat{\boldsymbol{\theta}}_k^{+T} & \Delta \hat{\boldsymbol{\beta}}_k^{+T} \end{bmatrix}^T$ $\mathbf{h}_k (\hat{\mathbf{x}}_k^-) = \begin{bmatrix} A(\hat{\mathbf{q}}^-) \mathbf{r}_1 \\ A(\hat{\mathbf{q}}^-) \mathbf{r}_2 \\ \vdots \\ A(\hat{\mathbf{q}}^-) \mathbf{r}_N \end{bmatrix}_{I_k}$ $\hat{\mathbf{q}}^* = \hat{\mathbf{q}}_k^- + \frac{1}{2} \Xi (\hat{\mathbf{q}}_k^-) \delta \hat{\boldsymbol{\theta}}_k^+$ $\hat{\mathbf{q}}_k^+ = \hat{\mathbf{q}}^* / \ \hat{\mathbf{q}}^*\ $ $\hat{\boldsymbol{\beta}}_k^+ = \hat{\boldsymbol{\beta}}_k^- + \Delta \hat{\boldsymbol{\beta}}_k^+$
Propagation	$\hat{\boldsymbol{\omega}}(t) = \tilde{\boldsymbol{\omega}}(t) - \hat{\boldsymbol{\beta}}(t)$ $\dot{\hat{\mathbf{q}}}(t) = \frac{1}{2} \Xi (\hat{\mathbf{q}}(t)) \hat{\boldsymbol{\omega}}(t)$ $\dot{P}(t) = F(t) P(t) + P(t) F^T(t) + G(t) Q(t) G^T(t)$

where $0_{3 \times 3}$ is a 3×3 matrix of zeroes. The vector $\tilde{\mathbf{y}}$ is made up of the body measurements, and R_k is a block diagonal matrix with blocks given by $\sigma_i^2 I_{3 \times 3}$. Also, P denotes the error covariance from the filter now. Discrete-time propagation of the kinematics and error covariance is also possible Markley and Crassidis (2014).

Summary and Future Work

This chapter provided a fundamental algorithm, called the q method, to perform attitude determination from a collection of body and reference unit vectors. Any attitude determination algorithm requires at least two non-collinear vectors to determine a unique attitude. The error covariance derived from a maximum likelihood analysis is useful to quantify the expected errors between the true attitude and the estimated attitude. The most common attitude filter, called the multiplicative extended Kalman filter, was also introduced using vector observations along with

gyro measurements. This is a complementary filter in the sense that the gyros are used to filter the noisy measurements, while the measurements are used to calibrate the gyros in real time. For high-quality gyros, the filtered attitude solution is typically about an order of magnitude better than the one found from attitude determination algorithms. Future work should focus on deriving truly nonlinear filters that can guarantee stochastic stability from any initial condition. Other work entails deriving filters that are robust to possible non-Gaussian errors.

Cross-References

- [Estimation, Survey on](#)
- [Extended Kalman Filters](#)
- [Kalman Filters](#)
- [Nonlinear Filters](#)
- [Particle Filters](#)
- [Satellite Control](#)
- [Space Robotics](#)
- [State Estimation for Batch Processes](#)

Bibliography

- Crassidis JL, Junkins JL (2012) Optimal estimation of dynamic systems, 2nd edn. CRC Press, Boca Raton
- Crassidis JL, Markley FL, Cheng Y (2007) Survey of nonlinear attitude estimation methods. *J Guid Control Dyn* 30(1):12–28
- Lefferts EJ, Markley FL, Shuster MD (1982) Kalman filtering for spacecraft attitude estimation. *J Guid Control Dyn* 5(5):417–429. <https://doi.org/10.2514/3.56190>
- Markley FL, Crassidis JL (2014) Fundamentals of spacecraft attitude determination and control. Springer, New York
- Markley FL, Mortari D (2000) Quaternion attitude estimation using vector observations. *J Astronaut Sci* 48(2/3):359–380
- Shuster MD (1989) Maximum likelihood estimation of spacecraft attitude. *J Astronaut Sci* 37(1):79–88
- Shuster MD, Oh SD (1981) Three-axis attitude determination from vector observations. *J Guid Control* 4(1):70–77. <https://doi.org/10.2514/3.19717>
- Wahba G (1965) A least-squares estimate of satellite attitude. *SIAM Rev* 7(3):409. <https://doi.org/10.1137/1007077>

Spatial Description of Biochemical Networks

Pablo A. Iglesias

Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD, USA

Abstract

Many biological behaviors require that biochemical species be distributed spatially throughout the cell or across a number of cells. To explain these situations accurately requires a spatial description of the underlying network. At the continuum level, this is usually done using reaction-diffusion equations. Here we demonstrate how this class of models arises. We also show how the framework is used in two popular models proposed to explain spatial patterns during development.

Keywords

Diffusion · Morphogen gradient · Pattern formation · Reaction-diffusion · Turing instability

Introduction

Cells are complex environments consisting of spatially segregated entities, including the nucleus and various other organelles. Even within these compartments, the concentrations of various biochemical species are not homogeneous, but can vary significantly. The proper localization of proteins and other biochemical species to their respective sites is important for proper cell function. This can be because the spatial distribution of signaling molecules itself confers information, such as when a cell needs to respond to a spatially graded cue to guide its motion (Iglesias and Devreotes 2008) or growth pattern (Lander 2013). Alternatively, information that is obtained in one part of the cell must be transmitted to another part of the cell, as when receptor-ligand binding at the cell surface leads to transcriptional responses in the nucleus. Frequently, describing the action of a biological network accurately requires not only that one account for the chemical interactions between the different components but that the spatial distribution of the signaling molecules also be considered.

Accounting for Spatial Distribution in Models

Mathematical models of biological networks usually assume that reactions take place in well-stirred vessels in which the concentrations of the interacting species are spatially homogeneous and hence need not be accounted for explicitly. These systems also assume that the volume is constant. When the spatial location of molecules in cells is important, the concentration of species changes in both time and space.

Compartmental Models

One way to account for spatial distribution of signaling components is through compartmental models. As the name suggests, in these models the cell is divided into different regions that are segregated by membranes. Within each compartment, the concentration of the network species

is assumed to be spatially homogeneous. The membranes in these models can be assumed to be either permeable or impermeable. In permeable membranes, information passes through small openings, such as ion channels or nuclear pores, which allow molecules to move from one side of the membrane to the other. With impermeable membranes, information must be transduced by transmembrane signaling elements, such as cell-surface receptors, that bind to a signaling molecule in one side of the membrane and release a secondary effector on the other side. Note that in this case, the membrane itself acts as a third compartment.

Compartmental models offer simplicity, since the reactions that happen in a single region obey the same reaction kinetics usually assumed in spatially homogeneous models. Even when the reactions involve more than one compartment, as in ligand-receptor binding, this can still be described by the usual reaction dynamics. Care must be taken, however, to account properly for the different effects on the respective concentrations as molecules move from one compartment to another. In models of spatially homogeneous systems, there is little practical difference between writing the ordinary-differential equations in terms of molecule numbers or concentrations, since the two are proportional to each other according to the volume, which is constant. In a compartmental model, if the molecule moves from one compartment to another, there is conservation of molecule numbers, but not concentrations. For example, if a species is found in two compartments with volumes V_1 and V_2 and transfer rates k_{12} and $k_{21} \text{ s}^{-1}$, then the differential equations describing transport between compartments can be expressed in terms of numbers (n_1 and n_2) as follows:

$$\begin{aligned}\frac{dn_1}{dt} &= -k_{12}n_1 + k_{21}n_2 \\ \frac{dn_2}{dt} &= +k_{12}n_1 - k_{21}n_2.\end{aligned}$$

Dividing by the respective volumes ($C_1 = n_1/V_1$ and $C_2 = n_2/V_2$), we obtain equations for the concentrations

$$\begin{aligned}\frac{dC_1}{dt} &= -k_{12}C_1 + k_{21}\left(\frac{V_2}{V_1}\right)C_2 \\ \frac{dC_2}{dt} &= +k_{12}\left(\frac{V_1}{V_2}\right)C_1 - k_{21}C_2.\end{aligned}$$

In the former case, the two equations add to zero, indicating that $n_1(t) + n_2(t) = \text{constant}$. In the latter, if $V_1 \neq V_2$, then $C_1(t) + C_2(t)$ varies over time as molecules move from one compartment to the other.

Diffusion and Advection

If the distribution of molecules inside any single compartment is spatially heterogeneous, then models must account for this spatial distribution. At the continuum level, this is done using reaction-diffusion equations. The basic assumption is a conservation principle expressed as a continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla j = f,$$

which relates the changes in the density (ρ) of a conserved quantity (in our case, the concentration of a species: $\rho = C$) to the flux j and any net production f . In biological networks, the latter represents the net effect of all the reactions that affect the concentration of the species including binding, unbinding, production, degradation, post-translational modifications, etc.

In biological models, the flux term usually comes from one of two sources: diffusion or advection. According to Fick's law, diffusive flux is proportional to the negative gradient of the concentration of the species as particles move from regions of high concentration to regions of low concentration. The coefficient of proportionality is the diffusion coefficient, D :

$$j_{\text{diff}} = -D\nabla C.$$

Fick's law describes thermally driven Brownian motion of molecules at the continuum level. If the species is embedded in a moving field, then the flux is proportional to the velocity of the underlying fluid. In this case, we have advective flow:

$$j_{\text{adv}} = vC.$$

In biological systems, advection can arise because of the movement of the cytoplasm, but it can also represent directed transport of molecules, such as the movement of cargo along filaments by processive motors. In general, molecules exhibit both diffusive and advective motion: $j = j_{\text{diff}} + j_{\text{adv}}$, leading to

$$\frac{\partial C}{\partial t} + \nabla(-D\nabla C + vC) = f,$$

which, under the assumption that the diffusion coefficient and the transport velocity are independent of spatial location, leads to the reaction-diffusion-advection equation:

$$\frac{\partial C}{\partial t} = D\nabla^2 C - v\nabla C + f.$$

Being a second-order partial differential, the solution requires an initial condition and two boundary conditions. Common choices for the latter include periodic (e.g., in models of closed boundaries) or no-flux (to describe the impermeability of membranes) assumptions.

Measuring Diffusion Coefficients

Invariably, solving the reaction-diffusion equation requires knowledge of the diffusion coefficient of the molecule. Experimentally, this can be done in a number of ways. In fluorescence recovery after photobleaching (FRAP), a laser is used to photobleach normally fluorescent molecules in a specific area of the cell. As these “dark” molecules are replaced by fluorescent molecules from non-bleached areas, the fluorescent intensity of the bleached area recovers. Higher diffusion leads to faster recovery. The time to half recovery, $\tau_{1/2}$, can be used to estimate D . If recovery occurs by lateral diffusion, then

$$D = \frac{r_0^2 \gamma}{4\tau_{1/2}}$$

where r_0 is the $1/e^2$ radius of the Gaussian profile laser beam and γ is a parameter that depends on the extent of photobleaching, which ranges from 1 to 1.2 (Chen et al. 2006).

These days, it is increasingly common to measure lateral diffusion coefficients by observing the trajectory of single molecules. A molecule with diffusion coefficient D undergoing Brownian motion in a two-dimensional environment is expected to have mean-square displacement (MSD) equal to

$$\langle r^2 \rangle = 4Dt.$$

Thus, the coefficient D can be obtained by measuring how the MSD changes as a function of the time interval t . This method can also show if the molecule is undergoing advection in which case

$$\langle r^2 \rangle = 4Dt + v^2 t^2.$$

This super-diffusive behavior can be seen in the concave nature of the plot of $\langle r^2 \rangle$ against t . This plot will also reveal barriers to diffusion. For example, if the molecule is confined to move in a circular region of radius a , then, as t increases, $\langle r^2 \rangle$ cannot exceed a^2 .

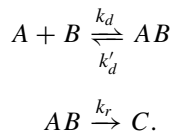
Both these methods work best for molecules diffusing on a membrane. For molecules diffusing in the cytoplasm, the three-dimensional imaging required is considerably more difficult, particularly since the diffusion of particles in the cytoplasm ($D \sim 1\text{--}10 \mu\text{m}^2 \text{s}^{-1}$) is usually orders of magnitude greater than for membrane-bound proteins ($D \sim 0.01\text{--}0.1 \mu\text{m}^2 \text{s}^{-1}$). In this case, an analytical expression can be used to estimate the diffusion coefficient. The diffusion coefficient of a spherical particle of radius r moving in a low Reynolds number liquid with viscosity η is given by the Stokes-Einstein equation:

$$D = \frac{k_B T}{6\pi\eta r}.$$

The exact viscosity of the cell is unknown, but estimates that η is approximately five times that of water lead to diffusion coefficients of cytoplasmic proteins that match those measured using FRAP.

Diffusion-Limited Reaction Rates

Even in compartments that are considered well stirred, the diffusion of molecules is necessary for reactions to take place. In particular, before two molecules can react, they must come together. To see how diffusion influences this, suppose that spherical molecules of species A and B with radii r_A and r_B , respectively, come together to form a complex AB at a rate k_d . This rate represents the likelihood that molecules of A and B collide at random and hence will depend on the diffusion properties of the two species. The molecules in this complex can dissociate at rate k'_d or can be converted to species C at rate k_r . Thus, the overall reaction involves two steps:



Assuming that the system is at quasi-steady-state, that is, the concentration of AB is constant, the effective rate of production C is given by

$$k_{\text{eff}} = \frac{k_d k_r}{k'_d + k_r}.$$

There are two regions of operation. If $k'_d \gg k_r$, then $k_{\text{eff}} \approx k_r (k_d/k'_d)$. In this case production is said to be reaction limited. If $k'_d \ll k_r$, then $k_{\text{eff}} \approx k_d$ and production is diffusion limited. In this case, it is possible to find k_d as a function of the species' diffusion coefficients.

Assume that species A is stationary, in which case the effective diffusion is the sum of the two diffusion coefficients: $D = D_A + D_B$. The concentration of species B depends on the distance away from molecules of A . Because we assume that the reaction rate is fast, at the point of contact ($r^* = r_A + r_B$) the concentration is zero since any molecules of AB are quickly converted to C . At the other extreme, as $r \rightarrow \infty$, the concentration approaches the bulk concentration B_0 . According to Fick's law, this concentration gradient causes a flux density given by $j = -D(\partial B/\partial r)$. The total flux into a sphere of radius r is then

$$J = 4\pi r^2 j = -4\pi D r^2 \frac{\partial B}{\partial r},$$

which, at steady state, is constant. Solving this equation for $B(r)$ using the two boundary equations leads to a flux

$$J = -4\pi D B_0 r^*,$$

from which we have that

$$k_d = 4\pi D r^*.$$

A typical value for k_d , using the Einstein-Stokes formula, is

$$\begin{aligned} 4\pi \left(2 \times \frac{k_B T}{6\pi \eta (r^*/2)} \right) r^* &= \frac{8k_B T}{3\eta} \\ &\approx 10^3 \mu\text{m}^{-1} \text{s}^{-1}. \end{aligned}$$

Spatial Patterns

The effect of spatial heterogeneities has been of long interest to developmental biologists, who study how spatial patterns arise. Two distinct models have been proposed to explain how this patterning can arise. Here we introduce these models and discuss their relative merits. Though usually seen as competing models, there is recent evidence suggesting that both models may play complementary roles during development (Reth et al. 2012).

Morphogen Gradients

A morphogen is a diffusible molecule that is produced or secreted at one end of an organism. Diffusion away from the localized source forms a concentration gradient along the spatial dimension. Morphogens are used to control gene expression of cells lying along this spatial domain. Thus, a morphogen gradient gives rise to spatially dependent expression profiles that can account for spatial developmental patterns (Rogers and Schier 2011).

The mathematics behind the formation of a morphogen gradient are relatively straightforward.

The concentration of the morphogen is denoted by $C(x, t)$. There is a constant flux (j_0) at one end ($x = 0$) of a finite one-dimensional domain of length L , but the morphogen cannot exit at the other end. The species diffuses inside the domain and also decays at a rate proportional to its concentration ($f = -kC$). Thus, the concentration is governed by the reaction-diffusion equation:

$$\frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial x^2} - kC,$$

with boundary conditions: $D \frac{\partial C}{\partial x} = -j_0$ at $x = 0$, and $D \frac{\partial C}{\partial x} = 0$ at $x = L$. We focus on the steady state:

$$\frac{d^2 \bar{C}}{dx^2} = \frac{k}{D} \bar{C},$$

so that the initial condition is not important. In this case, the distribution of the species is given by

$$\bar{C}(x) = \frac{\lambda j_0}{D} \frac{\cosh([L - x]/\lambda)}{\sinh(L/\lambda)}.$$

Thus, the shape of the gradient is roughly exponential with parameter $\lambda = \sqrt{D/k}$, known as the dispersion, which specifies the average distance that molecules diffuse into the domain before they are degraded or inactivated. Equally important in determining the gradient, however, is the spatial dimension (L) relative to the dispersion, $\Phi = L/\lambda$, a ratio known as the Thiele modulus. If $\Phi \ll 1$, then the concentration will be approximately homogeneous. Alternatively, $\Phi \gg 1$ leads to a sharp transition close to the boundary where there is flux and a relatively flat concentration thereafter.

Though morphogen gradients are commonly used to describe signaling during development, where the gradient can extend across a number of cells, the mathematics described above are equally suitable for describing concentration gradients of intracellular proteins. In this case, the dimension of the cell has a significant effect on the shape of the gradient (Meyers et al. 2006).

As discussed above, morphogen gradients are established in an open-loop mode. As such, the actual concentration experienced at a point

downstream of the source of the morphogen will vary depending on a number of parameters, including the flux j_0 and the rate of degradation k . Moreover, because the concentration of the morphogen decreases as the distance from the source grows, the relative stochastic fluctuations will increase. How to manage this uncertainty is an active area of research (Rogers and Schier 2011; Lander 2013).

Diffusion-Driven Instabilities

In 1952, Alan Turing proposed a model of how patterns could arise in biological systems (Turing 1952). His interest was in explaining how an embryo, initially spherical, could give rise to a highly asymmetric organism. He posited that the breaking of symmetry could be a result of the change in the stability of the homogeneous state of the network which would amplify small fluctuations inherent in the initial symmetry. Turing sought to explain how these instabilities could arise using only reaction-diffusion systems.

To illustrate how diffusion-driven instabilities can arise, we work with a single two-species linear reaction network:

$$\frac{\partial}{\partial t} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = A \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} + \frac{\partial^2}{\partial x^2} D \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

where $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ specifies the reaction terms and the diagonal matrix $D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$ the diffusion coefficients.

We assume that, in the absence of diffusion, the system is stable, so that $\det(A) > 0$ and $\text{trace}(A) < 0$. When considering diffusion in a one-dimensional environment of length L , we must consider the spatial modes, which are of the form $\exp(ikx)$. In this case, stability of the system requires that $\text{trace}(A - q^2 D) < 0$ and $\det(A - q^2 D) > 0$. The former is always true, since $\text{trace}(A - q^2 D) = \text{trace}(A) - q^2(D_1 + D_2) < \text{trace}(A) < 0$. However, the condition on the determinant can fail since

$$\det(A - q^2 D) = D_1 D_2 q^4 - q^2(a_{22} D_1 + a_{11} D_2) + \det(A). \quad (1)$$

Since $\det(A) > 0$, diffusion-driven instabilities can only occur if the term $a_{22}D_1 + a_{11}D_2 > 0$, by which it follows that at least one of a_{11} or a_{22} must be positive. Since $\text{trace}A < 0$, it follows that the diagonal terms must have opposite sign. Usually, it is assumed that $a_{11} > 0$ and that $a_{22} < 0$. Since $\det(A) > 0$, it follows that a_{12} and a_{21} must also have opposite sign.

These requirements in the sign pattern of the two molecules lead to one of two classes of systems. In the first class, known as activator/inhibitor systems, the activator (assume species 1) is autocatalytic ($a_{11} > 0$) and also stimulates the inhibitor ($a_{21} > 0$), which negatively regulates the activator ($a_{12} < 0$). In the other class, known as substrate-depletion systems, a product (species 1) is autocatalytic ($a_{11} > 0$), but in its production consumes ($a_{21} < 0$) the substrate (species 2) whose presence is needed for formation of the product ($a_{12} > 0$). Note that both systems involve an autocatalytic positive feedback loop ($a_{11} > 0$), as well as a negative feedback loop involving both species ($a_{12}a_{21} < 0$).

The stability condition also imposes a necessary condition on the dispersion of the two species, ($\lambda_i = \sqrt{D_i/|a_{ii}|}$), since

$$a_{22}D_1 + a_{11}D_2 > 0 \implies -\lambda_1^2 + \lambda_2^2 > 0$$

Thus, the species providing the negative feedback (inhibitor or substrate) must have higher dispersion ($\lambda_2 > \lambda_1$). This requirement is usually referred to as local activation and long-range inhibition.

These conditions are necessary, but not sufficient. They ensure that the parabola defined by Eq. 1 has real roots. However, when diffusion takes place in finite domains, the parameter q can only take discrete values $q = 2\pi n/L$ for integers n . Thus, for a spatial mode to be unstable, it must be that $\det(A - q^2D) < 0$ at specific values of q corresponding to integers n . If the dimension of the domain is changing, as would be expected in a growing domain, the parameter q^2 will decrease over time suggesting that higher modes may lose stability. Thus, the nature of the pattern may evolve over time.

Over the years, Turing's framework has been a popular model among theoretical biologists and has been used to explain countless patterns seen in biological systems. It has not had the same level of acceptance among biologists, likely because of the difficulty of mapping a complex biological system involving numerous interacting species into the simple nature of the theoretical model (Kondo and Miura 2010).

Summary and Future Directions

Spatial aspects of biochemical signaling are increasingly playing a role in the study of cellular signaling systems. Part of this interest is the desire to explain spatial patterns seen in sub-cellular localizations observed through live cell imaging using fluorescently tagged proteins. The ever-increasing computational power available for simulations is also facilitating this progress. Specially built spatial simulation software, such as the Virtual Cell, is freely available and tailor-made for biological simulations enabling simulation of spatially varying reaction networks in cells of varying size and shape (Cowan et al. 2012).

Of course, cell shapes are not static, but evolve in large part due to the effect of the underlying biochemical system. This requires simulation environments that solve reaction-diffusion systems in changing morphologies. This has received considerable interest in modeling cell motility (Holmes and Edelstein-Keshet 2013).

Another aspect of spatial models that is only now being addressed is the role of mechanics in driving spatially dependent models. For example, it has recently been shown that the interaction between biochemistry and biomechanics can itself drive Turing-like instabilities (Goehring and Grill 2013).

Finally, we note that our discussion of spatially heterogeneous signaling has been based on continuum models. As with spatially invariant systems, this approach is only valid if the number of molecules is sufficiently large that the stochastic nature of the chemical reactions can be

ignored. In fact, spatial heterogeneities may lead to localized spots requiring a stochastic approach, even though the molecule numbers are such that a continuum approach would be acceptable if the cell were spatially homogeneous. The analysis of stochastic interactions in these systems is still much in its infancy and is likely to be an increasingly important area of research (Mahmutovic et al. 2012).

Cross-References

- [Deterministic Description of Biochemical Networks](#)
- [Monotone Systems in Biology](#)
- [Robustness Analysis of Biological Models](#)
- [Stochastic Description of Biochemical Networks](#)

Bibliography

- Chen Y, Lagerholm BC, Yang B, Jacobson K (2006) Methods to measure the lateral diffusion of membrane lipids and proteins. *Methods* 39:147–153
- Cowan AE, Moraru II, Schaff JC, Slepchenko BM, Loew LM (2012) Spatial modeling of cell signaling networks. *Methods Cell Biol* 110:195–221
- Goehring NW, Grill SW (2013) Cell polarity: mechanochemical patterning. *Trends Cell Biol* 23: 72–80
- Holmes WR, Edelstein-Keshet L (2013) A comparison of computational models for eukaryotic cell shape and motility. *PLoS Comput Biol* 8:e1002793; 2012
- Iglesias PA, Devreotes PN (2008) Navigating through models of chemotaxis. *Curr Opin Cell Biol* 20: 35–40
- Kondo S, Miura T (2010) Reaction-diffusion model as a framework for understanding biological pattern formation. *Science* 329:1616–1620
- Lander AD (2013) How cells know where they are. *Science* 339:923–927
- Mahmutovic A, Fange D, Berg OG, Elf J (2012) Lost in presumption: stochastic reactions in spatial models. *Nat Methods* 9:1163–1166
- Meyers J, Craig J, Odde DJ (2006) Potential for control of signaling pathways via cell size and shape. *Curr Biol* 16:1685–1693
- Rogers KW, Schier AF (2011) Morphogen gradients: from generation to interpretation. *Annu Rev Cell Dev Biol* 27:377–407
- Sheth R, Marcon L, Bastida MF, Junco M, Quintana L, Dahn R, Kmita M, Sharpe J, Ros MA (2012) Hox genes regulate digit patterning by controlling the wavelength of a turing-type mechanism. *Science* 338: 1476–1480
- Turing AM (1952) The chemical basis of morphogenesis. *Philos Trans R Soc Lond* 237:37–72

Spectral Factorization

Michael Sebek

Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague 6, Czech Republic

Abstract

For more than half a century, spectral factorization is encountered in various fields of science and engineering. It is a useful tool in robust and optimal control and filtering and many other areas. It is also a nice control-theoretical concept closely related to Riccati equation. As a quadratic equation in polynomials, it is a challenging algebraic task.

Keywords

Controller design · H_2 -optimal control · H_∞ -optimal control · J-spectral factorization · Linear systems · Polynomial · Polynomial equation · Polynomial matrix · Polynomial methods · Spectral factorization

Polynomial Spectral Factorization

As a mathematical tool, the spectral factorization was invented by Wiener in 1940s to find a frequency domain solution of optimal filtering problems. Since then, this technique has turned up numberless applications in system, network and communication theory, robust and optimal control, filtration, prediction and state reconstruction. Spectral factorization of scalar polynomials is naturally encountered in the area of single-input single-output systems.

In the context of continuous-time problems, real polynomials in a single complex variable s are typically used. For such a polynomial $p(s)$, its *adjoint* $p^*(s)$ is defined by

$$p^*(s) = p(-s), \quad (1)$$

which results in flipping all roots across the imaginary axis. If the polynomial is *symmetric*, then $p^*(s) = p(s)$ and its roots are symmetrically placed about the imaginary axis.

The *symmetric spectral factorization* problem is now formulated as follows: Given a symmetric polynomial $b(s)$,

$$b^*(s) = b(s), \quad (2)$$

that is also positive on the imaginary axis

$$b(i\omega) > 0 \text{ for all real } \omega, \quad (3)$$

find a real polynomial $x(s)$, which satisfies

$$x(s)x^*(s) = b(s) \quad (4)$$

as well as

$$x(s) \neq 0, \quad \text{Res} \geq 0. \quad (5)$$

Such an $x(s)$ is then called a *spectral factor* of $b(s)$. By (5), the spectral factor is a stable polynomial in the continuous-time (Hurwitz) sense.

Obviously, (4) is a quadratic equation in polynomials and its stable solution is the desired spectral factor.

Example Given

$$b(s) = 4 + s^4 = (1 + j + s)(1 - j + s) \\ (1 + j - s)(1 - j - s),$$

(4) results in the spectral factor

$$x(s) = 2 + 2s + s^2 = (1 + j + s)(1 - j + s).$$

When the right-hand side polynomial $b(s)$ has some imaginary-axis roots, the problem

formulated strictly as above becomes unsolvable since (3) does not hold and hence (5) cannot be fulfilled. A more relaxed formulation may then find its use requiring only $b(i\omega) \geq 0$ instead of (3) and $x(s) \neq 0$ only for $\text{Res} > 0$ instead of (5). Clearly, the imaginary-axis roots of $b(s)$ must then appear in $x(s)$ and $x^*(s)$ as well.

In the realm of discrete-time problems, one usually encounters *two-sided polynomials*, which are polynomial-like objects (In fact, one can stay with standard one-sided polynomials (either in nonnegative or in nonpositive powers only), if every adjoint $p^*(z)$ is multiplied by proper power of z to create a one-sided polynomial $\bar{p}(z) = p^*(z)z^n$.) with positive and/or negative powers of a complex variable z , such as, for example, $p(z) = z^{-1} + 1 + 2z$. Here, the *adjoint* $p^*(z)$ stands simply for

$$p^*(z) = p(z^{-1}) \quad (6)$$

and the operation results in flipping all roots across the unit circle. If the two-sided polynomial is *symmetric*, then $p^*(z) = p(z)$ and its roots are symmetrically placed about the unit circle.

In its discrete-time version, the spectral factorization problem is stated as follows: Given a symmetric two-sided polynomial $b(z)$ that meets the conditions of symmetry

$$b^*(z) = b(z) \quad (7)$$

and positiveness (here on the unit circle)

$$b(e^{j\omega}) > 0 \text{ real } \omega, \quad -\pi < \omega \leq \pi, \quad (8)$$

find a real polynomial $x(z)$ in nonnegative powers of z to satisfy

$$x(z)x^*(z) = b(z) \quad (9)$$

and

$$x(z) \neq 0, \quad |z| \geq 1. \quad (10)$$

By (10), the spectral factor is a stable polynomial in the discrete-time (Schur) sense.

Example For

$$\begin{aligned}
 b(z) &= 2z^{-2} + 6z^{-1} + 9 + 6z + 2z^2 \\
 &= 2z^{-2}(z + 0.5 + 0.5j)(z + 0.5 - 0.5j) \\
 &\quad (z + 1 + j)(z + 1 - j) \\
 &= 4(z + 0.5 + 0.5j)(z + 0.5 - 0.5j) \\
 &\quad \times (z^{-1} + 0.5 + 0.5j) \\
 &\quad (z^{-1} + 0.5 - 0.5j)
 \end{aligned}$$

(9) yields

$$\begin{aligned}
 x(z) &= 1 + 2z + 2z^2 = 2(z + 0.5 + 0.5j) \\
 &\quad (z + 0.5 - 0.5j)
 \end{aligned}$$

as the desired spectral factor.

When the right-hand side $b(z)$ possesses some roots on the unit circle, this problem turns out to be unsolvable as (8) fails. If necessary, a less restrictive formulation can then be applied replacing (8) by $b(e^{i\omega}) \geq 0$ and with $x(z) \neq 0$ only for $|z| > 1$ instead of (10). Clearly, the unit-circle roots of $b(z)$ must then appear both in $x(z)$ and $x^*(z)$.

When formulated as above, the spectral factorization problem is always solvable and its solution is unique up to the change of sign (if x is a solution, so is $-x$ and no other solutions exist).

Polynomial Matrix Spectral Factorization

Matrix version of the problem has been encountered since 1960s. In the world of continuous-time problems, real polynomial matrices in a single complex variable s are used. For such a real polynomial matrix $P(s)$, its *adjoint* $P^*(s)$ is defined as

$$P^*(s) = P^T(-s). \quad (11)$$

A polynomial matrix $P(s)$ is *symmetric* or, more precisely, *para-Hermitian*, if $P^*(s) = P(s)$. Needless to say, only square polynomial matrices can be symmetric.

The matrix spectral factorization problem is defined as follows: Given a symmetric polynomial matrix $B(s)$,

$$B^*(s) = B(s), \quad (12)$$

that is also positive definite on the imaginary axis

$$B(i\omega) > 0 \quad \text{for all real } \omega, \quad (13)$$

find a square real polynomial matrix $X(s)$, which satisfies

$$X(s)X^*(s) = B(s) \quad (14)$$

and has no zeros in the closed right half plain $\text{Re } s \geq 0$. Such an $X(s)$ is then called a *left spectral factor* of $B(s)$. A *right spectral factor* $Y(s)$ is defined similarly by replacing (14) with

$$Y^*(s)Y(s) = B(s). \quad (15)$$

Example For a symmetric matrix

$$B(s) = \begin{bmatrix} 2 - s^2 & -2 - s \\ -2 + s & 4 - s^2 \end{bmatrix},$$

we have

$$X(s) = \begin{bmatrix} 1.4 + s & -0.2 \\ -1.2 & 1.6 + s \end{bmatrix}$$

as a left spectral factor and

$$Y(s) = \begin{bmatrix} 1 + s & 0 \\ -1 & 2 + s \end{bmatrix}$$

as a right one.

As in the scalar case, less restrictive definitions are sometimes used where the given right-hand side matrix $B(s)$ is only nonnegative definite on the imaginary axis and so the spectral factor is free of zeros in the open right half plain $\text{Re } s > 0$ only.

In the kingdom of discrete-time, *two-sided real polynomial* matrices $P(z)$ are used having in general entries with both positive and negative powers of the complex variable z . For such a matrix, its *adjoint* $P^*(z)$ is defined by

$$P^*(z) = P^T(z^{-1}). \quad (16)$$

Clearly, if $P(z)$ has only nonnegative powers of z , then $P^*(z)$ has only nonpositive powers of z and

vice versa. A square two-sided polynomial matrix $P(z)$ is (*para-Hermitian*) *symmetric* if $P^*(z) = P(z)$.

Here is the discrete-time version of matrix spectral factorization problem. Given a two-sided polynomial matrix $B(z)$ that is symmetric

$$B^*(z) = B(z) \quad (17)$$

and positively definite on the unit circle

$$B(e^{i\omega}) > 0 \quad \text{real } \omega, \quad -\pi < \omega \leq \pi, \quad (18)$$

find a real polynomial matrix $X(z)$ in nonnegative powers of z such that

$$X(z)X^*(z) = B(z) \quad (19)$$

and has no zeros on and outside of the unit circle. Such an $X(z)$ is then called a *left spectral factor* of $B(z)$. A *right* (The right and the left spectral factor are sometimes called the *factor* and the *cofactor*, respectively, but the terminology is not set at all.) *spectral factor* $Y(z)$ is defined similarly by replacing (19) with

$$Y^*(z)Y(z) = B(z) \quad (20)$$

Example A symmetric two-sided polynomial matrix

$$B(z) = \begin{bmatrix} -2z^{-1} + 5 - 2z & 2z^{-1} - 1 \\ -1 + 2z & 2z^{-1} + 6 + 2z \end{bmatrix}$$

has a left spectral factor

$$X(z) \cong \begin{bmatrix} -1.1 + 1.9z & 0.55 \\ -0.8z & 0.95 + 2.1z \end{bmatrix}$$

and a right spectral factor

$$Y(z) = \begin{bmatrix} 2z - 1 & 1 \\ 0 & 1 + 2z \end{bmatrix}.$$

As before, less restrictive formulations are sometimes encountered where the given symmetric $B(z)$ is only nonnegatively definite on the unit

circle and so the spectral factor must have no zeros only outside of the unit circle.

When formulated as above, the matrix spectral factorization problem is always solvable. The spectral factors are unique up to an orthogonal matrix multiple. That is, if X and X' are two left spectral factors of B , then

$$X' = UX \quad (21)$$

where U is a constant orthogonal matrix $UU^T = I$, while if Y and Y' are two right spectral factors of B , then

$$Y' = YV \quad (22)$$

where V is a constant orthogonal matrix $V^T V = I$.

J-Spectral Factorization

In robust control, game theory and several other fields, the symmetric right-hand side in the matrix spectral factorization may have a general signature. With such a right-hand side, standard (positive or nonnegative definite) factorization becomes impossible. Here, a similar yet different *J*-spectral factorization takes its role.

In the context of continuous-time problems, the *J-spectral factorization problem* is formulated as follows. Given a symmetric polynomial matrix $B(s)$,

$$B^*(s) = B(s), \quad (23)$$

find a square real polynomial matrix $X(s)$, which satisfies

$$X(s)JX^*(s) = B(s), \quad (24)$$

where $X(s)$ has no zeros in the open right half plain $\text{Re } s > 0$ and J is a *signature matrix* of the form

$$J = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & -I_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (25)$$

with I_1 and I_2 unit matrices of not necessarily the same dimensions. The bottom right block of zeros is often missing, yet it is considered here for generality. Such an $X(s)$ is called a *left J-spectral factor* of $B(s)$. A *right J-spectral factor*

is defined by

$$Y^*(s)JY(s) = B(s) \quad (26)$$

instead of (24). For discrete-time problems, the J -spectral factorization is defined analogously.

The J -spectral factorization problem is quite general having standard (either positive or nonnegative) spectral factorization as a particular case. No necessary and sufficient existence conditions appear to be known for J -spectral factorization. A sufficient condition by Jakubovič (1970) states that the problem is solvable if the multiplicity of the zeros on the imaginary axis of each of the invariant polynomials of the right-hand side matrix is even. In particular, this condition is satisfied whenever $\det B(s)$ has no zeros on the imaginary axis. In turn, the condition is violated if any of the invariant factors is not factorable by itself. An example of a nonfactorizable polynomial is $1 + s^2$.

The J -spectral factors are unique up to a J -orthogonal matrix multiple. That is, if X and X' are two left J -spectral factors of B , then

$$X' = UX, \quad (27)$$

where U is a J -orthogonal matrix $UJU^T = J$, while if Y and Y' are two right J -spectral factors of B , then

$$Y' = YV, \quad (28)$$

where V is a J -orthogonal matrix $V^TJV = J$.

Example For

$$B(s) = \begin{bmatrix} 0 & 1-s \\ 1+s & 2-s^2 \end{bmatrix}$$

the signature matrix reads

$$J = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

and the right J -spectral factor is

$$Y(s) = \begin{bmatrix} 1+s & \frac{3-s^2}{2} \\ 1+s & \frac{1-s^2}{2} \end{bmatrix}$$

Nonsymmetric Spectral Factorization

Spectral factorization can also be non-symmetric. For a scalar polynomial p (either in s or in z), this means to factor it directly as

$$p = p^+ p^- \quad (29)$$

where p^+ is a stable factor of p (having all its roots either in the open left half plane or inside of the unit disc, depending on the variable type) while p^- is the “remaining” that is unstable factor. Eventual roots of p at the stability boundary either associate to p^+ or to p^- , depending on the application problem at hand.

For a matrix polynomial P , the non-symmetric factorization is naturally twofold: Either

$$P = P^+ P^- \quad (30)$$

or

$$P = P^- P^+. \quad (31)$$

For scalar polynomials, symmetric and non-symmetric spectral factors are closely related. Given p and having computed a symmetric factor x for pp^* as in (4) or (9) to get

$$x^*x = p^*p \quad (32)$$

Then

$$p^+ = \gcd(p, x) \quad \text{and} \quad p^- = \gcd(p, x^*) \quad (33)$$

where $\gcd$ stands for a greatest common divisor. In reverse,

$$x = p^+ (p^-)^* \quad \text{and} \quad x^* = p^- (p^+)^*. \quad (34)$$

Unfortunately, no such relations exist for the matrix case.

Example For example,

$$p(s) = 1 - s^2$$

factorizes into

$$p^+(s) = 1 + s, \quad p^-(s) = 1 - s$$

while for

$$P(s) = \begin{bmatrix} 1 + s & 0 \\ 1 + s^2 & 1 - s \end{bmatrix}$$

we have

$$P^-(s) = \begin{bmatrix} 1 & 1 \\ s & 1 \end{bmatrix}, \quad P^+(s) = \begin{bmatrix} s & -1 \\ 1 & 1 \end{bmatrix}.$$

Algorithms and Software

Spectral factorization is a crucial step in the solution of various control, estimation, filtration, and other problems. It is no wonder that a variety of methods has been developed over the years for the computation of spectral factors. The most popular ones are briefly mentioned here. For details on particular algorithms, the reader is referred to the papers recommended for further reading.

Factor Extraction Method

If all roots of the right-hand side polynomial are known, the factorization becomes trivial. Just write the right-hand side as a product of first and second order factors and then collect the stable ones to create the stable factor. If the roots are not known, one can first enumerate them and then proceed as above. Somewhat surprisingly, a similar procedure can be used for the matrix case. To every zero, a proper matrix factor must be extracted. For further details, see Callier (1985) or Henrion and Sebek (2000).

Bauer's Algorithm

This procedure is an iterative scheme with linear rate of convergence. It relies on equivalence between the polynomial spectral factorization

and the Cholesky factorization of a related infinite-dimensional Toeplitz matrix. For further details, see Youla and Kazanjian (1978).

Newton-Raphson Iterations

An iterative algorithm with quadratic convergence rate based on consecutive solutions of symmetric linear polynomial Diophantine equations. It is inspired by the classical Newton's method for finding a root of a function. To learn more, read Davis (1963), Ježek and Kučera (1985), Vostrý (1975).

Factorization via Riccati Equation

In state-space solution of various problems, an algebraic Riccati equation plays the role of spectral factorization. It is therefore not surprising that the spectral factor itself can directly be calculated by solution of a Riccati equation. For further info, see e.g. Šebek (1992).

FFT Algorithm

This is the most efficient and accurate procedure for factorization of scalar polynomials with very high degrees (in orders of hundreds or thousands). Such polynomials appear in some special problems of signal processing in advanced audio applications involving inversions of dynamics of loudspeakers or room acoustics. The algorithm is based on the fact that logarithm of a product (such as the spectral factorization equation) turns into a sum of logarithms of particular entries. For details, see Hromčík and Šebek (2007)

All the procedures above are either directly programmed or can be easily composed from the functions of *Polynomial Toolbox for Matlab*, which is a third-party Matlab toolbox for polynomials, polynomial matrices and their applications in systems, signals, and control. For more details on the toolbox, visit www.polyx.com.

Consequences and Comments

Polynomial and polynomial matrix spectral factorization is an important step when frequency domain (polynomial) methods are used for optimal and robust control, filtering, estimation, or

prediction. Numerous particular examples can be found throughout this Encyclopedia as well as in the textbooks and papers recommended for further reading below.

Spectral factorization of rational functions and matrices is an equally important topic but it is omitted here due to lack of space. Inquiring readers are referred to the papers Oara and Varga (2000) and Zhong (2005).

Cross-References

- ▶ [Basic Numerical Methods and Software for Computer Aided Control Systems Design](#)
- ▶ [Classical Frequency-Domain Design Methods](#)
- ▶ [Computer-Aided Control Systems Design: Introduction and Historical Overview](#)
- ▶ [Control Applications in Audio Reproduction](#)
- ▶ [Discrete Optimal Control](#)
- ▶ [Extended Kalman Filters](#)
- ▶ [Frequency-Response and Frequency-Domain Models](#)
- ▶ [H-Infinity Control](#)
- ▶ [H₂ Optimal Control](#)
- ▶ [Kalman Filters](#)
- ▶ [Optimal Control via Factorization and Model Matching](#)
- ▶ [Optimal Sampled-Data Control](#)
- ▶ [Polynomial/Algebraic Design Methods](#)
- ▶ [Quantitative Feedback Theory](#)
- ▶ [Robust Synthesis and Robustness Analysis Techniques and Tools](#)

Recommended Reading

Nice tutorial books on polynomials and polynomial matrices in control theory and design are Kučera (1979), Callier and Desoer (1982), and Kailath (1980)

The concept of spectral factorization was introduced by Wiener (1949), for further information see later original papers Wilson (1972) or Kwakernaak and Šebek (1994) as well as survey papers Kwakernaak (1991), Sayed and Kailath (2001) or Kučera (2007).

Nice applications of spectral factorization in control problems can be found e.g. in Green et al. (1990), Henrion et al. (2003) or Zhou and Doyle (1998). For its use of in other engineering problems see e.g. Sternad and Ahlén (1993).

Bibliography

- Callier FM (1985) On polynomial matrix spectral factorization by symmetric extraction. *IEEE Trans Autom Control* 30:453–464
- Callier FM, Desoer CA (1982) *Multivariable feedback systems*. Springer, New York
- Davis MC (1963) Factorising the spectral matrix. *IEEE Trans Autom Control* 8:296
- Green M, Glover K, Limebeer DJN, Doyle J (1990) A J -spectral factorization approach to H -infinity control. *SIAM J Control Opt* 28:1350–1371
- Henrion D, Šebek M (2000) An algorithm for polynomial matrix factor extraction. *Int J Control* 73(8):686–695
- Henrion D, Šebek M, Kučera V (2003) Positive polynomials and robust stabilization with fixed-order controllers. *IEEE Trans Autom Control* 48:1178–1186
- Hromčík M, Šebek M (2007) Numerical algorithms for polynomial Plus/Minus factorization. *Int J Robust Nonlinear Control* 17(8):786–802
- Jakubovič VA (1970) Factorization of symmetric matrix polynomials. *Dokl. Akad. Nauk SSSR*, 194(3):532–535
- Ježek J, Kučera V (1985) Efficient algorithm for matrix spectral factorization. *Automatica* 29: 663–669
- Kailath T (1980) *Linear systems*. Prentice-Hall, Englewood Cliffs
- Kučera V (1979) *Discrete linear control: the polynomial equation approach*. Wiley, Chichester
- Kučera V (2007) Polynomial control: past, present, and future. *Int J Robust Nonlinear Control* 17:682–705
- Kwakernaak H (1991) The polynomial approach to a H -optimal regulation. In: Mosca E, Pandolfi L (eds) *H-infinity control theory*. Lecture Notes in Maths, vol 1496. Springer, Berlin
- Kwakernaak H, Šebek M (1994) Polynomial J -spectral factorization. *IEEE Trans Autom Control* 39:315–328
- Oara C, Varga A (2000) Computation of general inner-outer and spectral factorizations. *IEEE Trans Autom Control* 45:2307–2325
- Sayed AH, Kailath T (2001) A survey of spectral factorization methods. *Numer Linear Algebra Appl* 8(6–7):467–496
- Sternad M, Ahlén A (1993) Robust filtering and feedforward control based on probabilistic descriptions of model errors. *Automatica* 29(3):661–679
- Šebek M (1992) J -spectral factorization via Riccati equation. In: *Proceedings of the 31st IEEE CDC*, Tuscon, pp 3600–3603

- Vostrý Z (1975). New algorithm for polynomial spectral factorization with quadratic convergence. *Kybernetika* 11:415, 248
- Wiener N (1949) Extrapolation, interpolation and smoothing of stationary time series. Wiley, New York
- Wilson GT (1972) The factorization of matricial spectral densities. *SIAM J Appl Math* 23:420
- Youla DC, Kazanjian NN (1978) Bauer-type factorization of positive matrices and the theory of matrix polynomials orthogonal on the unit circle. *IEEE Trans Circuits Syst* 25:57
- Zhong QC (2005) J -spectral factorization of regular para-Hermitian transfer matrices. *Automatica* 41: 1289–1293
- Zhou K, Doyle JC (1998) Essentials of robust control. Prentice-Hall, Upper Saddle River

Stability and Performance of Complex Systems Affected by Parametric Uncertainty

Boris Polyak and Pavel Shcherbakov
Institute of Control Science, Moscow, Russia

Abstract

Uncertainty is an inherent feature of all real-life complex systems. It can be described in different forms; we focus on the parametric description. The simplest results on stability of linear systems under parametric uncertainty are the Kharitonov theorem, edge theorem, and graphical tests. More advanced results include sufficient conditions for robust stability with matrix uncertainty, LMI tools, and randomized methods. Similar approaches are used for robust control synthesis, where performance issues are crucial.

Keywords

Edge theorem · Kharitonov theorem · Linear systems · Matrix · Parametric uncertainty and robustness · Quadratic stability · Randomized methods · Robust and optimal design · Robust stability · Tsytkin–polyak plot

Introduction

Mathematical models for systems and control are often unsatisfactory due to the incompleteness of the parameter data. For instance, the ideas of off-line optimal control can only be applied to real systems if all the parameters, exogenous disturbances, state equations, etc. are known precisely. Moreover, feedback control also requires a detailed information which is not available in most cases. For example, to drive a car with four-wheel control, the controller should be aware of the total weight, location of the center of gravity, weather conditions, and highway properties as well as many other data which may not be known. In that respect, even such a relatively simple real-life system can be considered a *complex* one; in such circumstances, control under uncertainty is a highly important issue.

The focus in this entry is on the *parametric uncertainty*; other types of uncertainty can be treated in more general models of robustness. This topic became particularly popular in the control community in the mid- to late 1980s of the previous century; at large, the results of this activity have been summarized in the monographs (Ackermann 1993; Barmish 1994; Bhattacharyya et al. 1995).

We start with problems of stability of polynomials with uncertain parameters and present the simplest robust stability results for this case together with the most important machinery. Next, we consider stability analysis for the matrix uncertainty; most of the results are just sufficient conditions. We present some useful tools for the analysis, such as the LMI technique and randomized methods. Robust control under parametric uncertainty is the next step; we briefly discuss several problem formulations for this case.

Stability of Linear Systems Subject to Parametric Uncertainty

Consider the closed-loop linear, time-invariant continuous time state space system

$$\dot{x} = Ax, \quad x(0) = x_0, \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is the state vector, x_0 is an arbitrary finite initial condition, and $A \in \mathbb{R}^{n \times n}$ is the state matrix. The system is stable (i.e., no matter what x_0 is, the solutions tend to zero as $t \rightarrow \infty$) if and only if all eigenvalues λ_i of the matrix A have negative real parts:

$$\operatorname{Re} \lambda_i < 0, \quad i = 1, \dots, n, \quad (2)$$

in which case, A is said to be a *Hurwitz* matrix. If it is known precisely, checking condition (2) is immediate. For instance, one might compute the characteristic polynomial

$$\begin{aligned} p(s) &= \det(sI - A) \\ &= a_0 + a_1s + \dots + a_{n-1}s^{n-1} + s^n \end{aligned} \quad (3)$$

of A (here, I is the identity matrix) and use any of the stability tests (e.g., the Routh algorithm, Routh–Hurwitz test, and graphical tests such as the Mikhailov plot or Hermite–Biehler theorem); see Gantmacher (2000). Alternatively, the eigenvalues can be directly computed using the currently available software, such as MATLAB.

However, things get complicated if the knowledge of the matrix A is incomplete; for instance, it can depend on the (real) parameters $q = (q_1, \dots, q_m)$ which take arbitrary values within the given intervals:

$$A = A(q), \quad \underline{q}_i \leq q_i \leq \bar{q}_i, \quad i = 1, \dots, m. \quad (4)$$

In that case, we arrive at the *robust stability problem*; i.e., the goal is to check if condition (2) holds for *all matrices* in the family (4).

The two main components of any robust stability setup are the *feasible set* $\mathcal{Q} \subset \mathbb{R}^\ell$, in which the uncertain parameters are allowed to take their values (usually a ball in some norm, e.g., the box as in (4)), and the *uncertainty structure*, which defines the functional dependence of the coefficients on the uncertain parameters. Of the

most interest are the affine and multiaffine dependence; typically, more general situations are hard to handle.

Simple Solutions

In some cases, the robust stability problem admits a simple solution. Perhaps the most striking example is the so-called Kharitonov theorem (Kharitonov 1978); also see Barmish (1994), where this seminal result is referred to as a *spark* because of its transparency and elegance.

Namely, consider the *interval polynomial family*

$$\begin{aligned} \mathcal{P} &= \{p(s) = q_0 + q_1s + \dots + q_ns^n, \\ &\quad \underline{q}_i \leq q_i \leq \bar{q}_i, \quad i = 0, \dots, n\}, \end{aligned} \quad (5)$$

where the coefficients q_i are allowed to take values in the respective intervals *independently of each other*, and distinguish the following four elements in this family:

$$p_1(s) = \underline{q}_0 + \underline{q}_1s + \bar{q}_2s^2 + \bar{q}_3s^3 + \dots$$

$$p_2(s) = \underline{q}_0 + \bar{q}_1s + \bar{q}_2s^2 + \underline{q}_3s^3 + \dots$$

$$p_3(s) = \bar{q}_0 + \bar{q}_1s + \underline{q}_2s^2 + \underline{q}_3s^3 + \dots$$

$$p_4(s) = \bar{q}_0 + \underline{q}_1s + \underline{q}_2s^2 + \bar{q}_3s^3 + \dots$$

By the Kharitonov theorem, the interval family (5) is robustly stable (i.e., all polynomials in (5) are Hurwitz having all roots with negative real parts) if and only if the *four Kharitonov polynomials*, p_1 , p_2 , p_3 , and p_4 , are Hurwitz.

A simple and transparent proof of this result can be obtained using the *value set concept* (Zadeh and Desoer 1963) and the *zero exclusion principle* (Frazer and Duncan 1929), the two general tools which are in the basis of many results in the area of robust stability. We illustrate these concepts via robust stability of polynomials.

Given the uncertain polynomial family

$$\mathcal{P}(s, \mathcal{Q}) = \{p(s, q), \quad q \in \mathcal{Q}\},$$

the set

$$\mathcal{V}(\omega) = \{p(j\omega, q) : \omega \geq 0, q \in \mathcal{Q}\}$$

is referred to as the *value set*, which is, by definition, the set on the complex plane obtained by fixing the argument s to be $j\omega$ for a certain value of ω and letting the uncertain parameter vector q sweep the feasible domain.

The zero exclusion principle states that, under certain regularity requirements, the uncertain polynomial family is robustly stable if and only if it contains a stable element and the following condition holds:

$$0 \notin \mathcal{V}(\omega) \quad \forall \omega \geq 0. \quad (6)$$

To use this machinery, one has to be able to compute efficiently the value set and check condition (6). For the interval family (5), the value set can be shown to be the rectangle with coaxial edges and the vertices being the values of the four Kharitonov polynomials; see Fig. 1.

Being an extremely propelling result, the Kharitonov theorem is not free of drawbacks. First of all, it is not capable of determining the maximal lengths of the uncertainty intervals that retain the robust stability. This relates to an

important notion of *robust stability margin*; for simplicity, we define this quantity for the case of the interval family (5). Namely, introduce the *nominal polynomial* $p_0(s)$ with coefficients

$$q_i^0 = (\bar{q}_i + \underline{q}_i)/2,$$

and the *scaling factors*

$$\alpha_i = (\bar{q}_i - \underline{q}_i)/2$$

for the deviations of the coefficients. Then the robust stability margin $r_{\max}$ is defined as follows:

$$r_{\max} = \sup\{r : p(s, q) \text{ (5) is stable } \forall q_i :$$

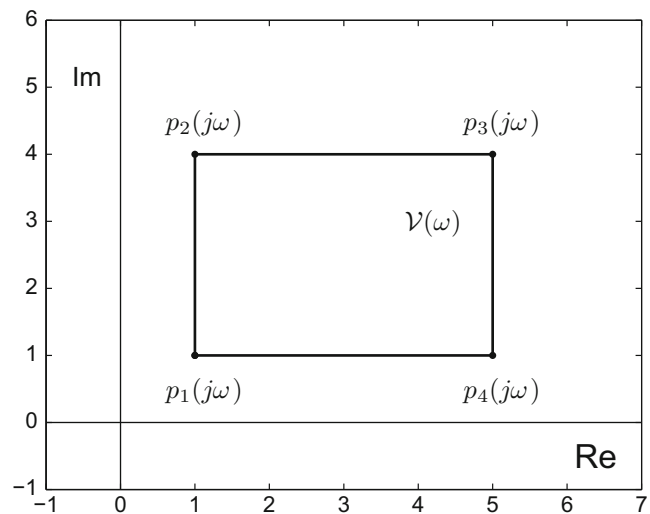
$$|q_i - q_i^0| \leq r\alpha_i, \quad i = 1, \dots, n\}.$$

Another drawback of the Kharitonov result is its inapplicability to the discrete-time case (Schur stability of polynomials).

A more flexible graphical test for robust stability uses the so-called Tsytkin–Polyak plot (Tsytkin and Polyak 1991), which is defined as the parametric curve on the complex plane:

$$z(\omega) = x(\omega) + jy(\omega), \quad j = \sqrt{-1}; \quad 0 \leq \omega < \infty,$$

Stability and Performance of Complex Systems Affected by Parametric Uncertainty,
Fig. 1 The Kharitonov rectangular value set



where

$$x(\omega) = \frac{q_0^0 - q_2^0 \omega^2 + \dots}{\alpha_0 + \alpha_2 \omega^2 + \dots},$$

$$y(\omega) = \frac{q_1^0 - q_3^0 \omega^2 + \dots}{\alpha_1 + \alpha_3 \omega^2 + \dots}.$$

Then, by the Tsytkin–Polyak criterion, the polynomial family (5) is robustly stable if and only if the following conditions hold: (i) $q_0^0 > \alpha_0$, $q_n^0 > \alpha_n$, and (ii) as ω changes zero to infinity, the curve $z(\omega)$ goes consecutively through n quadrants in the counterclockwise direction and does not intersect the unit square with the vertices $(\pm 1, \pm j)$.

Unlike the Kharitonov theorem, with this test, the robust stability margin of family (5) can be determined as the size of the maximal square inscribed in the curve $z(\omega)$; see Fig. 2. Moreover, with minor modifications, this test applies to *dependent uncertainty structures* where the coefficient vector $q = (q_0, \dots, q_n)^\top$ is confined to a ball in ℓ_p -norm, not to a *box* as in (5).

On top of that, the Tsytkin–Polyak plot can be built for discrete-time systems which do not admit any counterparts of the Kharitonov theorem.

It is fair to say that interval polynomial families are an idealization, since the coefficients of the characteristic polynomial can hardly be

thought of as the physical parameters of the real-world system. As a step toward more realistic formulations, consider the *affine polynomial family* of the form

$$P(s) = p_0(s) + \sum_{i=1}^m q_i p_i(s), |q_i| \leq 1, i=1, \dots, m, \quad (7)$$

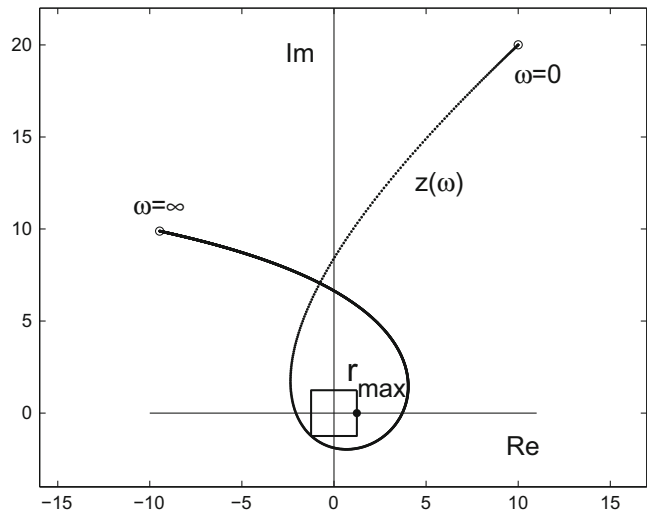
where p_i are the given polynomials and the q_i s are the uncertain parameters (clearly, they can be scaled to take values in the segment $[-1, 1]$). The famous *edge theorem* (Bartlett et al. 1988) claims that checking the robust stability of such a family is equivalent to checking the *edges* of the uncertainty box, i.e., the points $q \in \mathbb{R}^m$ with all but one components being fixed to ± 1 , while the “free” coordinate varies in $[-1, 1]$.

Complex Solutions

Obviously, the affine model (7) covers just a small part of problems with parametric uncertainty. Closed-form solutions cannot be obtained in the general case; however, many important classes of systems can be analyzed efficiently.

Thus, in the engineering practice, *block diagram description* of systems is often more convenient than differential equations of the form (1). The blocks are associated with typical elements such as amplifiers, integrators, lag elements, and

Stability and Performance of Complex Systems Affected by Parametric Uncertainty, Fig. 2 The Tsytkin–Polyak plot



oscillators, which are connected in a certain circuit. In this case, transfer functions are the most adequate tool for dealing with such systems. For instance, the transfer function of the lag element is given by

$$W(s) = 1/(Ts + 1),$$

where the scalar T is the *time constant* of the element. In terms of differential equations, this means that the input $u(t)$ of a block and its output $x(t)$ satisfy the equation $T\dot{x} + x = u$.

Assume now we have a set of m cascade connected elements with uncertain time constants

$$\underline{T}_i \leq T_i \leq \bar{T}_i, \quad i = 1, \dots, m, \quad (8)$$

with known lower and upper bounds. The characteristic polynomial of such a connection embraced by the feedback with *gain* k is known to have the form

$$p(s) = k + (1 + T_1 s) \cdots (1 + T_m s). \quad (9)$$

Hence, the robust stability problem reduces to checking if all polynomials (9) with constraints (8) are Hurwitz. Note that the coefficients of such a polynomial depend *multilinearly* on the uncertain parameters T_i (cf. linear dependence in (7)), making the problem much more complicated.

The solution of the problem above was obtained in Kiselev et al. (1997) for many important special cases; the closely related problem of finding the “critical gain” (the maximal value of k retaining the robust stability) was also addressed.

Using the similar technique, closed-form solutions can be obtained for a number of similar problems such as robust sector stability, robust stability of distributed systems, and robust D -decomposition, to name just a few.

Difficult Problems: Possible Approaches

In spite of the apparent progress obtained in the area of parametric robustness, the list of unsolved problems is still quite large. Moreover, some of

the formulations were shown to be NP-hard, making it hard to believe that any efficient solution methods will ever be found.

One of such fundamental problems is robust stability of the *interval matrix*. Specifically, assume that the entries a_{ij} of the matrix A in (1) are interval numbers

$$\underline{a}_{ij} \leq a_{ij} \leq \bar{a}_{ij}, \quad i, j = 1, \dots, n;$$

the problem is to check if the interval matrix is robustly stable, i.e., if the eigenvalues of all matrices in this family have negative real parts. Numerous attempts to prove a Kharitonov-like theorem for matrices have failed, and the results by Nemirovskii (1994) on NP-hardness showed that these generalizations are not possible. It was also shown that the edge theorem for matrix families is not valid. The other NP-hard problems in robustness include the analysis of systems with interval delays, parallel connection of uncertain blocks, problem (9)–(8) with nested segments $[\underline{T}_i, \bar{T}_i]$, and others.

However, a change in the statement of the problem often allows for simple and elegant solutions. We mention three fruitful reformulations.

In the first approach, the uncertain parameters are assumed to have random rather than deterministic nature; for instance, they are assumed to be uniformly distributed over the respective intervals of uncertainty. We next specify an acceptable tolerance ε , say $\varepsilon = 0.01$, and check if the resulting random family of polynomials is stable with probability no less than $(1 - \varepsilon)$; see Tempo et al. (2013) for a comprehensive exposition of such a *randomized approach to robustness*; also see Polyak and Shcherbakov (2018) for a more recent development.

In many of the NP-hard robustness problems, such a reformulation often leads to exact or approximate solutions. Moreover, the randomized approach has several attractive properties even in the situations where the deterministic solution is available. Indeed, the deterministic statements of robustness problems are minimax; hence, the answer is dictated by the “worst” element in the family, whereas these critical values of the uncertain parameters are rather

unlikely to occur. Therefore, by neglecting a small risk of violation of the stability, the admissible domains of variation of the parameters may be considerably extended. This effect is known as the *probabilistic enhancement of robustness margins*; it is particularly tangible for the large number of the parameters. Another attractive property of the randomized approach is its low computational complexity which only slowly grows with increase of the number of uncertain parameters.

To illustrate, let us turn back to problem (9)–(8) and use the value set approach. In the considered problem, this set can be efficiently built.

Assume now that the parameters T_i are independent random variables uniformly distributed over the respective segments (8) and consider the random variable

$$\eta = \eta(\omega) = \log(p(j\omega) - k) = \sum_{i=1}^m \log(1 + j\omega T_i).$$

The right-hand side of the last relation is the sum of independent complex-valued random variables; for m large, its behavior obeys the central limit theorem, so that the probability that η belongs to the respective *confident ellipse* $\mathcal{E} = \mathcal{E}(\omega)$ is close to unity. In other words, we have

$$p(j\omega) \approx k + e^{\mathcal{E}} \doteq \mathcal{G}(\omega),$$

and the set $\mathcal{G}(\omega)$ is referred to as a *probabilistic predictor* of the value set $\mathcal{V}(\omega)$; it is the shifted set of points of the form e^z , $z \in \mathcal{E} \subset \mathbb{C}$. The predictor $\mathcal{G}(\omega)$ constitutes a small portion of the deterministic value set $\mathcal{V}(\omega)$, yielding the probabilistic enhancement of the robustness margin.

Note also that the computation of $\mathcal{E}$ and $e^{\mathcal{E}}$ is nearly trivial, and, in contrast to the construction of the true value set $\mathcal{V}$, the complexity does not grow with increase of m .

The *second approach* to solving “hard” problems in robust stability relates to the notion of *superstability* (Polyak and Shcherbakov 2002). The matrix A of system (1) (and the system itself) is said to be *superstable*, if its entries a_{ij} , $i, j = 1, \dots, n$, satisfy the relations

$$a_{ii} < 0, \quad \min_i (-a_{ii} - \sum_{j \neq i} |a_{ij}|) = \sigma > 0.$$

The following estimate holds for the solutions of the superstable system (1):

$$\|x(t)\|_{\infty} \leq \|x(0)\|_{\infty} e^{-\sigma t},$$

i.e., it is stable, and the (nonsmooth) function $\|x\|_{\infty}$ is a Lyapunov function for the system. Since the condition of superstability is formulated in terms of linear inequalities on the entries of A , checking robust superstability of affine (and in particular, interval) matrix families is immediate. Similar situation holds for so-called positive systems.

The *third approach* to robustness analysis relates to *quadratic stability* (Leitmann 1979; Boyd et al. 1994). Namely, a family of systems is said to be *robustly quadratically stable* if it possesses a common quadratic Lyapunov function $V(x) = x^{\top} P x$ with positive definite matrix P . In other words, an uncertain family of matrices $A(q)$, $q \in \mathcal{Q}$ has to satisfy the following set of the matrix Lyapunov-type inequalities:

$$A(q)P + PA(q)^{\top} < 0, \quad q \in \mathcal{Q}, \text{ and } P > 0,$$

where the symbols $<$, $>$ stand for the sign definiteness of a matrix.

The inequality above is referred to as a *linear matrix inequality* (LMI), (Boyd et al. 1994); there exist both efficient numerical methods for solving such inequalities (*interior point methods*) and various software, e.g., MATLAB. This approach can be directly applied at least in the following two cases: (i) the set $\mathcal{Q}$ contains a finite number of points, and (ii) $\mathcal{Q}$ is a polyhedron, and the dependence $A(q)$ is affine. In the general setup or in the high-dimensional problems, randomized methods can be employed.

Finding the *quadratic robust stability margin* (by analogy with the stability margin, this is the maximum span of the feasible set $\mathcal{Q}$ that allows for the existence of the common Lyapunov function) in this problem is also possible; it reduces to the minimization of a linear function over the solutions of a similar LMI.

Note that the approaches based on superstability and quadratic stability provide only sufficient conditions for robustness.

Robust Control

So far, of our primary interest was in assessing the robust stability of a closed-loop system with synthesized linear feedback. A more important problem is to *design* a controller that makes the closed-loop system robustly stable and guarantees certain *robust performance* of the system.

Robust Stabilization

Let the linear system

$$\dot{x} = A(q)x + Bu$$

depend on the vector $q \in \mathcal{Q}$ of uncertain parameters. In the simplest form, the problem of *robust stabilization* consists in finding the linear static state feedback

$$u = Kx$$

that guarantees the robust stability of the closed-loop system. Alternatively, static or dynamic *output* robustly stabilizing controllers can be considered in the situations where only the linear output $y = Cx$ of the system is available, but not the complete state vector x .

If the number of controller parameters to be tuned is small (which is the case for PI or PID controllers), then the design can be accomplished using the *D-decomposition technique*.

In the general formulation, the problem of robust design is complicated; it can, however, be addressed with the use of randomized methods (Tempo et al. 2013). Other plausible approaches include superstability and quadratic stability; respectively, the problem reduces to solving linear programs or linear matrix inequalities in the coefficients of the controller.

Robust Performance

Needless to say, the robust stabilization problem is not the only one in the area of optimal control. As a rule, a certain cost function is always involved (say, integral quadratic), and its desired

value should be guaranteed for all admissible values of the uncertain parameters. Moreover, robust stability is a necessary condition for such a guaranteed estimate to exist. This sort of problems can often be cast in the form of LMIs which must be satisfied for all admissible values of the parameters. Such robust LMIs can be solved either directly or using various randomized techniques presented in Tempo et al. (2013).

Summary and Future Directions

In spite of the considerable progress attained in the parametric robustness of complex systems, this topic is still a vivid and active research area. To date, randomization, superstability, and quadratic stability present the most efficient and diverse tools for the analysis and design of systems affected by parametric uncertainty.

In this article, we paid no attention to several related issues that might be worth discussing. Among them is robustness assessment in discrete-time systems, circular (rather than interval) uncertainty, additive norm-bounded matrix uncertainty, and, importantly, parameter-dependent Lyapunov functions.

Cross-References

- [H-Infinity Control](#)
- [LMI Approach to Robust Control](#)
- [Optimization-Based Robust Control](#)
- [Randomized Methods for Control of Uncertain Systems](#)

Bibliography

- Ackermann J (1993) Robust control: systems with uncertain physical parameters. Springer, London
- Barmish BR (1994) New tools in robustness of linear systems. Macmillan, New York
- Bartlett AC, Hollot CV, Lin H (1988) Root locations of an entire polytope of polynomials: it suffices to check the edges. Math Control Sig Syst 1(1):61–71
- Bhattacharyya SP, Chapellat H, Keel LH (1995) Robust control: the parametric approach. Prentice Hall, Upper Saddle River
- Boyd S, El Ghaoui L, Feron E, Balakrishnan V (1994) Linear matrix inequalities in system and control theory. SIAM, Philadelphia

- Frazer RA, Duncan WJ (1929) On the criteria for the stability of small motions. *Proc R Soc Lond A* 124(795):642–654
- Gantmacher FR (2000) *The theory of matrices*. AMS, Providence
- Kharitonov VL (1978) Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. *Differentsial'nye Uravneniya* 14:2086–2088
- Kiselev ON, Le Hung Lan, Polyak BT (1997) Frequency responses under parametric uncertainty. *Autom Remote Control* 58(Pt. 2, 4):645–661
- Leitmann G (1979) Guaranteed asymptotic stability for some linear systems with bounded uncertainties. *J Dyn Syst Meas Control* 101(3):212–216
- Nemirovskii AS (1994) Several NP-hard problems arising in robust stability analysis. *Math Control Sig Syst* 6(1):99–105
- Polyak BT, Shcherbakov PS (2002) Superstable linear control systems. I. Analysis; II. Design. *Autom Remote Control* 63(8):1239–1254; 63(11):1745–1763
- Polyak B, Shcherbakov P (2018) Randomization in robustness, estimation, and optimization. In: Basar T (ed) *Uncertainty in complex networked systems: in honor of Roberto Tempo*. Springer, pp 181–208
- Tempo R, Calafiore G, Dabbene F (2013) *Randomized algorithms for analysis and control of uncertain systems, with applications*, 2nd edn. Springer, London
- Tsyppkin YZ, Polyak BT (1991) Frequency domain criteria for l^p -robust stability of continuous systems. *IEEE Trans Autom Control* 36(12):1464–1469
- Zadeh LA, Desoer CA (1963) *Linear system theory – a state space approach*. McGraw-Hill, New York

Stability Theory for Hybrid Dynamical Systems

Andrew R. Teel
Electrical and Computer Engineering
Department, University of California, Santa
Barbara, CA, USA

Abstract

This entry provides a short introduction to modeling of hybrid dynamical systems and then focuses on stability theory for these systems. It provides definitions of asymptotic stability, basin of attraction, and uniform asymptotic stability for a compact set. It points out

mild assumptions under which different characterizations of asymptotic stability are equivalent, as well as when an asymptotically stable compact set exists. It also summarizes necessary and sufficient conditions for asymptotic stability in terms of Lyapunov functions.

Keywords

Asymptotic stability · Basin of attraction · Hybrid system · Lyapunov function

Introduction

A hybrid dynamical system combines continuous change and instantaneous change. Instantaneous change is the only type of change available for variables like counters, switches, and logic variables. Instantaneous change may also be a good approximation of what occurs to velocities in mechanical systems at the time of an impact with a wall, floor, or some other rigid body. At other times, velocities evolve continuously. Continuous change is also natural for position variables, continuous timers, and voltages and currents. For mathematical convenience, it is typical in the analysis of hybrid dynamical systems to embed all of these variables into a Euclidean space, with the understanding that many points in the state space will never be reached. For example, a logic variable that naturally takes values in the set {off, on} is typically embedded in the real number line where its two distinct values are associated with two distinct numbers, the only numbers that this variable will visit during its evolution.

A finite-dimensional dynamical system that exhibits continuous change exclusively is typically modeled by an ordinary differential equation, or sometimes a more flexible differential inclusion. A system that exhibits purely instantaneous change is typically modeled by a difference equation or inclusion. Consequently, a hybrid dynamical system combines a differential equation or inclusion with a difference equation or inclusion. A big part of the modeling effort for hybrid systems is directed at determining which type of evolution should be allowed at each point

in the state space. To this end, subsets of the state space are specified where each type of behavior is allowed.

Though the behavior of a hybrid dynamical system can be quite complex and nonconventional, it is still reasonable to ask the same stability questions for them that might be asked about classical differential or difference equations. Moreover, the same stability analysis tools that are used for classical systems are also quite useful for hybrid dynamical systems. The emphasis of this entry is on basic stability theory for hybrid dynamical systems, focusing on definitions and tools that also apply to classical systems.

Mathematical Modeling

System Data

A hybrid dynamical system with state x belonging to a Euclidean space $\mathbb{R}^n$ combines a differential equation or inclusion, written formally as $\dot{x} = f(x)$ or $\dot{x} \in F(x)$, with a difference equation or inclusion $x^+ = g(x)$ or $x^+ \in G(x)$, where $\dot{x}$ indicates the time derivative and x^+ indicates the value after an instantaneous change. The mapping f or F is called the *flow map*, while the mapping g or G is called the *jump map*. A complete model also specifies where in the state space continuous evolution is allowed and where instantaneous change is allowed. The set where continuous evolution is allowed is called the *flow set* and is denoted C , whereas the set where instantaneous change is allowed is called the *jump set* and is denoted D . The overall model, using inclusions for generality, is written formally as

$$x \in C \quad \dot{x} \in F(x) \quad (1a)$$

$$x \in D \quad x^+ \in G(x). \quad (1b)$$

Solutions

It is natural for solutions of (1) to be functions of two different types of time: a variable t that keeps track of the amount of ordinary time that has elapsed and a variable j that counts the number of

jumps. There is a special structure to the types of domains that are allowed. A *compact hybrid time domain* is a set $E \subset \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0}$, that is, a subset of the product of the nonnegative real numbers and the nonnegative integers, of the form

$$E = \bigcup_{i=0}^J ([t_i, t_{i+1}] \times \{i\})$$

for some $J \in \mathbb{Z}_{\geq 0}$ and some sequence of non-decreasing times $0 = t_0 \leq t_1 \leq \dots \leq t_{J+1}$. It is possible for several of these times to be the same, which would correspond to more than one jump at the given time. A *hybrid time domain* is a set $E \subset \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ such that for each $(T, J) \in E$, the set $E \cap ([0, T] \times \{0, \dots, J\})$ is a compact hybrid time domain. In contrast to a compact hybrid time domain, a hybrid time domain may have an infinite number of intervals, or it may have a finite number of intervals with the last one being unbounded or of the form $[t_J, t_{J+1})$; that is, it may be open on the right. A *hybrid arc* is a function x , defined on a hybrid time domain, such that $t \mapsto x(t, j)$ is locally absolutely continuous for each j ; in particular, $t \mapsto x(t, j)$ is differentiable for almost every t where it is defined, and this mapping is the integral of its derivative. The notation “dom x ” denotes the domain of x . Finally, a hybrid arc is a *solution* of (1) if the following two properties are satisfied:

1. For $\varepsilon > 0$, $(s, j), (s + \varepsilon, j) \in \text{dom } x$ implies that $x(t, j) \in C$ and $\dot{x}(t, j) \in F(x(t, j))$ for almost all $t \in [s, s + \varepsilon]$.
2. $(t, j), (t, j+1) \in \text{dom } x$ implies that $x(t, j) \in D$ and $x(t, j+1) \in G(x(t, j))$.

For a hybrid system with no flow dynamics, each solution has a time domain of the form $\{0\} \times \{0, \dots, J\}$ for some $J \in \mathbb{Z}_{\geq 0}$ or $\{0\} \times \mathbb{Z}_{\geq 0}$. For a hybrid system with no jump dynamics, each solution has a time domain of the form $[0, \infty) \times \{0\}$, $[0, T] \times \{0\}$, or $[0, T) \times \{0\}$ for some $T \geq 0$. No assumptions are made in this entry to guarantee existence of nontrivial solutions since stability theory does not hinge

on existence of solutions; rather, it simply makes statements about the behavior of solutions when they exist. To ensure robustness of various stability properties, the following basic regularity assumptions are usually imposed.

Assumption 1 The data (C, F, D, G) satisfy the following conditions:

1. The sets C and D are closed.
2. The set-valued mapping F is outer semicontinuous, locally bounded, and $F(x)$ is nonempty and convex for each $x \in C$.
3. The set-valued mapping G is outer semicontinuous, locally bounded, and $G(x)$ is nonempty for each $x \in D$.

To elaborate further, a set-valued mapping, like F , is said to be *outer semicontinuous* if for each convergent sequence $\{(x_i, y_i)\}_{i=0}^{\infty}$ that satisfies $y_i \in F(x_i)$ for all $i \in \mathbb{Z}_{\geq 0}$, its limit, denoted (x, y) , satisfies $y \in F(x)$. It is said to be *locally bounded* if for each bounded set $K_1 \subset \mathbb{R}^n$ there exists a bounded set $K_2 \subset \mathbb{R}^n$ such that, for every $x \in K_1$, every $y \in F(x)$ belongs to K_2 ; the latter condition is sometimes written $F(K_1) \subset K_2$. If C is closed, f is a function $f : C \rightarrow \mathbb{R}^n$ that is continuous, and F is a set-valued mapping that has the single value $f(x)$ for each $x \in C$ and is empty for $x \notin C$, then F is outer semicontinuous, locally bounded, and $F(x)$ is nonempty and convex for each $x \in C$.

Stability Theory

Definitions and Relationships

Given a dynamical system, predicting or controlling the system's long-term behavior is of primary importance. A system's long-term behavior may be more complicated than just converging to an equilibrium point. This fact motivates studying stability of and convergence to a set of points. For simplicity, this entry focuses on stability of sets that are *compact*, that is, they are closed and bounded. A variety of stability concepts are defined below. Each of these concepts applies

to continuous-time or discrete-time systems as readily as to hybrid systems.

A compact set $\mathcal{A} \subset \mathbb{R}^n$ is said to be *Lyapunov stable* for (1) if for each $\varepsilon > 0$ there exists $\delta > 0$ such that for every solution of (1), $x(0, 0) \in \mathcal{A} + \delta\mathbb{B}$ implies $x(t, j) \in \mathcal{A} + \varepsilon\mathbb{B}$ for all $(t, j) \in \text{dom } x$, where $\mathcal{A} + \delta\mathbb{B}$ indicates the set of points whose distance to the set $\mathcal{A}$ is less than or equal to δ . In order for a compact set to be Lyapunov stable for (1), it must be *forward invariant* for (1), that is, each solution of (1) with $x(0, 0) \in \mathcal{A}$ satisfies $x(t, j) \in \mathcal{A}$ for all $(t, j) \in \text{dom } x$. However, forward invariance does not necessarily imply Lyapunov stability.

For a compact set $\mathcal{A} \subset \mathbb{R}^n$, its *basin of attraction* for (1), denoted $\mathcal{B}_{\mathcal{A}}$, is the set of points from which each solution to (1) is bounded and each solution to (1) having an unbounded time domain converges to $\mathcal{A}$, the latter being written mathematically as $\lim_{t+j \rightarrow \infty} |x(t, j)|_{\mathcal{A}} = 0$ where $|x(t, j)|_{\mathcal{A}}$ denotes the distance of $x(t, j)$ to the set $\mathcal{A}$. Each point that does not belong to $C \cup D$ belongs to $\mathcal{B}_{\mathcal{A}}$ since there are no solutions from such points. A compact set $\mathcal{A}$ is said to be *attractive* for (1) if its basin of attraction contains a neighborhood of itself, that is, there exists $\varepsilon > 0$ such that $\mathcal{A} + \varepsilon\mathbb{B} \subset \mathcal{B}_{\mathcal{A}}$. A compact set $\mathcal{A}$ is said to be *globally attractive* if $\mathcal{B}_{\mathcal{A}} = \mathbb{R}^n$.

A compact set is said to be *asymptotically stable* for (1) if it is Lyapunov stable and attractive for (1). A compact set is said to be *globally asymptotically stable* for (1) if it asymptotically stable for (1) and $\mathcal{B}_{\mathcal{A}} = \mathbb{R}^n$. It is useful to know that the basin of attraction for an asymptotically stable set is always open.

Theorem 1 Under Assumption 1, if a compact set is asymptotically stable for (1), then its basin of attraction is an open set.

A compact set $\mathcal{A} \subset \mathbb{R}^n$ is said to be *uniformly attractive* for (1) if it is attractive for (1) and for each compact set $K \subset \mathcal{B}_{\mathcal{A}}$ and each $\delta > 0$ there exists $T > 0$ such that for every solution x of (1), $x(0, 0) \in K$ and $t + j \geq T$ imply $x(t, j) \in \mathcal{A} + \delta\mathbb{B}$. A compact set is said to be *uniformly globally attractive* for (1) if it is globally attractive and uniformly attractive for (1). Uniform attractivity goes beyond attractivity

by asking that the amount of time it takes each solution to get close to $\mathcal{A}$ is uniformly bounded over initial conditions in compact subsets of the basin of attraction.

A compact set $\mathcal{A} \subset \mathbb{R}^n$ is said to be *Lagrange stable* relative to an open set $O \supset \mathcal{A}$ for (1) if for each compact set $K_1 \subset O$ there exists a compact set $K_2 \subset O$ such that for every solution of (1), $x(0, 0) \in K_1$ implies $x(t, j) \in K_2$ for all $(t, j) \in \text{dom } x$. In Lagrange stability for the case $O = \mathbb{R}^n$, a bound on the initial conditions is given and a bound on the ensuing solutions must be found; this is in contrast to Lyapunov stability where a bound on the solutions is given and a bound on the initial conditions must be found.

A compact set is said to be *uniformly asymptotically stable* for (1) if it is Lyapunov stable, attractive, Lagrange stable relative to its basin of attraction, and uniformly attractive for (1). A compact set is said to be *uniformly globally asymptotically stable* for (1) if it is uniformly asymptotically stable for (1) and $\mathcal{B}_{\mathcal{A}} = \mathbb{R}^n$. There is no difference between asymptotic stability and uniform asymptotic stability under Assumption 1.

Theorem 2 *Under Assumption 1, a compact set is uniformly asymptotically stable for (1) if and only if it is locally asymptotically stable for (1).*

As noted earlier, forward invariance does not imply Lyapunov stability. However, when coupled with uniform attractivity, Lyapunov stability ensues.

Theorem 3 *Under Assumption 1, a compact set is uniformly asymptotically stable for (1) if and only if it is forward invariant and uniformly attractive for (1).*

Asymptotic stability can be converted to global asymptotic stability by shrinking the flow and jump sets to be compact subsets of the basin of attraction. However, global asymptotic stability of a compact set $\mathcal{A}$ for $x \in C$, $\dot{x} = f(x)$ for each compact set C does not necessarily imply global asymptotic stability of $\mathcal{A}$ for $\dot{x} = f(x)$.

In some situations it is easier to assert the existence of a compact asymptotically stable set than it is to find one explicitly. In this direction, given a

set $X \subset \mathbb{R}^n$, consider the set of points z with the property that there exist a sequence of solutions $\{x_i\}_{i=0}^{\infty}$ to (1) with initial conditions in X and an unbounded sequence of times $\{(t_i, j_i)\}_{i=0}^{\infty}$ with $(t_i, j_i) \in \text{dom } x_i$ for each $i \in \mathbb{Z}_{\geq 0}$ such that $z = \lim_{i \rightarrow \infty} x_i(t_i, j_i)$. This set of points is called the ω -limit set of X for (1) and is denoted $\Omega(X)$.

Theorem 4 *Let Assumption 1 hold. For the system (1), if X is compact and $\Omega(X)$ is nonempty and contained in the interior of X (i.e., there exists $\varepsilon > 0$ such that $\Omega(X) + \varepsilon\mathbb{B} \subset X$), then the set $\Omega(X)$ is compact and uniformly asymptotically stable with basin of attraction containing X and equal to the basin of attraction for X .*

Robustness

A given model (C, F, D, G) may have some mismatch with a physical process that it aims to describe. One way to capture some of this mismatch is to consider the behavior of solutions to a system with inflated data $(C_{\delta}, F_{\delta}, D_{\delta}, G_{\delta})$, $\delta \geq 0$, defined as follows:

$$C_{\delta} := \{x \in \mathbb{R}^n : (x + \delta\mathbb{B}) \cap C \neq \emptyset\} \quad (2a)$$

$$F_{\delta}(x) := \overline{\text{co}}F((x + \delta\mathbb{B}) \cap C) + \delta\mathbb{B} \quad (2b)$$

$$D_{\delta} := \{x \in \mathbb{R}^n : (x + \delta\mathbb{B}) \cap D \neq \emptyset\} \quad (2c)$$

$$G_{\delta} := G((x + \delta\mathbb{B}) \cap D) + \delta\mathbb{B}. \quad (2d)$$

The notation $x + \delta\mathbb{B}$ indicates a closed ball of radius δ centered at the point x . Evaluating a set-valued mapping at a set of points means to collect all vectors that belong to the set-valued mapping at any point in the set that serves as the argument of the set-valued mapping. The notation “ $\overline{\text{co}}F((x + \delta\mathbb{B}) \cap C)$ ” indicates the closed, convex hull of the set $\{f \in \mathbb{R}^n : f \in F(z), z \in (x + \delta\mathbb{B}) \cap C\}$. Note that $(C_0, F_0, D_0, G_0) = (C, F, D, G)$. More generally, the components of (C, F, D, G) are contained in $(C_{\delta}, F_{\delta}, D_{\delta}, G_{\delta})$. The inflation data in (2) satisfy the regularity properties of Assumption 1 when (C, F, D, G) do.

Proposition 3 *If the data (C, F, D, G) satisfy Assumption 1 then, for each $\delta > 0$, the inflated data $(C_\delta, F_\delta, D_\delta, G_\delta)$ satisfy Assumption 1.*

From the point of view of asymptotic stability, the behavior of solutions to $(C_\delta, F_\delta, D_\delta, G_\delta)$ for $\delta > 0$ small is not too different from those of (C, F, D, G) .

Theorem 5 *Under Assumption 1, if $\mathcal{A}$ is asymptotically stable with basin of attraction $\mathcal{B}_{\mathcal{A}}$ for the hybrid system with data (C, F, D, G) , then for each $\varepsilon > 0$ and each compact set K satisfying $K \subset \mathcal{B}_{\mathcal{A}}$, there exist $\delta > 0$ and a compact set $\mathcal{A}_\delta \subset \mathcal{A} + \varepsilon\mathbb{B}$ that is asymptotically stable with $K \subset \mathcal{B}_{\mathcal{A}_\delta}$ for $(C_\delta, F_\delta, D_\delta, G_\delta)$.*

The robustness result of Theorem 5 has several consequences beyond the observations in the preceding examples. One of the consequences is the following reduction principle.

Theorem 6 *Under Assumption 1, if $\mathcal{A}_1$ is asymptotically stable with basin of attraction $\mathcal{B}_{\mathcal{A}_1}$ for the hybrid system with data (C, F, D, G) and the compact set $\mathcal{A}_2 \subset \mathcal{A}_1$ is globally asymptotically stable for the hybrid system with data $(C \cap \mathcal{A}_1, F, C \cap \mathcal{A}_2, G)$, then the compact set $\mathcal{A}_2$ is asymptotically stable with basin of attraction $\mathcal{B}_{\mathcal{A}_1}$ for the hybrid system with data (C, F, D, G) .*

Lyapunov Functions

Arguably the most common method for establishing asymptotic stability is known as *Lyapunov's method* and uses a Lyapunov function. A function $V : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$ is a *Lyapunov function candidate* for (1) if it is continuously differentiable on an open neighborhood of the flow set C , it is defined for all $x \in C \cup D \cup G(D)$ (dom V denotes the set of points where it is defined), and it is continuous on its domain. Some of these conditions can be relaxed but are imposed in this entry to keep the discussion simple. Given a compact set $\mathcal{A}$ and an open set O satisfying $\mathcal{A} \subset O \subset \mathbb{R}^n$, a Lyapunov function candidate for (1) is called a *Lyapunov function* for $(\mathcal{A}, O)$ if:

(L1) For $x \in (C \cup D \cup G(D)) \cap O$, $V(x) = 0$ if and only if $x \in \mathcal{A}$.

(L2) For each $x \in C \cap O$ and $f \in F(x)$, $\langle \nabla V(x), f \rangle \leq 0$.

(L3) For each $x \in D \cap O$ and $g \in G(x)$, $V(g) - V(x) \leq 0$.

A Lyapunov function for $(\mathcal{A}, O)$ is called a *proper Lyapunov function* for $(\mathcal{A}, O)$ if, in addition,

(L4) $\lim_{i \rightarrow \infty} V(x_i) = \infty$ when the sequence $\{x_i\}_{i=0}^\infty$, satisfying $x_i \in (C \cup D \cup G(D)) \cap O$ for all $i \in \mathbb{Z}_{\geq 0}$, is unbounded or approaches the boundary of O .

The next result does not use Assumption 1, though the rest of the results in this entry do.

Theorem 7 *Let $\mathcal{A} \subset O \subset \mathbb{R}^n$ with $\mathcal{A}$ compact and O open. If there exists a Lyapunov function for $(\mathcal{A}, O)$, then $\mathcal{A}$ is Lyapunov stable for (1). If there exists a proper Lyapunov function for $(\mathcal{A}, O)$ then $\mathcal{A}$ is also Lagrange stable with respect to O for (1).*

We can also conclude asymptotic stability from a Lyapunov function when it is known that there are no complete solutions along which the Lyapunov function is equal to a positive constant.

Theorem 8 *Let $\mathcal{A} \subset O \subset \mathbb{R}^n$ with $\mathcal{A}$ compact and O open. Under Assumption 1, if there exists a Lyapunov function for $(\mathcal{A}, O)$ and there is no solution x of (1) starting in $O \setminus \mathcal{A}$ that has an unbounded time domain and satisfies $V(x(t, j)) = V(x(0, 0))$ for all $(t, j) \in \text{dom } x$, then $\mathcal{A}$ is uniformly asymptotically stable for (1). If the Lyapunov function is a proper Lyapunov function for $(\mathcal{A}, O)$, then the basin of attraction for $\mathcal{A}$ contains O .*

The simplest way to rule out solutions that keep a Lyapunov function equal to a positive constant is by finding a (proper) *strict Lyapunov function* for $(\mathcal{A}, O)$, which is a (proper) Lyapunov function for $(\mathcal{A}, O)$ that also satisfies:

(L2') For each $x \in (C \cap O) \setminus \mathcal{A}$ and $f \in F(x)$, $\langle \nabla V(x), f \rangle < 0$.

(L3') For each $x \in (D \cap O) \setminus \mathcal{A}$ and $g \in G(x)$,
 $V(g) - V(x) < 0$.

Theorem 9 Let $\mathcal{A} \subset O \subset \mathbb{R}^n$ with $\mathcal{A}$ compact and O open. Under Assumption 1, if there exists a strict Lyapunov function for $(\mathcal{A}, O)$, then $\mathcal{A}$ is uniformly asymptotically stable for (1). If there exists a proper strict Lyapunov function for $(\mathcal{A}, O)$, then $\mathcal{A}$ is uniformly asymptotically stable for (1) with basin of attraction containing O .

While a strict Lyapunov function can be difficult to find, and this fact has motivated other more sophisticated stability analysis tools that have appeared in the literature, it is reassuring to know that whenever $\mathcal{A}$ is compact and asymptotically stable, there exists a proper strict Lyapunov function for $(\mathcal{A}, \mathcal{B}_{\mathcal{A}})$.

Theorem 10 Under Assumption 1, if the compact set $\mathcal{A}$ is asymptotically stable for (1), then there exists a proper strict Lyapunov function for $(\mathcal{A}, \mathcal{B}_{\mathcal{A}})$. More specifically, for each $\lambda > 0$ there exists a smooth function V with $\text{dom } V = \mathcal{B}_{\mathcal{A}}$ that $V(x) = 0$ if and only if $x \in \mathcal{A}$, $\lim_{i \rightarrow \infty} V(x_i) = \infty$ when the sequence $\{x_i\}_{i=0}^{\infty}$, satisfying $x_i \in \mathcal{B}_{\mathcal{A}}$ for all $i \in \mathbb{Z}_{\geq 0}$, is unbounded or tends to the boundary of $\mathcal{B}_{\mathcal{A}}$, and such that:

1. For all $x \in C \cap \mathcal{B}_{\mathcal{A}}$ and $f \in F(x)$,
 $\langle \nabla V(x), f \rangle \leq -\lambda V(x)$.
2. For all $x \in D \cap \mathcal{B}_{\mathcal{A}}$ and $g \in G(x)$, $V(g) \leq \exp(-\lambda)V(x)$.

Summary and Future Directions

Under Assumption 1, stability theory for hybrid dynamical systems is very similar to stability theory for differential equations or difference equations with continuous right-hand sides. In particular, Lyapunov functions are a very common analysis tool for hybrid dynamical systems, though a Lyapunov function can be difficult to find in the same way that they are challenging to find for classical systems. With stability theory for hybrid dynamical systems firmly in place,

future research is expected to exploit this theory more fully for the development of control algorithms with new capabilities.

Cross-References

- Hybrid Dynamical Systems, Feedback Control of
- Lyapunov's Stability Theory

Bibliography

- Bainov DD, Simeonov PS (1989) Systems with impulse effect: stability, theory, and applications. Ellis Horwood Limited, Chichester
- Branicky MS (1998) Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. IEEE Trans Autom Control 43:1679–1684
- DeCarlo RA, Branicky MS, Pettersson S, Lennartson B (2000) Perspectives and results on the stability and stabilizability of hybrid systems. Proc IEEE 88(7): 1069–1082
- Goebel R, Sanfelice RG, Teel AR (2009) Hybrid dynamical systems. IEEE Control Syst Mag 29(2):28–93
- Goebel R, Sanfelice RG, Teel AR (2012) Hybrid dynamical systems. Princeton University Press, Princeton
- Haddad W, Chellaboina V, Nersisov SG (2006) Impulsive and hybrid dynamical systems. Princeton University Press, Princeton
- Hespanha JP (2004) Uniform stability of switched linear systems: extensions of LaSalle's invariance principle. IEEE Trans Autom Control 49(4):470–482
- Lakshmikantham V, Bainov DD, Simeonov PS (1989) Theory of impulsive differential equations. World Scientific, Singapore/Teaneck
- Liberzon D (2003) Switching in systems and control. Birkhauser, Boston
- Liberzon D, Morse AS (1999) Basic problems in stability and design of switched systems. IEEE Control Syst Mag 19(5):59–70
- Lygeros J, Johansson KH, Simić SN, Zhang J, Sastry SS (2003) Dynamical properties of hybrid automata. IEEE Trans Autom Control 48(1):2–17
- Matveev A, Savkin AV (2000) Qualitative theory of hybrid dynamical systems. Birkhauser, Boston
- Michel AN, Hou L, Liu D (2008) Stability of dynamical systems: continuous, discontinuous, and discrete systems. Birkhauser, Boston
- van der Schaft A, Schumacher H (2000) An introduction to hybrid dynamical systems. Springer, London/New York
- Yang T (2001) Impulsive control theory. Springer, Berlin/New York

Stability: Lyapunov, Linear Systems

Alessandro Astolfi

Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma Tor Vergata, Rome, Italy

Abstract

The notion of stability allows to study the qualitative behavior of dynamical systems. In particular it allows to study the behavior of trajectories close to an equilibrium point or to a motion. The notion of stability that we discuss has been introduced in 1882 by the Russian mathematician A.M. Lyapunov in his doctoral thesis; hence it is often referred to as Lyapunov stability. In this article we discuss and characterize Lyapunov stability for linear systems.

Keywords

Linear systems · Equilibrium point · Motion · Stability · Eigenvalues

Introduction

Consider a linear, time-invariant, finite-dimensional system, i.e., a system described by equations of the form

$$\begin{aligned}\sigma x &= Ax + Bu, \\ y &= Cx + Du,\end{aligned}\tag{1}$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$, $t \in T \subset \mathbb{R}$, and $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, and $D \in \mathbb{R}^{p \times m}$ constant matrices. In equation (1) $\sigma x(t)$ stands for $\dot{x}(t)$ if the system is continuous-time and for $x(t+1)$ if the system is discrete-time. Since the system is time-invariant,

it is assumed, without loss of generality, that all signals are defined for $t \geq 0$, that is, if the system is continuous-time then $T = \mathbb{R}^+$, i.e., the set of nonnegative real numbers, whereas if the system is discrete-time then $T = \mathbb{Z}^+$, i.e., the set of nonnegative integers. For ease of notation, the argument “ t ” is dropped whenever this does not cause confusion and we use the notation $t \geq 0$ to denote either $\mathbb{R}^+$ or $\mathbb{Z}^+$. Finally, we use either $x(t, x(0), u)$ or $x(t)$ to denote the solution of the first of equations (1) at a given time $t \geq 0$, with the initial condition $x(0)$ and the input signal u . The former is used when it is important to keep track of the initial state and external input u , whereas the latter is used whenever there is not such a need.

Definition 1 (Equilibrium) Consider the system (1). Assume the input u is constant, i.e., $u(t) = u_0$ for all $t \geq 0$ and for some constant u_0 . A state x_e is an equilibrium of the system associated to the input u_0 if $x_e = x(t, x_e, u_0)$, for all $t \geq 0$.

Proposition 1 (Equilibria of linear systems) Consider the system (1) and assume $u(t) = u_0$, for all t , where u_0 is a constant vector. Then the following holds

- If $u_0 = 0$ then the origin is an equilibrium.
- For continuous-time systems, if A is invertible, for any u_0 there is a unique equilibrium $x_e = -A^{-1}Bu_0$. If A is not invertible, the system has either infinitely many equilibria or it has no equilibria.
- For discrete-time systems, if $I - A$ is invertible, for any u_0 there is a unique equilibrium $x_e = (I - A)^{-1}Bu_0$. If $I - A$ is not invertible the system has either infinitely many equilibria or it has no equilibria.

Proposition 2 Consider the continuous-time, time-invariant, linear system described by the equations

$$\begin{aligned}\dot{x} &= Ax + Bu, \\ y &= Cx + Du,\end{aligned}$$

and the initial condition $x(0) = x_0$. Then, for all $t \geq 0$,

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau \quad (2)$$

and

$$y(t) = Ce^{At}x_0 + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t). \quad (3)$$

Proposition 3 Consider the discrete-time, time-invariant, linear system described by the equations (to simplify notation we use $x^+(t)$ to denote $x(t+1)$ and we drop the argument t)

$$\begin{aligned} x^+ &= Ax + Bu, \\ y &= Cx + Du, \end{aligned}$$

and the initial condition $x(0) = x_0$. Then, for all $t \geq 0$,

$$x(t) = A^t x_0 + \sum_{i=0}^{t-1} A^{t-1-i} Bu(i) \quad (4)$$

and

$$y(t) = CA^t x_0 + \sum_{i=0}^{t-1} CA^{t-1-i} Bu(i) + Du(t). \quad (5)$$

Definitions

In this section we provide some notions and definitions which are applicable to general dynamical systems, despite being formulated with reference to the system (1).

Definition 2 (Lyapunov stability) Consider the system (1) with $u(t) = u_0$, for all $t \geq 0$ and for some constant u_0 . Let x_e be an equilibrium point. The equilibrium is stable (in the sense of Lyapunov) if for every $\epsilon > 0$ there exists a $\delta = \delta(\epsilon) > 0$ such that (the notation $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^n$) $\|x(0) - x_e\| < \delta$ implies $\|x(t) - x_e\| < \epsilon$, for all $t \geq 0$.

In stability theory the quantity $x(0) - x_e$ is called initial perturbation, and $x(t)$ is called perturbed evolution. Therefore, the definition

of stability can be interpreted as follows. An equilibrium point x_e is stable if however we select a *tolerable* deviation ϵ , there exists a (possibly small) neighborhood of the equilibrium x_e such that all initial conditions in this neighborhood yield trajectories which are within the *tolerable* deviation.

The property of stability dictates a condition on the evolution of the system for all $t \geq 0$. Note, however, that in the definition of stability we have not requested that the perturbed evolution converge asymptotically, that is, for $t \rightarrow \infty$, to x_e . This convergence property is very important in applications, as it allows to characterize the situation in which not only the perturbed evolution remains close to the unperturbed evolution, but it also converges to the initial (unperturbed) evolution. To capture this property we introduce a new definition.

Definition 3 (Asymptotic stability) Consider the system (1) with $u(t) = u_0$, for all $t \geq 0$ and for some constant u_0 . Let x_e be an equilibrium point. The equilibrium is asymptotically stable if it is stable and if there exists a constant $\delta_a > 0$ such that $\|x(0) - x_e\| < \delta_a$ implies $\lim_{t \rightarrow \infty} \|x(t) - x_e\| = 0$.

In summary, an equilibrium point is asymptotically stable if it is stable and, whenever the initial perturbation is inside a certain neighborhood of x_e , the perturbed evolution converges, asymptotically, to the equilibrium point, which is thus said to be attractive. From a physical point of view this means that all sufficiently small initial perturbations give rise to effects which can be a priori bounded (stability) and which vanish asymptotically (attractivity).

It is important to highlight that, in general, attractivity does not imply stability: it is possible to have an equilibrium of a system which is not stable (i.e., it is unstable), yet for all initial perturbations the perturbed evolution converges to the equilibrium. This however is not the case for linear systems, as discussed in the section “[Stability of Linear Systems](#)”. We conclude the section with two simple examples illustrating the notions that have been introduced.

Example 1 Consider the discrete-time system $x^+ = -x$, with $x(t) \in \mathbb{R}$. This system has a unique equilibrium at $x_e = 0$. Note that for any initial condition $x_0 \in \mathbb{R}$ one has

$$x(2t - 1) = -x_0, \quad x(2t) = x_0,$$

for all $t \geq 1$ and integer. This implies that the equilibrium is stable, but not attractive.

Example 2 Consider the continuous-time system

$$\dot{x}_1 = \omega x_2, \quad \dot{x}_2 = -\omega x_1,$$

with ω a positive constant. The system has a unique equilibrium at $x_e = 0$. This equilibrium is stable, but not attractive. To see this note that, along the trajectories of the system, $x_1 \dot{x}_1 + x_2 \dot{x}_2 = 0$, and this implies that, along the trajectories of the system, $x_1^2(t) + x_2^2(t)$ is constant, i.e., $x_1^2(t) + x_2^2(t) = x_1^2(0) + x_2^2(0)$. Therefore the state of the system remains on the circle centered at the origin and with radius $\sqrt{x_1^2(0) + x_2^2(0)}$ for all $t \geq 0$: the condition for stability holds with $\delta(\epsilon) = \epsilon$.

Definition 4 (Global asymptotic stability)

Consider the system (1) with $u(t) = u_0$, for all $t \geq 0$ and for some constant u_0 . Let x_e be an equilibrium point. The equilibrium is globally asymptotically stable if it is stable and if, for all $x(0)$, $\lim_{t \rightarrow \infty} \|x(t) - x_e\| = 0$.

The property of (global) asymptotic stability can be strengthened imposing conditions on the convergence speed of $\|x(t) - x_e\|$.

Definition 5 (Exponential stability)

Consider the system (1) with $u(t) = u_0$, for all $t \geq 0$ and for some constant u_0 . Let x_e be an equilibrium point. The equilibrium is exponentially stable if there exists $\lambda > 0$, in the case of continuous-time systems, and $0 < \lambda < 1$, in the case of discrete-time systems, such that for all $\epsilon > 0$ there exists a $\delta = \delta(\epsilon) > 0$ such that $\|x(0) - x_e\| < \delta$ implies $\|x(t) - x_e\| < \epsilon e^{-\lambda t}$, in the case of continuous-time systems, and $\|x(t) - x_e\| < \epsilon \lambda^t$, in the case of discrete-time systems, for all $t \geq 0$.

Definition 6 (Stability of motion) Consider the system (1). Let

$$\mathcal{M} = \{(t, x(t)) \in T \times \mathbb{R}^n\},$$

with $x(t) = x(t, x_0, u)$, for given x_0 and u , and $T = \mathbb{R}^+$, in the case of continuous-time systems, and $T = \mathbb{Z}^+$, in the case of discrete-time systems, be a motion. The motion is stable if for every $\epsilon > 0$ there exists a $\delta = \delta(\epsilon) > 0$ such that $\|x(0) - x_0\| < \delta$ implies

$$\|x(t, x(0), u) - x(t, x_0, u)\| < \epsilon, \quad (6)$$

for all $t \geq 0$.

The notion of stability of a motion is substantially the same as the notion of stability of an equilibrium. The important issue is that the time parametrization is important, i.e., a motion is stable if, for small initial perturbations, the perturbed evolution is close, for any fixed $t \geq 0$, to the unperturbed evolution. This does not mean that if the perturbed and unperturbed trajectories are close, then the motion is stable: in fact the trajectories may be close but may be followed with different timing, which means that for some $t \geq 0$ condition (6) may be violated.

Stability of Linear Systems

The notion of stability relies on the knowledge of the trajectories of the system. As a result, even if this notion is very elegant, and useful in applications, it is in general hard to assess stability of an equilibrium or of a motion. There are, however, classes of systems for which it is possible to give stability conditions without relying upon the knowledge of the trajectories. Linear systems belong to one such class. In this section we study the stability properties of linear systems and we show that, because of the linear structure, it is possible to assess the properties of stability and attractivity in a simple way. To begin with, we recall some properties of linear systems.

Proposition 4 Consider a linear, time-invariant system. (Asymptotic) stability of one motion

implies (asymptotic) stability of all motions. In particular, (asymptotic) stability of any motion implies and is implied by (asymptotic) stability of the equilibrium $x_e = 0$.

The above statement, together with the result in Proposition 1, implies the following important properties.

Proposition 5 *If the origin of a linear system is asymptotically stable then the origin is the only equilibrium of the system for $u = 0$. Moreover, asymptotic stability of the zero equilibrium is always global. Finally, asymptotic stability implies exponential stability.*

The above discussion shows that the stability properties of a motion (e.g., an equilibrium) of a linear system are inherited by all motions of the system. Moreover, for linear systems, local properties are always global properties. This means that, with some abuse of terminology, we can refer the stability properties to the linear system, for example we say that a linear system is stable to mean that all its motions are stable. Stability properties of a linear, time-invariant, system are therefore properties of the *free* evolution of its state: for this class of systems it is possible to obtain simple stability tests.

Proposition 6 *A linear, time-invariant, system is stable if and only if $\|e^{At}\| \leq k$, for continuous-time systems, or $\|A^t\| \leq k$, for discrete-time systems, for all $t \geq 0$ and for some $k > 0$. It is asymptotically stable if and only if $\lim_{t \rightarrow \infty} e^{At} = 0$, for continuous-time systems, or $\lim_{t \rightarrow \infty} A^t = 0$, for discrete-time systems.*

Proposition 7 *The equilibrium $x_e = 0$ of a linear, time-invariant, system is stable if and only if the following conditions hold*

- *In the case of continuous-time systems, the eigenvalues of A with geometric multiplicity equal to one have nonpositive real part, and the eigenvalues of A with geometric multiplicity larger than one have negative real part.*
- *In the case of discrete-time systems, the eigenvalues of A with geometric multiplicity equal to one have modulo not larger than one, and*

the eigenvalues of A with geometric multiplicity larger than one have modulo smaller than one.

Remark 1 To define the geometric multiplicity of an eigenvalue, we need to recall a few facts. Consider a matrix $A \in \mathbb{R}^{n \times n}$ and a polynomial $p(\lambda)$. The polynomial $p(\lambda)$ is a zeroing polynomial for A if $p(A) = 0$. Note that, by Cayley-Hamilton Theorem, the characteristic polynomial of A is a zeroing polynomial for A . Among all zeroing polynomials, there is a unique monic polynomial $p_M(\lambda)$ with smallest degree. This polynomial is called the minimal polynomial of A . Note that the minimal polynomial of A is a divisor of the characteristic polynomial of A . If A has $r \leq n$ distinct eigenvalues $\lambda_1, \dots, \lambda_r$ then

$$p_M(\lambda) = (\lambda - \lambda_1)^{m_1} (\lambda - \lambda_2)^{m_2} \dots (\lambda - \lambda_r)^{m_r},$$

where the number m_i denotes, by definition, the geometric multiplicity of λ_i , for $i = 1, \dots, r$. This means that the geometric multiplicity of λ_i equals the multiplicity of λ_i as a root of $p_M(\lambda)$. Recall, finally, that the multiplicity of λ_i as a root of the characteristic polynomial is called algebraic multiplicity.

Proof. Let $\lambda_1, \lambda_2, \dots, \lambda_r$, with $r \geq 1$, be the distinct eigenvalues of A , i.e., the distinct roots of the characteristic polynomial of A . Then

$$e^{At} = \sum_{i=1}^r \sum_{k=1}^{m_i} R_{ik} \frac{t^{k-1}}{(k-1)!} e^{\lambda_i t},$$

for some matrices R_{ik} , where m_i is the geometric multiplicity of the eigenvalue λ_i . This matrix is bounded if and only if the conditions in the statement hold. Similarly,

$$A^t = \sum_{i=1}^r \sum_{k=1}^{m_i} R_{ik} \frac{t^{k-1}}{(k-1)!} \lambda_i^{t-k+1},$$

for some matrices R_{ik} , and this is bounded if and only if the conditions in the statement hold. $\triangleleft$

Proposition 8 *The equilibrium $x_e = 0$ of a linear, time-invariant, system is asymptotically stable if and only if the following conditions hold*

- *In the case of continuous-time systems, the eigenvalues of A have negative real part.*
- *In the case of discrete-time systems, the eigenvalues of A have modulo smaller than one.*

Proof. The proof is similar to the one of the previous proposition, once it is noted that, for the considered class of systems, and as stated in Proposition 6, asymptotic stability implies and is implied by boundedness and convergence of e^{At} or A^t . $\triangleleft$

Remark 2 For linear, time-varying, systems, i.e., systems described by equations of the form

$$\begin{aligned}\dot{x} &= A(t)x + B(t)u, \\ y &= C(t)x + D(t)u,\end{aligned}$$

it is possible to provide stability conditions in the spirit of the boundedness and convergence conditions in Proposition 6. These require the definition of a matrix, the so-called state transition matrix, which describes the free evolution of the state of the system. It is, however, not possible to provide conditions in terms of eigenvalues of the matrix $A(t)$, for each t , similar to the conditions in Propositions 7 and 8.

We conclude this discussion with an alternative characterization of asymptotic stability in terms of linear matrix inequalities.

Proposition 9 *The equilibrium $x_e = 0$ of a linear, time-invariant, system is asymptotically stable if and only if the following conditions hold*

- *In the case of continuous-time systems, there exists a positive definite matrix $P = P'$ such that $A'P + PA < 0$.*

- *In the case of discrete-time systems, there exists a positive definite matrix $P = P'$ such that $A'PA - P < 0$.*

Remark 3 P' denotes the transpose of the matrix P . A square and symmetric matrix $P \in \mathbb{R}^{n \times n}$ is positive definite, denoted $P > 0$, if $v'Pv > 0$, for all nonzero vectors $v \in \mathbb{R}^n$. A matrix P is negative definite, denoted $P < 0$, if $-P > 0$.

To complete our discussion we stress that stability properties are invariant with respect to changes in coordinates in the state space.

Corollary 1 *Consider a linear, time-invariant, system and assume it is (asymptotically) stable. Then any representation obtained by means of a change of coordinates of the form $x(t) = L\hat{x}(t)$, with L constant and invertible, is (asymptotically) stable.*

Proof. The proof is based on the observation that the change of coordinates transforms the matrix A into $\tilde{A} = L^{-1}AL$ and that the matrices A and $\tilde{A}$ are similar, that is, they have the same characteristic and minimal polynomials. $\triangleleft$

Summary and Future Directions

The property of Lyapunov stability is instrumental to characterize the qualitative behavior of dynamical systems. For linear, time-invariant, systems this property can be studied on the basis of the location, and multiplicity, of the eigenvalues of the matrix A . The property of Lyapunov stability can be studied for more general classes of systems, including nonlinear systems, distributed parameter systems, and hybrid systems, to which the basic definitions given in this article apply.

Cross-References

- [Feedback Stabilization of Nonlinear Systems](#)
- [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)

- ▶ Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions
- ▶ Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions
- ▶ Linear Systems: Discrete-Time, Time-Varying, State Variable Descriptions
- ▶ Lyapunov Methods in Power System Stability
- ▶ Lyapunov's Stability Theory
- ▶ Power System Voltage Stability
- ▶ Small Signal Stability in Electric Power Systems
- ▶ Stability and Performance of Complex Systems Affected by Parametric Uncertainty

Bibliography

- Antsaklis PJ, Michel AN (2007) A linear systems primer. Birkhäuser, Boston
- Brockett RW (1970) Finite dimensional linear systems. Wiley, London
- Hahn W (1967) Stability of motion. Springer, New York
- Khalil HK (2002) Nonlinear systems, 3rd edn. Prentice-Hall, Upper Saddle River
- Lyapunov AM (1992) The general problem of the stability of motion. Taylor & Francis, London
- Trentelman HL, Stoorvogel AA, Hautus MLJ (2001) Control theory for linear systems. Springer, London
- Zadeh LA, Desoer CA (1963) Linear system theory. McGraw-Hill, New York

State Estimation for Batch Processes

Wolfgang Mauntz
Fakultät Bio- und Chemieingenieurwesen,
Technische Universität Dortmund, Dortmund,
Germany

Abstract

The information about certain safety or quality parameters during a batch process is valuable for a variety of reasons. In case a direct measurement is too expensive, too slow, or nonexisting, a state estimator estimating the desired

quantities based on a model and various other measurements may be a good alternative. The most prominent method is calorimetry, where the heat of reaction is measured. This article gives an overview of different alternatives that support a safe and successful batch operation.

Keywords

Observer · Soft sensor · State estimator · Calorimetry

Introduction

Continuous processes are used to produce a product at a constant rate. They are designed to operate at constant conditions, i.e., the state of the process (conversion, temperatures, pressures, concentrations, etc.) does not vary. In contrast, (semi-) batch processes execute a recipe which means that they are typically operated within a wide range of states. The state of the (semi-batch) process should constantly be monitored. This information is useful for several purposes:

- *Process safety*: abnormal process states such as the accumulation of hazardous substances or reactive materials may lead to dangerous situations such as runaway reactions. The earlier an abnormal state is detected, the better it can be corrected, and the higher is the probability that loss can be avoided.
- *Quality*: if the batch is not operated along the standard trajectory, off-spec product may result which in turn results in extra effort and/or second grade product if this is discovered in time and in a customer complaint if not discovered before delivery.
- *Profit*: the better the state is known, the less conservative the underlying control scheme needs to be, and the more the process can be pushed to its limits. This may lead to a higher throughput, less by-products, or less energy consumption. Advanced control schemes which are typically applied for this purpose require knowledge of the state of the process.

The literature offers a wide range of ways to monitor a batch process. In some processes, the observation of simple measurements like temperatures, pressures, and the time that a process step takes for execution is sufficient to guarantee for safe standard product in minimum time. Examples include some melt polymerizations.

However, as soon as the process is more complex, more information than just temperatures and pressures is required to monitor the process to meet the goals mentioned above. It may be sufficient to measure other easy-to-measure properties like conductivities, flowrates, pH values, sound velocities, attenuations, etc.; however in many cases these measurements do not give the complete state of the system. Properties like complex gas phase compositions cannot be measured this way. This might require the installation of more sophisticated measurements, e.g., NIR spectroscopy, online gas chromatography, Raman spectroscopy, or ion mobility spectroscopy. These measurements require significant effort in terms of installation cost and maintenance. In other situations, no online measurement may be available at all. These cases include the measurement of the distribution of the molecular weight in a polymer melt.

In these cases, where direct online measurements are either too expensive or not available at all, several methods are available to obtain information on the status of the batch Estimation in Systems and Control.

- *Statistical Methods*

Experiences from historical batches are used in a statistical way to predict whether a batch runs normally. This can, e.g., be accomplished by defining a golden batch and a corresponding corridor around these trajectories. More sophisticated methods use principal component analysis (PCA) or partial least squares (PLS) to get a hint at abnormal situations. These methods are even capable of pointing at the origin of a possible problem. They are restricted to problem detection and typically cannot be used for control purposes.

- *Model-Based State Estimation*

The state of the system (temperatures, pressures, concentrations, etc.) is estimated online which allows for problem detection as well as control applications. This method will be described in more detail in section [Model-Based State Estimation](#)

General reviews of state estimation techniques can be found in Besancon (2007) and Schei (2008), and a review of industrial applications is, e.g., given in Fortuna et al. (2007).

Model-Based State Estimation

The basic idea of a state estimator (which is frequently also called *observer* or *soft sensor*) is to run a mathematical model of the process in parallel to the process itself, to compare the available measurements to the values which are predicted by the model, and to correct the estimated state by a suitable function of the observed error, usually an additive correction term that depends on the error. For a state estimator to converge to the true state, the considered system needs to be observable. For details see ► [“Controllability and Observability”](#). The scheme of a state estimator is sketched in Fig. 1. The real system processes the input $\mathbf{u}$ to give the system state $\mathbf{x}$ which is affected by the system noise ξ . The measurements $\mathbf{y}$ are perturbed by the measurement noise φ . The model predicts a system state $\hat{\mathbf{x}}$ and a measurement $\hat{\mathbf{y}}$. The difference between the measured value $\mathbf{y}$ and predicted value $\hat{\mathbf{y}}$ is then fed back to correct the estimated state.

For **linear systems**, the most commonly used state estimators are the *Luenberger Observer* and the *Kalman filter* ► [“Kalman Filters”](#). Both multiply the prediction error ($\mathbf{y} - \hat{\mathbf{y}}$) by a weighting matrix $\mathbf{K}$ to update the estimated state $\hat{\mathbf{x}}$:

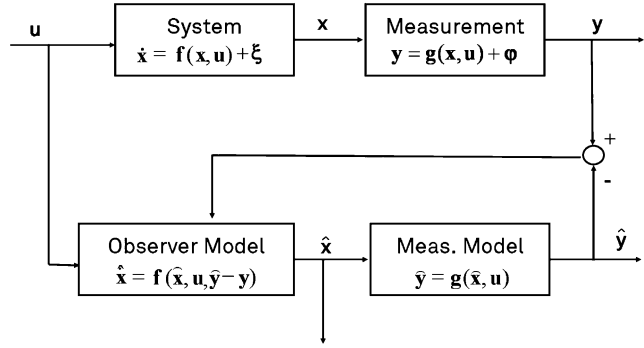
$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}(\mathbf{y} - \hat{\mathbf{y}})$$

The two techniques use different approaches for determining the matrix $\mathbf{K}$:

Luenberger Observer The basic assumption is that the deviation $\mathbf{e}(t)$ between $\mathbf{x}$ and $\hat{\mathbf{x}}$ is due

State Estimation for Batch Processes, Fig. 1

Principle of a state estimator



to wrong initial values $\hat{\mathbf{x}}_0$. $\mathbf{K}$ is computed by choosing the desired speed of convergence of the error

$$\begin{aligned}\dot{\mathbf{e}}(t) &= \dot{\mathbf{x}}(t) - \dot{\hat{\mathbf{x}}}(t) \\ &= (\mathbf{A} - \mathbf{K}\mathbf{C}) \mathbf{e}(t).\end{aligned}$$

to zero. This is done by placing the eigenvalues of the matrix $(\mathbf{A} - \mathbf{K}\mathbf{C})$ in the left half plane.

Kalman filter The basic assumption is that the error $\mathbf{e}(t)$ is caused by white noise in the system ξ as well as in the measurement φ . The idea is to minimize the expectation of the quadratic error

$$\min_{\hat{\mathbf{x}}} E \left((\hat{\mathbf{x}}(t) - \mathbf{x}(t))^T (\hat{\mathbf{x}}(t) - \mathbf{x}(t)) \right).$$

$\mathbf{K}$ is computed from the noise covariance matrices and the system dynamics and varies with time.

The tuning of the state estimators is not trivial. The larger the absolute value of the eigenvalues in the Luenberger approach, the faster the error will converge to zero, but the more prone the state estimator will be to measurement noise. A similar trade-off exists for the Kalman filter where the covariance matrices of the noise terms ξ and φ

and the covariance of the initial state ξ_0 need to be defined.

For **nonlinear systems**, a variety of approaches is available. The most frequently used estimators are based on using the nonlinear model for the prediction of the state, and linearizations of the system dynamics are used to update the matrix $\mathbf{K}$. The *extended Kalman filter (EKF)* ▶ “[Extended Kalman Filters](#)” and the *extended Luenberger Observer (ELO)* are representatives of this class of approaches. The EKF is most widely used. Extensions are the *constrained EKF* and the *unscented EKF*.

As examples are known where the EKF fails due the nonlinearity of the system, methods based on ideas other than the linearization of system dynamics have been developed. These methods include the *moving horizon estimator (MHE)* ▶ “[Moving Horizon Estimation](#)” and the *particle filter*. Because of the increasing capabilities of modern computers and significant improvements in dynamic optimization algorithms, the MHE is a very promising alternative. The idea of the method is to minimize the sum of the squared errors of the system noise ξ_l , the measurement noise φ_l , and the error of the initial state ξ_{k-N} which are weighted by weighing matrices $\mathbf{P}_k$, $\mathbf{Q}$, and $\mathbf{R}$ over a pre-defined horizon of past sampling steps $k - N, \dots, k$

$$\min_{\xi_i, \varphi_j} \xi_{k-N}^T \mathbf{P}_k^{-1} \xi_{k-N} + \sum_{l=k-N+1}^{k-1} \xi_l^T \mathbf{Q}^{-1} \xi_l + \sum_{l=k-N+1}^k \varphi_l^T \mathbf{R}^{-1} \varphi_l$$

s.t. the system model and the measurement equations are satisfied

and further inequality constraints (e.g., physical limits of variables) hold.

The possibility to define constraints on the estimated states, e.g., that concentrations must be nonnegative, is an important advantage of the MHE approach. If the horizon is reduced to one single measurement, the constrained extended Kalman filter results which combines the simplicity of the EKF with the possibility to include constraints on the estimated states. Efficient implementations of the MHE have led to the method being capable of estimating the state of rather large systems in real time (Diehl et al. 2006; Küpper and Engell 2007).

Calorimetry

Temperature measurements are probably the cheapest available measurements in chemical processes, and most plants are typically well equipped with temperature sensors. To exploit temperature measurements, e.g., for the observation of exothermic or endothermic reactions, heat balances are set up and solved for the heat of reaction which then enables the computation of the reaction rate. This is typically referred to as **calorimetry**. Reviews are given, e.g., in Hergeth (2006), McKenna et al. (2000), and Landau (1996). For a jacketed reactor, the heat balance around a semi-batch reactor typically reads (see also Fig. 2)

$$C_{P,R} \frac{dT_R}{dt} = \dot{Q}_R + kA(T_J - T_R) + \sum_i \dot{m}_{F,i} c_{p,Fi} (T_{F,i} - T_R), \quad (1)$$

where $\dot{Q}_R$ represents the heat of reaction, kA is the overall heat transfer coefficient between the reactor content and the jacket, T_R is the reactor temperature, T_J is the jacket temperature, T_F is the feed temperature, $C_{P,R}$ is the overall heat capacity of the reactor, and the last term on the right side is the enthalpy added by the feed to the reactor. If kA is known, $\dot{Q}_R$ can directly be computed as all other quantities in Eq. (1) are known or measured. This is referred to as *heat flow calorimetry*.

In industrial practice, kA usually is not known and varies over time due to changes of the filling level, changes of the viscosity of the reaction mixture and fouling. Then other heat balances and measurements can be added to enable a direct computation or estimation of kA . Typically, the jacket heat balance is chosen

$$C_{P,J} \frac{dT_J}{dt} = kA(T_R - T_J) + kA_{\text{jack}}(T_{\text{env}} - T_J) + \dot{m}_J c_{p,J} (T_{J,\text{in}} - T_J). \quad (2)$$

If necessary, also other phenomena like direct heat losses from the reactor content to the environment or the influence of the reactor lid can be taken into account by adding additional terms or additional heat balances. This method is called *heat balance calorimetry*.

In order to compute $\dot{Q}_R$ and kA from Eqs. (1) and (2), two different approaches can be used:

1. Equations (1) and (2) are solved to give

$$\hat{kA} = \frac{C_{P,J} \frac{dT_J}{dt} - kA_{\text{jack}}(T_{\text{env}} - T_J) - \dot{m}_J c_{p,J} (T_{J,\text{in}} - T_J)}{T_R - T_J} \quad (3a)$$

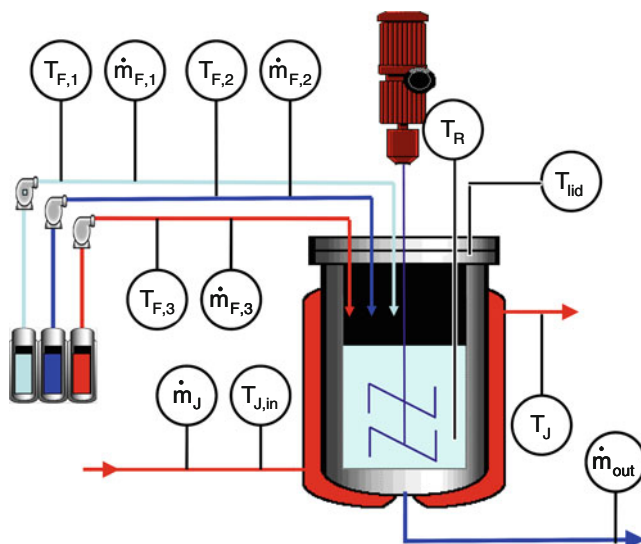
$$\hat{\dot{Q}}_R = C_{P,R} \frac{dT_R}{dt} - kA(T_J - T_R) - \sum_i \dot{m}_{F,i} c_{p,Fi} (T_{F,i} - T_R). \quad (3b)$$

In this approach, the derivatives need to be computed from the measurements which introduce noise in the evaluation and require a filtering either of the derivatives or of the estimates.

2. Equations (1) and (2) are implemented in a nonlinear state estimator. To estimate the unknown quantities $\hat{kA}$ and $\hat{\dot{Q}}_R$ by this approach, additional assumptions about their dynamics must be made. A common approach

State Estimation for Batch Processes, Fig. 2

The reactor and its jacket as considered for calorimetry



is to add the so-called dummy derivatives

$$\frac{d \hat{Q}_R}{dt} = 0$$

$$\frac{d \hat{k}A}{dt} = 0.$$

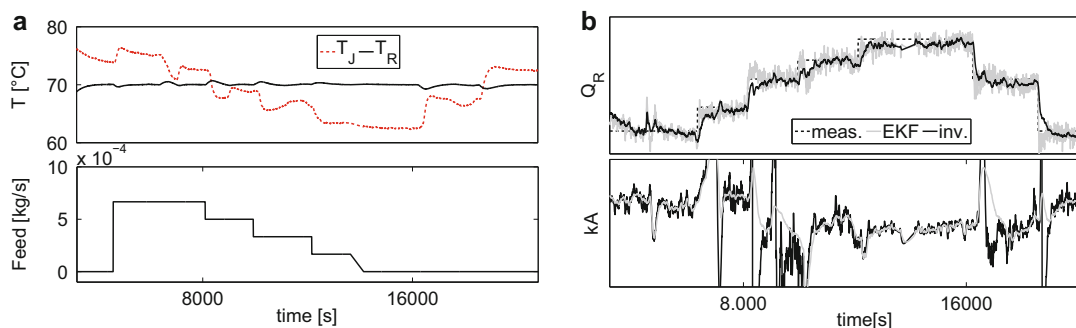
The tuning of calorimetric estimation schemes has been discussed in the literature, but for each case, tests in simulation runs using recorded batch data should be performed.

Experimental results of the application of the direct solution Eq. (3) and an EKF for the estimation of $\dot{Q}_R$ and kA are shown in Fig. 3. A laboratory scale 10 l metal reactor was filled with water. Cold water was injected into the reactor to simulate the feed of reactants. The reactor is equipped with a heating rod by which different values of $\dot{Q}_R$ could be simulated. Figure 3a shows the measured temperatures and the feed stream; Fig. 3b shows the estimates. The dotted line displays the measured power uptake, the thin, black line represents the estimates from the evaluation of Eq. (3), and the gray line shows the results obtained with an EKF. The EKF was tuned slightly more aggressively than the PT1 filter that was used to filter the

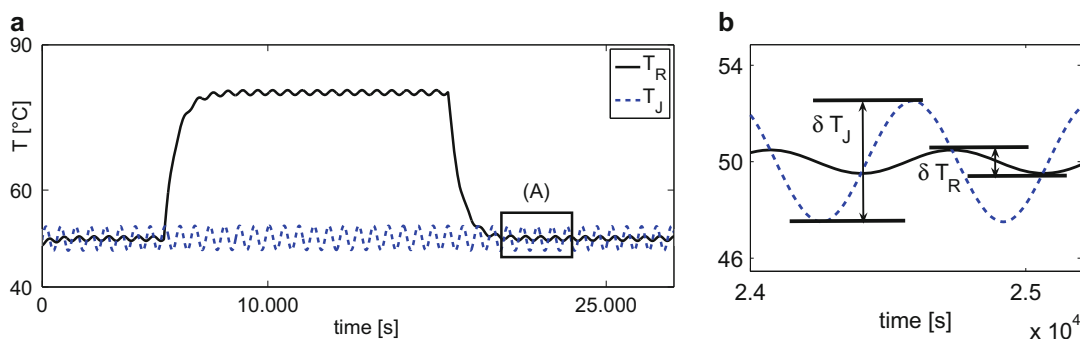
values of $\hat{Q}_R$ and $\hat{k}A$ that were obtained from Eqs. (3).

It can be seen that the quality of both evaluation methods is comparable. A difference in performance can be seen in the estimation of kA at the points in time where $T_R \approx T_J$. This is due to the denominator in Eq. (3a) which becomes ≈ 0 . At this point, kA is unobservable. The EKF estimates of kA are more smooth. This does not have an impact on the estimation of $\dot{Q}_R$ because the heat transfer from the jacket to the reactor is zero at this point. This behavior is of importance if $\hat{k}A$ is used in other algorithms, e.g., for control purposes.

A practical problem is the determination of the parameters of the system model. Especially the heat capacity of the reactor $C_{P,R}$ is difficult to determine as it is not clear how much impact the reactor material has. Also the heat capacities of intermediate products and mixtures with the raw materials and final products may not be known. That is why typically $C_{P,R}$ is considered a "free" parameter which is used to fit the estimates to measured data. If the adjustment of the available parameters is not sufficient to yield a satisfactory performance of the estimator, further extensions can be considered:



State Estimation for Batch Processes, Fig. 3 Illustration of the results of direct estimation (Eq. 3) and the use of an EKF. (a) Measured data. (b) Estimates from inverted equations (3) and EKF as well as measured $\hat{Q}_R$



State Estimation for Batch Processes, Fig. 4 Example experiment where TOC is applied. (a) Complete example. (b) Zoom of rectangle (A)

- If pressurized vessels are considered, the wall thickness may be considerable, and the heat accumulation may influence the results. In this case, the extension of the set of equations by an equation for the heat transfer through the wall may be considered (Saenz de Buruaga et al. 1997).
- If large-scale vessels are considered, the cooling fluid in the jacket may not be perfectly mixed, and a temperature gradient will be present. In many cases, cooling coils are welded on the outside surface of the reactor. In this case, the equation for the perfectly mixed jacket (Eq. 2) should be replaced by a model for a plug flow reactor (Krämer and Gesthuisen 2005).
- For large industrial reactors, the perfect mixing assumption of the reactor contents does

not necessarily hold true. Especially if polymerization reactions are considered, the reactor content may become rather viscous. A straightforward method to cope with this problem is a detailed computational fluid dynamics (CFD) simulation. However, due to the numerical complexity, this appears infeasible for online applications. A practical alternative is the placement of several temperature sensors and using a weighted average over their readings. A different approach is the usage of a multi-zonal model, the idea of which resembles the idea of a CFD model; however the number of zones (elements) is much smaller (Bezzo et al. 2004).

Heat balance calorimetry becomes inaccurate if the mass flow through the jacket is so large that

the temperature difference between the cooling stream entering the jacket and leaving the jacket ($T_{J,in} - T_J$) is in the order of magnitude of the measurement error. This mode of operation is typically used in laboratory scale reactors to avoid temperature gradients in the jacket. To estimate the states in such setups, a technique called *temperature oscillation calorimetry* (TOC) can be used. The idea is to add a small but well measurable sinusoidal signal to the typically constant setpoint of the reactor temperature T_R (see Fig. 4 for an example). The reaction of the jacket temperature to the oscillating reactor temperature can be used to compute kA , e.g., by estimating its amplitude δT_J (Tietze et al. 1996) or by adding an additional equation which describes the second derivative of the reactor temperature $\frac{d^2 T_R}{dt^2}$ to the set of heat balances (Mauntz et al. 2007).

Calorimetry estimates the total heat of the reactions in the reactor. It can be used to estimate the overall chemical conversion of a process. Due to its integral character, the heat of reaction of parallel and consecutive reactions cannot be estimated separately (Hergeth 2006). However, if models of the chemical kinetics are known and reliable, it is possible to couple this kinetic model with calorimetry and to observe the complete state of the reaction based on calorimetric estimates. This solution may however not be robust as slight errors in the kinetic model may lead to significant errors in the estimates of all concentrations. In order to build a more robust state estimator, additional measurements should be installed and integrated into the state estimator. For example, for reactions including a phase change from the gas phase to the liquid phase, a pressure measurement may be suitable. For some polymerization reactions, sound velocity and sound attenuation measurements can be valuable (Brandt et al. 2012). The additional measurement can be incorporated into the observation scheme by augmenting the measurement model $\mathbf{g}$ (see Fig. 1) by the corresponding measurement equation.

Summary

In this contribution, different methods that can be used to determine the states of (semi-) batch reactions have been described. State estimation is useful to reconcile measurement errors and whenever direct online measurements are either too expensive or not available at all.

Linear state estimation is a mature topic. However as chemical batch reactors in most cases have nonlinear dynamics, nonlinear methods should be applied. Extensions of linear state estimators based on linearizations of the system (e.g., the EKF) are the most widely used nonlinear state estimators. However examples are known where these estimators fail. Thus, other approaches, e.g., based on online optimization (MHE), have been developed. They deliver promising results in terms of observation quality and computational speed even for large-scale systems.

The most widespread application of state estimation techniques in batch processes is calorimetry which is suitable for significantly exothermic or endothermic reactions. The heat balances around the reactor contents and the jacket are set up and solved. The estimated heat of reaction is used to estimate the chemical conversion of the process. The method makes use of commonly installed temperature measurements in the reactor. Extensions to include other measurements have been discussed. Problems that typically occur in laboratory scale reactors can be overcome with the help of temperature oscillation calorimetry.

Cross-References

- [Control and Optimization of Batch Processes](#)
- [Kalman Filters](#)
- [Moving Horizon Estimation](#)
- [Networked Control Systems: Estimation and Control over Lossy Networks](#)
- [Observers in Linear Systems Theory](#)

Bibliography

- Besancon G (2007) Nonlinear observers and applications. Springer, Berlin/New York
- Bezzo F, Macchietto S, Pantelides CC (2004) A general methodology for hybrid multizonal/CFD models, part I. Theoretical framework AND part II. Automatic zoning. *Comput Chem Eng* 28: 501–525
- Brandt H, Sühling D, Engell S (2012) Monitoring emulsion polymerization processes by means of ultra-sound velocity measurements. In: AICHE annual meeting, Pittsburgh, 28 Oct – 2 Nov
- Diehl M, Kühl P, Bock HG, Schlöder JP, Mahn B, Kallrath J (2006) Combined nonlinear MPC and MHE for a copolymerization process. In: 16th European symposium on computer aided process engineering, Garmisch- Patenkirchen, 10–13 July, pp 1527–1532
- Fortuna L, Graziani S, Rizzo A, Xibilia MG (2007) Soft sensors for monitoring and control of industrial processes. Springer, London
- Hergeth WD (2006) On-line monitoring of chemical reactions. In: Ullmann's encyclopedia of industrial chemistry, 7th online edn. Wiley-VCH, Weinheim
- Küpper A, Engell S (2007) Optimizing control of the Hashimoto SMB process: experimental application. In: 8th international IFAC symposium on dynamics and control of process control, 6–8 June 2007, Cancun
- Krämer S, Gesthuisen R (2005) Simultaneous estimation of the heat of reaction and the heat transfer coefficient by calorimetry: estimation problems due to model simplification and high jacket flow rates – theoretical development. *Chem Eng Sci* 60: 4233–4248
- Landau RN (1996) Expanding the role of reaction calorimetry. *Thermochim Acta* 289:101–126
- Mauntz W, Diehl M, Engell S (2007) Moving horizon estimation and optimal excitation in temperature oscillation calorimetry. In: DYCOPS, 6.-8.6.2007, Cancun
- McKenna TF, Othman S, Févotte G, Santos AM, Ham-mouri H (2000) An integrated approach to polymer reaction engineering: a review of calorimetry and state estimation. *Polym React Eng* 8/1: 1–38
- Saenz de Buruaga I, Armitage PD, Leiza JR, Asua JM (1997) Nonlinear control for maximum production rate of latexes of well- defined polymer composition. *Ind Eng Chem Res* 36:4243–4254
- Schei TS (2008) On-line estimation for process control and optimization applications. *J Process Control* 18:821–828
- Tietze A, Lüdke I, Reichert K-H (1996) Temperature oscillation calorimetry in stirred tank reactors. *Chem Eng Sci* 51/11:3131–3137

Statistical Process Control in Manufacturing

O. Arda Vanli¹ and Enrique Del Castillo²

¹Department of Industrial and Manufacturing Engineering, High Performance Materials Institute, Florida A&M University, Florida State University, Tallahassee, FL, USA

²Department of Industrial and Manufacturing Engineering, The Pennsylvania State University, University Park, PA, USA

Abstract

Statistical process control (SPC) has been successfully utilized for process monitoring and variation reduction in manufacturing applications. This entry aims to review some of the important monitoring methods. We discuss fundamental process monitoring topics including Shewhart's model, $\bar{X}$ and R control charts, EWMA and CUSUM charts for monitoring small process shifts, process monitoring for autocorrelated data, and integration of statistical and engineering (or automatic) control techniques. We also illustrate the application of SPC in the more recently emerging areas including monitoring profiles, surfaces, and point cloud data sets in manufacturing. The goal is to provide readers from control theory, mechanical engineering, and electrical engineering an expository overview of the key topics in statistical process control.

Keywords

Statistical process control · EWMA · CUSUM · Profile monitoring · Point clouds

Introduction

Variation control is an important goal in manufacturing. The main tool for variation control used in discrete part manufacturing industries up to the 1960s is developed by W. Shewhart in the 1920s

and is what is known today as statistical process control or SPC (Shewhart 1939). Shewhart's SPC model assumes that the process varies about a fixed mean and that consecutive observations from a process are independent, as follows:

$$Y_t = \mu_0 + \epsilon_t \quad (1)$$

in which μ_0 is the in-control process mean and ϵ_t is iid (independent identically distributed) white noise $\epsilon \stackrel{iid}{\sim} N(0, \sigma^2)$. The Shewhart model can be used in distinguishing assignable cause variation from common cause variation. For example, a mean change from μ_0 to $\mu_1 = \mu_0 + \delta$ (where δ is the unknown magnitude of change) or a variance increase from σ_0^2 to σ_1^2 at an unknown point in time can be detected as assignable causes.

The objective of this entry is to highlight some of the important references in the SPC methods applied to manufacturing and to discuss similarities and joint applications SPC has with automatic process control. We discuss in detail fundamental process monitoring topics including Shewhart's model, $\bar{X}$ and R control charts, EWMA and CUSUM charts, process monitoring for autocorrelated data, and integration of statistical and engineering (or automatic) control techniques. In addition, we present some of the more advanced SPC methods developed recently for monitoring profiles, surfaces, and point clouds. The literature on statistical process control and applications to engineering problems is vast; therefore no effort is made for an exhaustive review. More complete reviews of the literature on statistical process control and adjustment methods can be found in texts including Montgomery (2013), Ryan (2011), and Del Castillo (2002).

Shewhart Control Charts

Shewhart's $\bar{X}$ and R control charts are used for monitoring, respectively, the process mean and process variance to distinguish between

common cause and assignable causes of variation (Shewhart 1939). "Common cause" variation is the natural variability of the process due to uncontrollable factors in the environment that is not avoidable without substantial changes to the process. "Assignable cause" variation is due to unwanted disturbances or upsets to the process that can be detected and removed to produce acceptable quality products. When only common cause variation exists, the process is said to be operating "in statistical control." Assignable causes of variation include operator changes, machine calibration errors, or raw material variation between suppliers.

Another concept that is closely related to the Shewhart's model is process capability. Process capability indices are used to assess whether the process is operating in a satisfactory manner with respect to the engineering specifications. It is crucial to attain a stable process (by identifying and eliminating all assignable causes of variation) before undertaking such a capability analysis because only when the samples come from a stable probability distribution can the future behavior of the process be predicted "within probability limits determined by the common-cause system" (Box and Kramer 1992).

Figure 1 illustrates the two main phases, referred to as Phase I and Phase II, in constructing Shewhart charts (Sullivan 2002), using semiconductor lithography process data given in Montgomery (2013). It is desired to establish a statistical control of the width of the resist using $\bar{X}$ and R charts. Twenty-five preliminary subgroups, each of size five wafers, were taken at 1-hour intervals, and the resist width is measured. In Phase I, a "retrospective analysis" is conducted, in which a historical set of data from the process is analyzed to bring an initially out-of-control process into statistical control. Subgroups $y_1, \dots, y_n$ of size n are taken and subgroup average $\bar{y}$ is used to monitor process mean μ_0 , and the subgroup range is used to monitor standard deviation of the process mean $\sigma_{\bar{y}} = \sigma/\sqrt{n}$. The upper and lower control limits are found for the $\bar{X}$ chart

as $\{UCL, LCL\} = \mu_0 \pm L\sigma_{\bar{Y}}$ where L is a constant representing the width of the control limits. Commonly chosen three-sigma limits (i.e., $L = 3$) provide a probability $p = 0.0027$ that a single point falls outside the limits when process is in control (“false alarm probability”). Points that fall outside the control limits are investigated and if an assignable cause was identified, then this point is omitted, and control limits are recalculated. This is repeated until no further points plot outside the limits. In Phase II, a “prospective analysis” is conducted, in which these charts are used to detect shifts in the process mean and variability by taking and measuring a new subgroup from the process sequentially.

The $\bar{X}$ and R charts from Phase I data shown in Fig. 1a indicate statistical control; hence the computed control limits can be used for Phase II monitoring. Twenty additional subgroups (also of size 5) are taken in Phase II while the control charts are in use. The Phase II charts shown in Fig. 1b indicate that process variability is stable, but the process mean has shifted at subgroup 18. The general trend in the $\bar{X}$ chart indicates that process mean probably has shifted earlier, around subgroup 13.

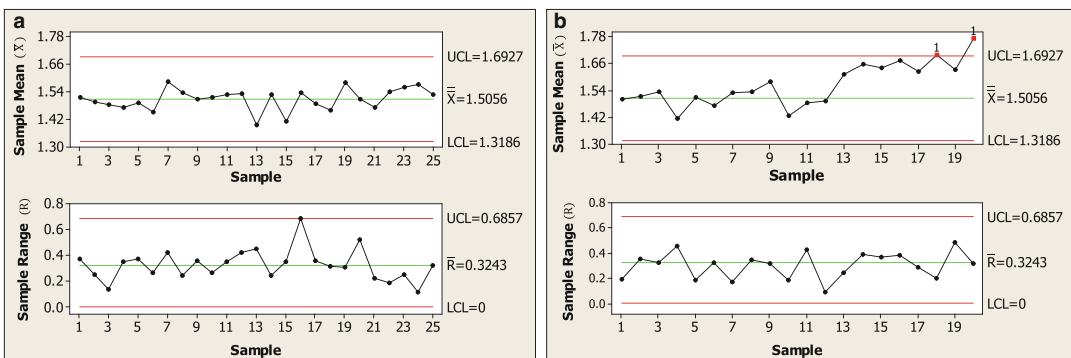
EWMA, CUSUM, and Changepoint Estimation

Shewhart charts can detect large magnitude process upsets reasonably well; however, they are relatively slow to detect small shifts. In order

to reduce the reaction time for smaller shifts, a set of “runs” rules (e.g., two out of three runs beyond 2σ limits or four out of five runs beyond 1σ limits) has been proposed (Western Electric Company 1956). A more systematic method is to accumulate information over successive observations using CUSUM and EWMA statistics rather than basing the detection on a single sample. In the cumulative sum (CUSUM) chart, a running total $\sum_{i=1}^t (\bar{Y}_i - \mu_0)$ is plotted against subgroup number t , and a shift from the in-control mean μ_0 is signaled by an upward or downward linear trend in the plot. A two-sided CUSUM is defined as (Woodall et al. 2004):

$$S_t^{\pm} = \max\{\pm Z_t - k + S_{t-1}^{\pm}, 0\} \text{ for } t = 1, 2, \dots \quad (2)$$

where S_t^+ and S_t^- are the one-sided upper and lower CUSUMs, respectively, $Z_t = (\bar{Y}_t - \mu_0)/\sigma_{\bar{Y}}$ is the standardized subgroup average, $k = |\mu_1 - \mu_0|/(2\sigma)$ is the reference value, and μ_1 is the level of process mean to be detected. An out-of-control signal is given at the first t for which $S_t > h$ where h is a suitably chosen threshold, usually selected based on the desired average number of samples to signal an alarm, also called the Average Run Length (ARL). The recommended value for the threshold h is 4 or 5 (corresponding to four or five times the process standard deviation σ), and the value for the reference k is almost always taken as 0.5 (corresponding to shift size $|\mu_1 - \mu_0| = \sigma$) (Montgomery 2013).



Statistical Process Control in Manufacturing, Fig. 1 Shewhart $\bar{X}$ and R chart from (a) Phase I analysis (b) Phase II analysis

Another chart that accumulates deviations over several samples is the exponentially weighted moving average (EWMA) which is based on the statistic (Lucas and Saccucci 1990):

$$Z_t = \lambda \bar{Y}_t + (1 - \lambda)Z_{t-1} \quad (3)$$

where $0 < \lambda < 1$ is a smoothing constant. Smaller λ provides large smoothing (similar to a large subgroup size n in the Shewhart charts). The starting value is the in-control mean $Z_0 = \mu_0$. It can be shown that Z_t is a weighted average of all previous sample means, where the weights decrease geometrically with the age of the subgroup mean. The EWMA statistic is plotted against the control limits $\mu_0 \pm L\sigma_{\bar{Y}}\sqrt{(\lambda/(2-\lambda))[1 - (1-\lambda)^{2t}]}$. Shewhart charts that are effective for large shifts are more useful for Phase I, and CUSUM or EWMA charts that are effective for small shifts are more appropriate for Phase II.

We illustrate in Fig. 2 how to monitor a process with CUSUM and EWMA charts using the lithography data studied in the previous section. The in-control process mean and standard deviation μ_0 and σ are found from the Phase I data. CUSUM upper and lower statistics $S_t^\pm$ computed with Phase II data are plotted in Fig. 2a (reference value $k = 0.5$ and threshold $h = 4$ are used). The upper CUSUM statistic S_t^+ crosses the upper control limit indicating an upward shift at subgroup 15. The EWMA statistic applied with $\lambda = 0.2$ on Phase II data, shown Fig. 2b, crosses the upper control limit at subgroup 16. As it can be seen, both CUSUM and EWMA charts can improve the reaction times of the Shewhart chart (recall from Fig. 1b that Shewhart chart signaled at subgroup 18).

When a control chart signals an assignable cause, it does not indicate when the process change actually occurred. Estimating the instant of the change, or *change point* estimation, is especially useful in Phase I analysis where little is known about the process and it is important to identify and remove the out-of-control samples from consideration (Hawkins et al. 2003; Basseville and Nikiforov 1993; Pignatiello and Samuel 2001). In a change point

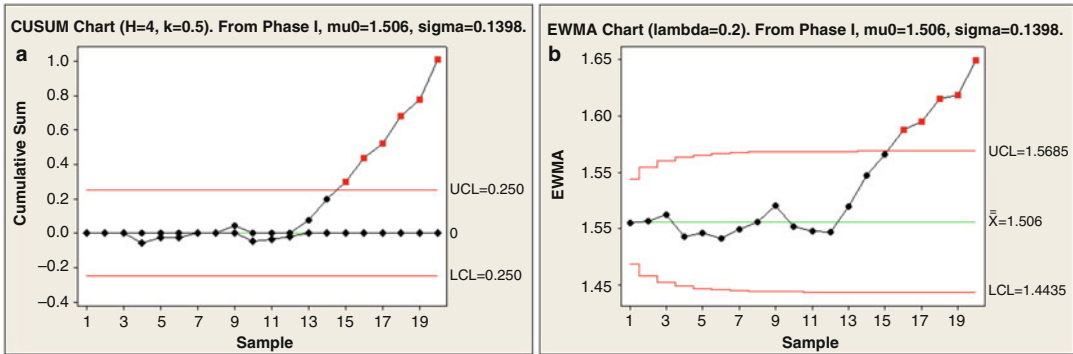
model, the process is modeled as:

$$\begin{aligned} Y_i &\sim N(\mu_1, \sigma^2) \text{ for } i = 1, 2, \dots, \tau \\ Y_i &\sim N(\mu_2, \sigma^2) \text{ for } i = \tau + 1, \dots, n \end{aligned} \quad (4)$$

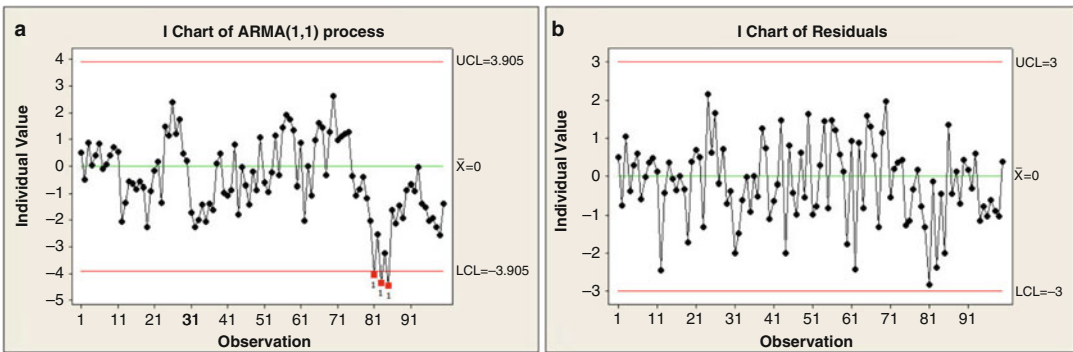
where τ is the unknown change point, at which the in-control mean μ_1 is assumed to shift to a new value μ_2 assuming μ_1 and σ are known but μ_2 is unknown. A Generalized Likelihood Ratio (GLR) test statistic $\Lambda_t = \sum_{i=1}^t \log f_2(y_i)/f_1(y_i)$ is used to test the hypothesis of a change point against the null hypothesis that there is no change. Assuming normality $f(y) = 1/\sqrt{2\pi\sigma} \exp[-(y - \mu)^2/(2\sigma^2)]$ is the probability density function of the quality characteristic. The change point model is equivalent to the CUSUM chart when all parameters μ_1 , μ_2 , and σ are known a priori. For the lithography Phase II data in Fig. 1b, it can be shown that the change point can be estimated as subgroup 13.

SPC on Controlled and Autocorrelated Processes

It is well known that automatic control performance relies heavily on the accuracy of the process models. An active field of research in recent years is the monitoring of controlled systems using SPC charts (Box and Kramer 1992) in order to reduce the effect of model accuracy. Shewhart charts can be used to monitor the output of a feedback controlled process; however, as the controller effectively corrects the shift, only a short window of opportunity is available to detect the shift (Vander Wiel et al. 1992). Tsung and Tsui (2003) showed that monitoring the control actions gives better run length performance than monitoring the output for small- and medium-sized shifts, and monitoring the output gives better performance for large shifts. In monitoring controlled processes, measurements taken at short intervals with positive autocorrelation usually inflate the rate of false alarms (Harris and Ross 1991). Widening the control limits and monitoring the residuals of a time-series model fitted to the observations are suggested strategies



Statistical Process Control in Manufacturing, Fig. 2 Phase II charts for lithography data (a) CUSUM chart and (b) EWMA chart

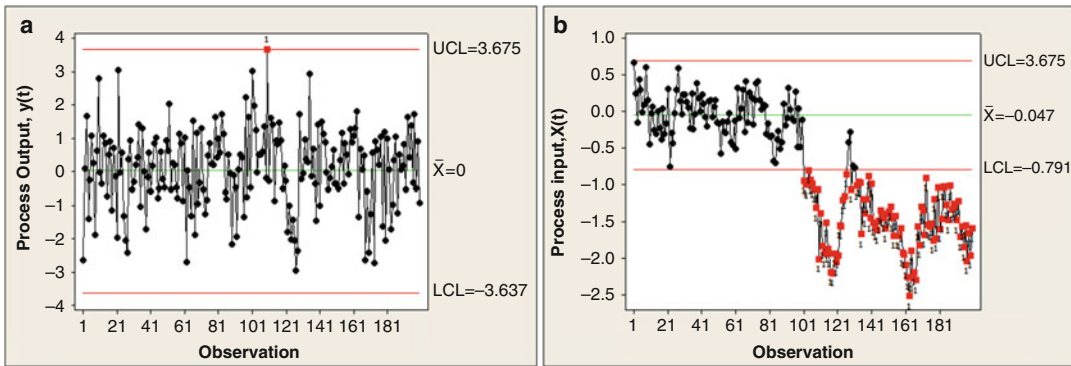


Statistical Process Control in Manufacturing, Fig. 3 (a) Shewhart chart for autocorrelated process. (b) Shewhart chart for residuals

to reduce the number of false alarms when monitoring positively autocorrelated processes (Alwan and Roberts 1988).

To illustrate the effects of autocorrelation, we consider simulated data from an autoregressive moving average ARMA(1,1) time-series disturbance process $D_t = 0.8D_{t-1} + \epsilon_t - 0.3\epsilon_{t-1}$ (Box et al. 1994) defined with the white noise process $\epsilon_t \stackrel{iid}{\sim} N(0, 1^2)$ (with in-control mean $\mu_0 = 0$ and variance $\sigma_D^2 = 1.694$). Figure 3a shows a realization of the process monitored with a Shewhart chart (control limits at $\mu_0 \pm 3\sigma_D$). Due to autocorrelation, false alarms are signaled at samples 81 to 83. Figure 3b shows the control chart monitoring of the residuals of an ARMA(1,1) model. Residuals (standard normal with mean 0 and variance 1) are not autocorrelated, so the Shewhart chart for residuals does not signal any false alarms.

We illustrate monitoring of controlled processes with simulated data from a transfer function model $Y_t = 2X_{t-1} + D_t$ where X_t are the adjustments made on the process. A proportional integral control rule $X_t = -0.1Y_t - 0.15 \sum_{i=1}^t Y_i$ is employed, and the disturbance D_t is assumed to follow the ARMA model considered earlier. As an assignable cause, the mean of the disturbance was shifted by a magnitude of $3\sigma_D$ after sample 100. Figure 4 shows the Shewhart charts monitoring the output Y_t and the input X_t . The effect of assignable cause (at sample 100) on the output is quickly removed by the controller; however, a sustained shift remains in the control input. The control chart for the input Fig. 4b signals the first alarm at sample 101 (much quicker) than the control chart for the output Fig. 4a which signals at sample 110.



Statistical Process Control in Manufacturing, Fig. 4 (a) Shewhart chart for controlled process Y_t . (b) Shewhart chart for input X_t

SPC for Monitoring Surface, Image, and Point Cloud Data

As advanced measurement technologies are more commonly utilized in manufacturing processes, more recent applications of SPC have focused on handling high-dimensional and more complex data in a more effective way. One such measurement technology is the three-dimensional (3D) laser scanners that can rapidly provide tens of thousands of data points, also called point clouds, allowing the entire surface geometry of a discrete part to be measured and represented quickly and inexpensively (Wells et al. 2013). Compared to the more traditional coordinate measurement systems (CMMs) that measure the product only at a limited number of points, such high-density measurement technologies allow one to monitor the entire surface and detect the occurrence of a large variety of fault patterns. Another emerging area is the use of high-density data in image and surface monitoring (Megahed et al. 2011; Qiu 2018; Colosimo 2018). Machine vision systems had long been applied as medical diagnosis tools using technologies including ultrasound, functional MRI, and X-ray (Qiu 2018). However, industrial applications for in-line process monitoring of surfaces in manufacturing have attracted an increasing interest in the recent years. Image-based SPC relies on sequences of images or videos taken from a process or a product with the goal

of quickly detecting and localizing the onset of product defects or process abnormalities by comparing a current image to the image that corresponds to an in-control state. Some applications include defect detection in textiles (Megahed et al. 2011), additively manufactured parts (Colosimo 2018), and liquid crystal display (LCD) surfaces (Jiang et al. 2005).

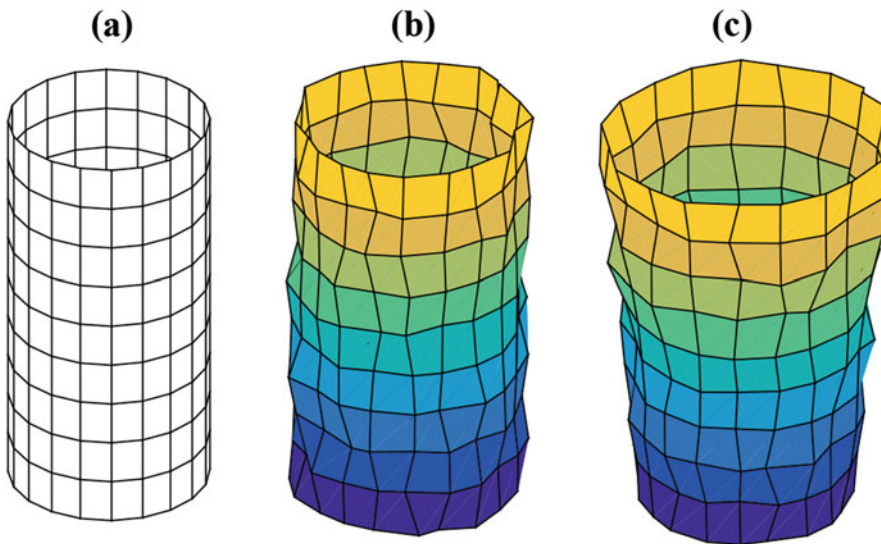
In the high-dimensional data environments, the quality of the product is often more adequately characterized by a relationship between the quality characteristic and one or more explanatory variables. For example in semiconductor manufacturing, the oxide thickness (quality characteristic) on a semiconductor wafer surface can be measured at given spatial locations (explanatory variable) (Gardner et al. 1997). In such cases, at each sampling instant, a set of measurements of the quality characteristic, as opposed to only one, is observed from the product under a range of explanatory variable values. In profile monitoring the goal is to represent the functional relationship between the quality characteristic and the explanatory variables by a statistical model, referred to as a “profile”, and detect any type of shift in the function (Woodall et al. 2004). Other examples of profiles in manufacturing include radius measurements in a turning process (Colosimo et al. 2014) and stamping force as a function of crank-arm angle in a steel stamping operation (Jin and Shi 2001). In the area of surface monitoring, both two- and three-dimensional

shapes of products are monitored using profile monitoring techniques. It has been shown that representing the surface shape with a functional model can be more efficient than the standard image monitoring approaches. A fundamental difference is that image intensity functions have jumps and singularities, but in profile monitoring, the surfaces are often assumed to be continuous (Qiu 2018).

Monitoring using 3D point clouds and image data has been studied based on the profile monitoring framework. Wells et al. (2013) proposed to monitor the deviations between points scanned from each actual part and the Computer Aided Design (CAD) surface of the part after the part and nominal are registered. This may be computationally expensive and may increase the variance due to the registration step, as pointed out by Del Castillo and Zhao (2019, to appear) and Zhao and del Castillo (2019), who propose instead a registration-free approach based on monitoring the spectrum of the point cloud. Spatiotemporal approaches were proposed by Megahed et al. (2011) to detect both the spatial location and size of a defect and the changepoint in time within the image stream. Jiang et al. (2005) developed

a spatial control chart to inspect uniformity of LCD monitors and detect type, size, and location of defects.

As an illustration of how SPC is applied in a surface monitoring problem, Fig. 5 shows simulated results for the cylindricity monitoring problem studied in Colosimo et al. (2014). These authors considered monitoring the cylindricity of carbon steel bars machined by lathe turning, and the goal is to signal an alarm when a “form error” is encountered. The cylindrical surface profiles are modeled using Gaussian processes (GP), and the GP-predicted deviations of the surface from the in-control pattern are monitored to detect form errors. Figure 5a shows the regular grid on which the surface coordinates are measured (e.g., using a 3D scanner). Figure 5b shows a case with only random measurement variation and no form error, and Fig. 5c shows a case with a form error (a “tapered” shape) which can be detected and identified with a surface monitoring approach. Colosimo et al. (2014) report that the use of GP-based methods can improve the detection speed of out-of-control states by up to 80 times compared to the current industrial practice.



Statistical Process Control in Manufacturing, Fig. 5 Surface monitoring of cylindrical parts. (a) Grid of measurement points. (b) Measured part subject to random measurement noise. (c) Measured part with a taper form error

Summary and Future Directions

In this entry we reviewed some of the commonly used statistical process control (SPC) methods for manufacturing systems. We reviewed several fundamental SPC concepts, including Phase I and Phase II monitoring and the Shewhart, EWMA, and CUSUM control charts. In addition, we highlighted several, more recently developed high-dimensional monitoring methods applied for profiles, surfaces, images, and point clouds. The concepts were illustrated where possible, with numerical examples. Other important SPC research areas that have not been reviewed in this entry include multivariate methods for monitoring processes with multiple quality characteristics taking advantage of relationships among them (Lowry and Montgomery 1995), multistage monitoring for processes with multiple processing steps and variation transmission (Tsung et al. 2008) and permutation-based nonparametric SPC methods that are robust to distributional assumptions (Chen et al. 2016; Capizzi and Masarotto 2017).

Cross-References

- ▶ [Control and Optimization of Batch Processes](#)
- ▶ [Controller Performance Monitoring](#)
- ▶ [Multiscale Multivariate Statistical Process Control](#)
- ▶ [Stream of Variations Analysis](#)

Bibliography

- Alwan LC, Roberts HV (1988) Time-series modeling for statistical process control. *J Bus Econ Stat* 6(1):87–95
- Basseville ME, Nikiforov IV (1993) Detection of abrupt changes: theory and application. Prentice-Hall, Englewood Cliffs, NJ
- Box GEP, Kramer T (1992) Statistical process monitoring and feedback adjustment: a discussion. *Technometrics* 34(3):251–267
- Box GEP, Jenkins GW, Reinsel GC (1994) Time Series Analysis, Forecasting and Control, Prentice Hall, Inc
- Capizzi G, Masarotto G (2017) Phase I distribution-free analysis of multivariate data. *Technometrics* 59(4):484–495
- Chen N, Zi X, Zou C (2016) A distribution-free multivariate control chart. *Technometrics* 58(4):448–459
- Colosimo BM (2018) Modeling and monitoring methods for spatial and image data. *Qual Eng* 30(1):94–111
- Colosimo BM, Ciccorella P, Pacella M, Blaco M (2014) From profile to surface monitoring: SPC for cylindrical surfaces via Gaussian processes. *J Qual Technol* 46(2):95–113
- Del Castillo E (2002) Statistical process adjustment for quality control. Wiley, New York
- Del Castillo E, Zhao X (2019, to appear) Industrial statistics and manifold data (with discussion). *Qual Eng*
- Gardner MM, Lu JC, Gyurcsik RS, Wortman JJ, Hornung BE, Heinisch HH, Rying EA, Davis JC, Mozumder PK (1997) Equipment fault detection using spatial signatures. *IEEE Trans Compon Packag Manuf Technol Part C* 20(4):295–304
- Harris TJ, Ross WH (1991) Statistical process control procedures for correlated observations. *Can J Chem Eng* 69(1):48–57
- Hawkins DM, Peihua Q, Chang WK (2003) The change-point model for statistical process control. *J Qual Technol* 35(4):355–366
- Jiang BC, Wang CC, Liu HC (2005) Liquid crystal display surface uniformity defect inspection using analysis of variance and exponentially weighted moving average techniques. *Int J Prod Res* 43(1):67–80
- Jin J, Shi J (2001) Automatic feature extraction of waveform signals for in-process diagnostic performance improvement. *J Intell Manuf* 12(3):257–268
- Lowry CA, Montgomery DC (1995) A review of multivariate control charts. *IIE Trans* 27(6):800–810
- Lucas JM, Saccucci MS (1990) Exponentially weighted moving average control schemes: properties and enhancements. *Technometrics* 32(1):1–12
- Megahed FM, Woodall WH, Camelio JA (2011) A review and perspective on control charting with image data. *J Qual Technol* 43(2):83–98
- Montgomery DM (2013) Introduction to statistical quality control, 7th edn. Wiley, New York
- Pignatiello JJ Jr, Samuel TR (2001) Estimation of the change point of a normal process mean in SPC applications. *J Qual Technol* 33(1):82–95
- Qiu P (2018) Jump regression, image processing, and quality control. *Qual Eng* 30(1):137–153
- Ryan TP (2011) Statistical methods for quality improvement, 3rd edn. Wiley, New York
- Shewhart WA (1939) Statistical method from the viewpoint of quality control. The Graduate School of the Department of Agriculture. Washington, DC
- Sullivan JH (2002) Detection of multiple change points from clustering individual observations. *J Qual Technol* 34(4):371–383
- Tsung F, Tsui KL (2003) A mean-shift pattern study on integration of SPC and APC for process monitoring. *IIE Trans* 35(3):231–242
- Tsung F, Li Y, Jin M (2008) Statistical process control for multistage manufacturing and service operations. *Int J Serv Oper Informatics* 3(2):191–204

- Vander Wiel SA, Tucker WT, Faltin FW, Doganaksoy N (1992) Algorithmic statistical process control: concepts and an application. *Technometrics* 34(3):286–297
- Wang A, Wang K, Tsung F (2014) Statistical surface monitoring by spatial-structure modeling. *J Qual Technol* 46(4):359–376
- Wells LJ, Megahed FM, Niziolek CB, Camelio JA, Woodall WH (2013) Statistical process monitoring approach for high-density point clouds. *J Intell Manuf* 24(6):1267–1279
- Western Electric (1956) *Statistical Quality Control Handbook*, Western Electric Corporation, Indianapolis, IN
- Woodall WH, Adams BM (1993) The statistical design of CUSUM charts. *Qual Eng* 5(4):559–570
- Woodall WH, Spitzner DJ, Montgomery DC, Gupta S (2004) Using control charts to monitor process and product quality profiles. *J Qual Technol* 36(3):309–320
- Zang Y, Qiu P (2018) Phase II monitoring of free-form surfaces: an application to 3D printing. *J Qual Technol* 50(4):379–390
- Zhao X, del Castillo E (2019) An intrinsic geometrical approach for statistical process control of surface data. Working paper, Engineering Statistics and Machine Learning Laboratory, Department of Industrial Engineering, PSU

Stochastic Adaptive Control

Tyrone Duncan and Bozena Pasik-Duncan
Department of Mathematics, University of
Kansas, Lawrence, KS, USA

Abstract

Stochastic adaptive control focuses on the control of partially known stochastic systems. These systems occur in both continuous and discrete time and are described by Markov chains, stochastic difference equations, and stochastic differential equations. Two major goals for the solutions are self-tuning and self-optimizing. These two goals are typically determined asymptotically so that self-optimality denotes the convergence of the average costs to the optimal long-run average cost for the true system.

Keywords

Stochastic systems · Stochastic control · System identification · Parameter estimation · Adaptive control

Introduction

Stochastic adaptive control denotes the control of partially known stochastic control systems. The stochastic control systems can be described by discrete- or continuous-time Markov chains or Markov processes, linear and nonlinear difference equations, and linear and nonlinear stochastic differential equations. The solution of a stochastic adaptive control problem typically requires the identification of the partially known stochastic system and the simultaneous control of the partially known system using the information from the concurrent identification scheme. Two desirable goals for the solution of a stochastic adaptive control problem are called self-tuning and self-optimality. Self-tuning denotes the convergence of the family of adaptive controls indexed by time to the optimal control for the true system. Self-optimizing denotes the convergence of the long-run average costs to the optimal long-run average cost for the true system. Typically to achieve the self-optimality, it is important that the family of parameter estimators from the identification scheme be strongly consistent, that is, this family converges (almost surely) to the true parameter values. Thus, with self-optimality, asymptotically a partially known system can be controlled as well as the corresponding known system.

Motivation and Background

In almost every formulation of a stochastic control problem from a physical system, the physical system is incompletely known so the stochastic system model is only partially known. This lack of knowledge can often be described by some unknown parameters for a mathematical model, and furthermore the noise inputs for the model can describe unmodeled dynamics or perturbations to the physical system. The lack of knowledge of some parameters of the model can be modeled either by random variables with known prior distributions or as fixed unknown values. The former description requires Bayesian estimation, and the latter description requires parameter

estimation such as least squares, weighted least squares or maximum likelihood.

Stochastic adaptive control arose as a natural evolution from the results in stochastic control; in fact stochastic adaptive control is an application of stochastic control, and in particular stochastic adaptive control developed from some well-known control problems (Davis and Vinter 1985). The optimal control of Markov chains had been developed for some time, so it was natural to investigate the adaptive control of Markov chains. Mandl (1973) was probably the first to consider this adaptive control problem in generality. His conditions for strong consistency of a family of estimators were fairly restrictive. Borkar and Varaiya (1982) simplified the conditions for the estimation part of the problem by only requiring convergence of the estimators of the parameters so that the resulting transition probabilities of the Markov chain are identical to the transition probabilities for the true optimal solution.

A second major direction for stochastic adaptive control is described by ARMAX (autoregressive moving average with exogenous inputs) models. These are discrete-time models that can be described in terms of polynomials in a time shift operator. A closely related and often equivalent model is multidimensional linear difference equations in a state-space form. Since the solution of the infinite time horizon discrete-time stochastic control problem was available in the late 1950s, it was natural to consider the corresponding adaptive control problem. Methods such as least squares, weighted least squares, maximum likelihood, and stochastic approximation were used for parameter identification and a certainty equivalence adaptive control for the system, that is, using the current estimate of the parameters as the true parameters to verify self-optimality. An important development in stochastic adaptive control is a result called the self-tuning regulator where the convergence of estimators of unknown parameters implied the convergence of the output tracking error (Astrom and Wittenmark 1973; Goodwin et al. 1981; Guo 1995, 1996; Guo and Chen 1991; Kumar 1990).

A number of monographs treat various aspects of stochastic adaptive control problems, e.g.,

Astrom and Wittenmark 1989; Chen and Guo 1991; Kumar and Varaiya 1986; Ljung and Soderstrom 1983. An extensive survey article on the early years of stochastic adaptive control is given by Kumar (1985).

Structures and Approaches

Various requirements can be made for the adaptive control of a stochastic system. However it is important to understand the stochastic systems. It can only be required that the family of adaptive controls is stabilizing the unknown system or that the family of adaptive controls converges to the optimal control for the true system or that the family of adaptive controls has a long-run average cost that is equal to the optimal average cost for the true system. The identification part of the adaptive control problem can be Bayesian estimation (Kumar 1990) if the parameters are assumed to be random variables or parameter estimation (Bercu 1995; Lai and Wei 1982) if the parameters are assumed to be unknown constants. The identification scheme may also incorporate information about the running cost.

For linear systems with white noise inputs, it is well-known to use least squares (or equivalently maximum likelihood) estimation to estimate parameters. However, for stochastic adaptive control problems, the sufficient conditions for the family of estimators to be strongly consistent are fairly restrictive (e.g., Lai and Wei 1982), and in fact the family of estimators may not even converge in general. A weighted least squares estimation scheme (Bercu 1995) can guarantee convergence of the family of estimators without the need of any excitation condition (Guo 1996). Some other estimation methods are stochastic approximation (Guo and Chen 1991) and an ordinary differential equation approach (Ljung and Soderstrom 1983). For discrete-time nonlinear systems, a family of strongly consistent estimators may not converge sufficiently rapidly even to stabilize the nonlinear system (Guo 1997).

The study of stochastic adaptive control of continuous-time linear stochastic systems with long-run average quadratic costs developed

somewhat after the corresponding discrete-time study (e.g., Duncan and Pasik-Duncan 1990). A solution with basically the natural assumptions from the solution of the known system problem using a weighted least squares identification scheme is given in Duncan et al. (1999).

Another family of stochastic adaptive control problems is described by linear stochastic equations in an infinite dimensional Hilbert space. These models can describe stochastic partial differential equations and stochastic hereditary differential equations. Some linear-quadratic-Gaussian control problems have been solved, and these solutions have been used to solve some corresponding stochastic adaptive control problems (e.g., Duncan et al. 1994a).

Optimal control methods such as Hamilton-Jacobi-Bellman equations and a stochastic maximum principle have been used to solve stochastic control problems described by nonlinear stochastic differential equations (Fleming and Rishel 1975). Thus, it was natural to consider stochastic adaptive control problems for these systems. The results are more limited than the results for linear stochastic systems (e.g., Duncan et al. 1994b).

Other stochastic adaptive control problems have recently emerged that are modeled by multi-agents, such as mean field stochastic adaptive control problems (e.g., Nourian et al. 2012).

A Detailed Example: Adaptive Linear-Quadratic-Gaussian Control

This example is a model that is the most well-known continuous-time stochastic adaptive control problem. Likewise for a known continuous-time system, this stochastic control problem is the most basic and well-known solvable problem. The controlled system is described by the following multidimensional stochastic differential equation:

$$\begin{aligned} dX(t) &= AX(t)dt + BU(t)dt + CdW(t) \\ X(0) &= X_0 \end{aligned}$$

where $X(t) \in \mathbb{R}^n$, $U(t) \in \mathbb{R}^m$, and $(W(t), t \geq 0)$ is an $\mathbb{R}^p$ -valued standard Brownian motion and (A, B, C) are appropriate linear transformations. $X(t)$ is the state of the system at time t , and $U(t)$ is the control at time t . It is assumed that A, B , and C are unknown linear transformations. The cost functional, $J(\cdot)$, is a long-run average (ergodic) quadratic cost functional that is given by

$$\begin{aligned} J(U) &= \limsup_{T \rightarrow \infty} \frac{1}{T} \int_0^T \langle QX(t), X(t) \rangle \\ &\quad + \langle RU(t), U(t) \rangle dt \end{aligned}$$

where $R > 0$ and $Q \geq 0$ are symmetric linear transformations and $\langle \cdot, \cdot \rangle$ is the canonical inner product in the appropriate Euclidean space. The standard assumptions for the control of the known system are made also for the adaptive control problem, that is, the pair (A, B) is controllable, and $(A, Q^{\frac{1}{2}})$ is observable. An optimal control for the known system is

$$U^0(t) = -R^{-1}B^T S X(t)$$

where S is the unique positive, symmetric solution of the following algebraic Riccati equation:

$$A^T S + SA - SBR^{-1}B^T S + Q = 0$$

The optimal cost is

$$J(U^0) = \text{tr}(C^T S C)$$

The unknown quantity $C^T C$ can be identified given $(X(t), t \in [a, b])$ for $a < b$ arbitrary from the quadratic variation of Brownian motion, so the identification of C is not considered here. Since it is assumed that the pair (A, B) is unknown, the system equation is rewritten in the following form:

$$dX(t) = \theta^T \varphi(t)dt + CdW(t)$$

where $\theta^T = [A \ B]$ and $\varphi^T(t) = [X^T(t) \ U^T(t)]$. A family of continuous-time weighted least squares recursive estimators $(\theta(t), t \geq 0)$ of

θ is given by the following stochastic equation (Guo 1996):

$$\begin{aligned}d\theta(t) &= a(t)P(t)\varphi(t)[dX^T(t) - \varphi^T(t)\theta(t)dt] \\dP(t) &= -a(t)P(t)\varphi(t)\varphi^T(t)P(t)dt\end{aligned}$$

where $(a(t), t \geq 0)$ is a suitable family of positive stochastic weights (Duncan et al. 1999). A family of estimates $(\hat{\theta}(t), t \geq 0)$ is obtained from $(\theta(t), t \geq 0)$ and is expressed as $\hat{\theta}(t) = [A(t) B(t)]$ (Duncan et al. 1999). A process $(S(t), t \geq 0)$ is obtained using $(A(t), B(t))$ in place of (A, B) by solving the following stochastic algebraic Riccati equation for each $t \geq 0$:

$$\begin{aligned}A^T(t)S(t) + S(t)A(t) \\- S(t)B(t)R^{-1}B^T(t)S(t) + Q = 0\end{aligned}$$

A certainty equivalence method is used to determine the control, that is, it is assumed that the pair $(A(t), B(t))$ is the correct pair for the true system, so a certainty equivalence adaptive control $U(t)$ is given by

$$U(t) = R^{-1}B^T S(t)X(t)$$

It can be shown (Duncan et al. 1999) that the family of estimators $((A(t), B(t)), t \geq 0)$ is strongly consistent and that the family of adaptive controls given by the previous equality is self-optimizing, that is, the long-run average cost $J(U) = J(U^0) = \text{tr}(C^T S C)$ where S is the solution of the algebraic Riccati equation for the true system.

Future Directions

A number of important directions for stochastic adaptive control are easily identified. Only four of them are described briefly here. The adaptive control of the partially observed linear-quadratic-Gaussian control problem (Fleming and Rishel 1975) is a major problem to be solved using the same assumptions of controllability and observability as for the known system. This

problem is a generalization of the example given above where the output (linear transformation) of the system is observed with additive noise and the family of controls is restricted to depend only on these observations. Another major direction is to modify the detailed example above by replacing the Brownian motion in the stochastic equation for the state by an arbitrary fractional Brownian motion (Duncan and Pasik-Duncan 2013) or by an arbitrary square-integrable stochastic process with continuous sample paths. For this latter problem it is necessary to use recent results for optimal controls for the true system and to have strongly consistent families of estimators. A third major direction is the adaptive control of nonlinear stochastic systems. A fourth major direction is the adaptive control of mean field games (Huang et al. 2006; Lasry and Lions 2007; Duncan and Tembine 2018).

Cross-References

- [Adaptive Control: Overview](#)
- [Mean Field Games](#)
- [System Identification: An Overview](#)
- [Use of Gaussian Processes in System Identification](#)

Bibliography

- Astrom KJ, Wittenmark B (1973) On self-tuning regulators. *Automatica* 9:185–199
- Astrom KJ, Wittenmark B (1989) *Adaptive control*. Addison-Wesley, Reading
- Bercu B (1995) Weighted estimation and tracking for ARMAX models. *SIAM J Control Optim* 33:89–106
- Borkar V, Varaiya P (1982) Identification and adaptive control of Markov chains. *SIAM J Control Optim* 20: 470–489
- Chen HF, Guo L (1991) *Identification and stochastic adaptive control*. Birkhauser, Boston
- Davis MHA, Vinter RD (1985) *Stochastic modelling and control*. Chapman and Hall, London
- Duncan TE, Pasik-Duncan B (1990) Adaptive control of continuous time linear systems. *Math Control Signals Syst* 3:43–60

- Duncan TE, Maslowski B, Pasik-Duncan B (1994a) Adaptive boundary and point control of linear stochastic distributed parameter systems. *SIAM J Control Optim* 32:648–672
- Duncan TE, Pasik-Duncan B, Stettner L (1994b) Almost self-optimizing strategies for the adaptive control of diffusion processes. *J Optim Theory Appl* 81: 470–507
- Duncan TE, Guo L, Pasik-Duncan B (1999) Adaptive continuous-time linear quadratic Gaussian control. *IEEE Trans Autom Control* 44:1653–1662
- Duncan TE, Tembine T (2018) Linear-quadratic mean field type games: a direct method. *Games J* 9:7–16
- Duncan TE, Pasik-Duncan B (2013) Linear-quadratic fractional Gaussian control. *SIAM J Control Optim* 51:4604–4619
- Fleming WH, Rishel RW (1975) *Deterministic and stochastic optimal control*. Springer, New York
- Goodwin G, Ramadge P, Caines PE (1981) Discrete time stochastic adaptive control. *SIAM J Control Optim* 19:820–853
- Guo L (1995) Convergence and logarithm laws of self-tuning regulators. *Automatica* 31:435–450
- Guo L (1996) Self-convergence of weighted least squares with applications. *IEEE Trans Autom Control* 41:79–89
- Guo L (1997) On critical stability of discrete time adaptive nonlinear control. *IEEE Trans Autom Control* 42:1488–1499
- Guo L, Chen HF (1991) The Astrom-Wittenmark self-tuning regulator revisited and ELS based adaptive trackers. *IEEE Trans Autom Control* 36: 802–812
- Huang M, Caines P E, Malhame, R (2006) Large population stochastic dynamical games, closed loop McKean-Vlasov systems and the Nash certainty equivalence. *IEEE Trans Autom Control* 52:1560–1571
- Kumar PR (1985) A survey of some results in stochastic adaptive control. *SIAM J Control Optim* 23: 329–380
- Kumar PR (1990) Convergence of adaptive control schemes with least squares estimates. *IEEE Trans Autom Control* 35:416–424
- Kumar PR, Varaiya P (1986) *Stochastic systems, estimation, identification and adaptive control*. Prentice-Hall, Englewood Cliffs
- Lai TL, Wei CZ (1982) Least square estimation is stochastic regression models with applications to identification and control of dynamic systems. *Ann Stat* 10: 154–166
- Lasry JM, Lions PL (2007) Mean field games. *Jpn J Math* 2:227–269
- Ljung L, Soderstrom T (1983) *Theory and practice of recursive identification*. MIT Press, Cambridge
- Mandl P (1973) On the adaptive control of finite state Markov processes. *Z Wahr Verw Geb* 27:263–276
- Nourian M, Caines PE, Malhame RP (2012) Mean field LQG control in leader-follower stochastic multi-agent systems: likelihood ratio based adaptation. *IEEE Trans Autom Control* 57:2801–2816

Stochastic Approximation with Applications

Han-Fu Chen

Key Laboratory of Systems and Control,
Institute of Systems Science, AMSS, Chinese
Academy of Sciences, Beijing, People's
Republic of China

Abstract

The topic of stochastic approximation (SA) and its pioneer algorithm (the Robbins-Monro (RM) algorithm) with methods for its convergence analysis are described. Algorithms modified from the RM algorithm such as the SA algorithm with constant step-size and the SA algorithm with expanding truncations (SAAWET) are demonstrated. Applications of SA to problems arising from optimization, system identification, adaptive control, signal processing, etc. are presented.

Keywords

Robbins-Monro algorithm · Optimization · Expanding truncations · System identification · Control · Signal processing · Distributed algorithm

Introduction

Estimating unknown parameters is ubiquitous in systems and control. For a fixed parameter, say x^0 , there are infinitely many functions having x^0 as its root. Let x^0 be the root of a function $f(\cdot) : f(x^0) = 0$. If $f(\cdot)$ is known, then there are many well-known numerical methods, which can be used to search its roots. In practice, however, $f(\cdot)$ possibly is not exactly known, and the value of $f(x)$ at any x may be available only with errors. In other words, at any time $k + 1$, the available observation about $f(x_k)$ is

$$y_{k+1} = f(x_k) + \epsilon_{k+1}, \quad (1)$$

where ϵ_{k+1} is the observation error. The topic of SA is to design algorithms which recursively generate $x_1, x_2, \dots$ converging to x^0 , starting from an initial value x_0 .

The first such an algorithm Robbins and Monro (1951) is presented in section “[RM Algorithm and Its Convergence Analysis](#).” SA algorithms with constant step-size and with expanding truncations are given in section “[SA Algorithms Modified from RM Algorithm](#).” Applications of SA to optimization and to problems from systems and control are presented in sections “[Application of SA to Optimization](#)” and “[Application of SA to Systems and Control](#),” respectively.

RM Algorithm and Its Convergence Analysis

To seek the roots of an unknown function $f(\cdot) : R^l \rightarrow R^l$ by using the observation (1), the following recursive algorithm is proposed in Robbins and Monro (1951):

$$x_{k+1} = x_k + a_k y_{k+1} \quad \text{with}$$

$$a_k > 0, \quad a_k \rightarrow 0, \quad \text{and} \quad \sum_{k=1}^{\infty} a_k = \infty. \quad (2)$$

This algorithm is called the RM algorithm and the function $f(\cdot)$ the regression function. The root set of $f(\cdot)$ is denoted by $J \triangleq \{x \in R^l : f(x) = 0\}$, which may be a singleton x^0 . However, the algorithm is not necessary to have the desired convergence. For example, in the case where $f(x) = (x - 10)^3$ and $x_0 = 0$, the RM algorithm never converges to the root 10, even if $\epsilon_k \equiv 0$.

For convergence analysis of the RM algorithm, there are the probabilistic method and the ODE (ordinary differential equation) method. In what follows $(\Omega, \mathcal{F}, P)$ denotes the probability space, and algorithms are considered for fixed $\omega \in \Omega$.

Probabilistic Method

This approach is based on probabilistic assumptions on the observation noise $\{\epsilon_k\}$. For example,

if $\{\epsilon_k\}$ is assumed to be a sequence of zero-mean mutually independent and identically distributed (iid) random vectors with finite second moment, then the resulting sequence of estimates $\{x_k\}$ is a Markov process. In addition, assume that (i) there exists a Lyapunov function $v(\cdot) : R^l \rightarrow R$ such that $v(x) > 0 \quad \forall x \neq x^0$, $v(x^0) = 0$, $v(x) \rightarrow \infty$ as $\|x\| \rightarrow \infty$, and

$$\sup_{\|x - x^0\| > \epsilon} v_x^T(x) f(x) < 0 \quad \forall \epsilon > 0$$

with $v_x(x)$ being the gradient of $v(\cdot)$; and (ii) $\sum_{i=1}^{\infty} a_i^2 < \infty$, and $\|f(x)\|^2 < cv(x)$ for some constant c . Then, $x_k \rightarrow x^0$ with probability 1. If $v(x)$ is a quadratic function, then the last condition implies that as $\|x\| \rightarrow \infty$, $\|f(x)\|$ should not grow up faster than linearly.

The observation noise $\{\epsilon_k\}$ may be a martingale difference sequence, a Markov chain, or a stochastic process with some specified properties, in particular, it may depend on the past estimates. Nevelson and Khasminskii (1976). Then, various tools developed in stochastic analysis can be applied to consider the convergence problem of $\{x_k\}$. The convergence with probability one or in the mean-square sense are usually expected to obtain. However, for this the probabilistic assumptions are often hard to be met in applications.

In addition to convergence, the convergence rate, asymptotic normality, and asymptotic efficiency can also be derived (Nevelson and Khasminskii 1976; Chen 2002). To achieve asymptotic efficiency, the averaging technique is introduced in Polyak and Judisky (1992).

ODE Method

To avoid the restrictive conditions used by the probabilistic method for convergence analysis, the ODE method is introduced in Kushner and Clark (1978) and Ljung (1977). The probabilistic assumptions made on $\{\epsilon_k\}$ are replaced by the following path-wise condition:

$$\lim_{T \rightarrow 0} \limsup_{n \rightarrow \infty} \frac{1}{T} \left\| \sum_{k=n}^{m(n,T)} a_i \epsilon_{i+1} \right\| = 0, \quad (3)$$

$$a_k \epsilon_{k+1} \xrightarrow[k \rightarrow \infty]{} 0,$$

where

$$m(k, T) \triangleq \max \left\{ m : \sum_{i=k}^m a_i \leq T \right\}. \quad (4)$$

It is clear that (3) holds whenever either $\epsilon_{k+1} \xrightarrow[k \rightarrow \infty]{} 0$ or $\sum_{i=1}^{\infty} a_i \epsilon_{i+1} < \infty$. Further, the growth rate restriction on $f(\cdot)$ is replaced by its continuity.

Assume the estimates $\{x_k\}$ generated by the RM algorithm are bounded. Interpolate them into a continuous function $x^0(t)$ with interpolating length $\{a_k\}$. Define $t_n = \sum_{i=0}^{n-1} a_i$, $x^n(t) = x^0(t + t_n)$ for $t \geq -t_n$ and $x^n(t) = x_0$ for $t \leq -t_n$. With possible exception of a set with probability zero, the limit $x(\cdot)$ of any convergent subsequence of $\{x^n(\cdot)\}$ satisfies the ODE $\frac{dx(t)}{dt} = f(x(t))$. Let x^0 be its stable equilibrium with domain of attraction $DA(x^0)$. If the sequence $\{x_k\}$ enters infinitely often a compact subset of $DA(x^0)$, then $x_k \xrightarrow[k \rightarrow \infty]{} x^0$ (Kushner and Clark 1978). Without any stability assumption on the ODE, it can still be established that the limit set of $\{x_k\}$ is a connected set internally chain-recurrent for the ODE (Benaim 1996).

It is noticed that in comparison with the probabilistic method, the ODE method requires much weaker conditions on both $\{\epsilon_k\}$ and $f(\cdot)$. So, it is widely used in applications (Benveniste et al. 1990).

SA Algorithms Modified from RM Algorithm

Algorithm with Constant Step-Size

In the RM algorithm, the step-size $\{a_k\}$ is tending to zero. When the constant step-size, i.e., $a_k \equiv \epsilon$ with $\epsilon > 0$, is used, write x_k as x_k^ϵ . Then $x_{k+1}^\epsilon = x_0^\epsilon + \epsilon \sum_{i=0}^k [f(x_i^\epsilon) + \epsilon_{i+1}]$, which, in general, cannot converge path-wisely as $k \rightarrow \infty$.

By the weak convergence method, the obtained sequence $\{x_k^\epsilon\}$ is interpolated to a piece-wisely constant function $x^\epsilon(t) \triangleq x_k^\epsilon$ for $t \in [\epsilon k, \epsilon(k+1))$, and it is shown that as $\epsilon \rightarrow 0$, $x^\epsilon(\cdot)$ weakly converges to $x(\cdot)$ which is a solution to ODE $\dot{x}(t) = f(x(t))$, $x(0) = x_0$. Further, if x^0 is its unique asymptotically stable point, then the distance between $\{x^\epsilon(t_\epsilon + \cdot)\}$ and x^0 tends to zero in probability, whenever $t_\epsilon \rightarrow \infty$ as $\epsilon \rightarrow 0$. Under some additional conditions, it appears that $E\|x_n^\epsilon - x^0\| = O(\sqrt{\epsilon})$, and as $\epsilon \rightarrow 0$ the limit for the interpolated function of $\{\frac{x_n^\epsilon - x^0}{\sqrt{\epsilon}}\}$ is a solution to a stochastic differential equation driven by a standard Wiener process (Kushner and Yin 2003).

It is noticed that the stochastic gradient descent algorithm used in Goodfellow et al. (2016) for deep learning is an SA algorithm with constant step-size.

SA Algorithm with Expanding Truncations (SAAWET)

In the case where a region the sought-for root x^0 belongs to is known, then a truncated RM algorithm can be defined so that the estimates generated by the algorithm may path-wisely converge to x^0 .

However, the availability of the region and the boundedness assumption on $\{x_k\}$ greatly prevent SA algorithm from applications. To overcome the difficulty, the RM algorithm is truncated, but the truncation bound is not fixed but adaptively selected.

Let $\{M_k\}$ be a sequence of positive numbers increasingly diverging to infinity, and let x^* be a fixed point in R^l . With an arbitrarily fixed initial value x_0 , SAAWET is recursively defined as follows (Chen and Zhu 1986):

$$x_{k+1} = (x_k + a_k y_{k+1}) I_{[\|x_k + a_k y_{k+1}\| \leq M_{\sigma_k}]} + x^* I_{[\|x_k + a_k y_{k+1}\| > M_{\sigma_k}]}, \quad (5)$$

$$\sigma_k = \sum_{i=1}^{k-1} I_{[\|x_i + a_i y_{i+1}\| > M_{\sigma_i}]}, \quad \sigma_0 = 0, \quad (6)$$

where $I_{[A]}$ is the indicator of an ω -set A : $I_{[A]} = 1$ if $\omega \in A$, and $I_{[A]} = 0$ if $\omega \notin A$. The observation noise ϵ_{k+1} may depend on x_k .

Assume the following conditions: A1: $a_k > 0$, $a_k \xrightarrow[k \rightarrow \infty]{} 0$, and $\sum_{k=1}^{\infty} a_k = \infty$; A2: There exists a continuously differentiable function $v(\cdot) : R^l \rightarrow R$ such that $\sup_{\delta \leq d(x, J) \leq \Delta} f^T(x) v_x(x) < 0$ for any $\Delta > \delta > 0$, where $d(x, J) \triangleq \inf_y \{\|x - y\| : y \in J\}$, $J \triangleq \{x : f(x) = 0\}$, and $v_x(\cdot)$ denotes the gradient of $v(\cdot)$. Further, $v(J) \triangleq \{v(x) : x \in J\}$ is nowhere dense, and x^* used in (5) is such that $v(x^*) < \inf_{\|x\|=c_0} v(x)$ with $\|x^*\| < c_0$ for some $c_0 > 0$; A3: $f(\cdot)$ is measurable and locally bounded.

Let the noise ϵ_{k+1} be given by $\epsilon_{k+1} = \epsilon_{k+1}(x_k, \omega)$, $\omega \in \Omega$, where $\epsilon_{k+1}(\cdot, \cdot) : (R^l \times \Omega, \mathcal{B}^l \times \mathcal{F}) \rightarrow (R^l \times \mathcal{B}^l)$ is a measurable function defined on the product space.

Further, assume

A4: For the fixed sample path ω under consideration, the following limit takes place:

$$\lim_{T \rightarrow 0} \limsup_{k \rightarrow \infty} \frac{1}{T} \left\| \sum_{i=n_k}^{m(n_k, T_k)} a_i \epsilon_{i+1}(x_i(\omega), \omega) \right\| = 0 \quad \forall T_k \in [0, T] \quad (7)$$

along subscripts $\{n_k\}$ of any convergent subsequence $\{x_{n_k}(\omega)\}$ of $\{x_k(\omega)\}$.

General Convergence Theorem 1 (Chen 2002)

Let $\{x_k\}$ be given by (5) and (6) with a given initial value x_0 . Assume A1–A3 hold. Then, $d(x_k, J) \xrightarrow[k \rightarrow \infty]{} 0$ for any sample path ω for which A4 holds.

If the root set is a singleton $J = x^0$, further if $f(\cdot)$ is continuous at x^0 , then under A1–A3 for $x_k \xrightarrow[k \rightarrow \infty]{} x^0$ the necessary and sufficient condition is that ϵ_k can be decomposed into two parts $\epsilon_k = \epsilon_k^{(1)} + \epsilon_k^{(2)}$ such that $\epsilon_k^{(1)} \xrightarrow[k \rightarrow \infty]{} 0$ and $\sum_{i=1}^{\infty} a_i \epsilon_{i+1}^{(2)} < \infty$ (Wang et al. 1996).

Application of SA to Optimization

Kiefer-Wolfowitz Algorithm and Its Randomization

Searching extremums of a function $L(x)$, $x \in R^l$ when its gradient $L_x(x) \triangleq f(x)$ can be observed (possibly with noise) is the topic of SA consisting in finding the root set $J \triangleq \{x : f(x) = 0\}$. The problem is how to proceed when only the function $L(\cdot)$ itself rather than its gradient can be observed. Let x_k be the estimate at time k for the minimizer (maximizer) of $L(\cdot)$, and let $y_{k+1}^+ = L(x_k^+) + \xi_{k+1}^+$, $y_{k+1}^- = L(x_k^-) + \xi_{k+1}^-$ be two observations of $L(\cdot)$ at time $k+1$ with noises ξ_{k+1}^+ and ξ_{k+1}^- , respectively, where $x_k^+ = [x_k^1, \dots, x_k^{i-1}, x_k^i + c_k, x_k^{i+1}, \dots, x_k^l]^T$, and $x_k^- = [x_k^1, \dots, x_k^{i-1}, x_k^i - c_k, x_k^{i+1}, \dots, x_k^l]^T$.

Then, $y_{k+1}^i \triangleq \frac{y_{k+1}^+ - y_{k+1}^-}{2c_k}$ can naturally serve as the observation of $f_i(x_k)$, the i th component of the gradient $L_x(x) (\triangleq f(x))$. Thus, $y_{k+1} \triangleq [y_{k+1}^1, \dots, y_{k+1}^l]^T = f(x_k) + \epsilon_{k+1}$, where $\epsilon_{k+1} \in R^l$ with i th component equal to

$$\frac{L(x_k^+) - L(x_k^-)}{2c_k} - f_i(x_k) + \frac{\xi_{k+1}^+ - \xi_{k+1}^-}{2c_k}.$$

The resulting RM algorithm for searching extremum is called Kiefer-Wolfowitz (KW) algorithm (Kiefer and Wolfowitz 1952). The KW algorithm may fall into a local attraction domain of L , and for the global optimization, the simulating annealing method may be applied, but it provides convergence in probability only. The KW algorithm combined with initial value selection incorporated with some switch mechanism may solve the global optimization problem with probability one (Fang and Chen 2000).

It is noticed that each updating requires $2l$ observations, which may cause difficulty when l is large. A randomized version is proposed in Spall (1992) consisting in replacing the component-wise perturbations described above with simultaneous random perturbations, i.e., for a recursion only two observations $y_{k+1}^+ = L(x_k +$

$c_k \Delta_k) + \xi_{k+1}^+$ and $y_{k+1}^- = L(x_k - c_k \Delta_k) + \xi_{k+1}^-$ are needed, where $\Delta_k \in R^l$ with iid components Δ_k^i with both $|\Delta_k^i|$ and $|\frac{1}{\Delta_k^i}|$ bounded by a constant and $E \frac{1}{\Delta_k^i} = 0$. The convergence analysis is given in Spall (1992), while the version with expanding truncations is given in Chen et al. (1999).

In connection with the constrained optimization, the constrained or projected algorithms are proposed (Kushner and Clark 1978; Kushner and Yin 2003).

Distributed Stochastic Optimization

Let a network consist of N agents. In some cases, for example, if the time delays caused by information exchange among agents are not negligible, or the computation time spent on one step recursion is not the same for different agents, then it is natural to consider the asynchronous SA (Borkar 1998; Tsitsiklis 1994) in order to solve the optimization problem. SAAWET works well in the asynchronous case, but the truncation mechanism should be adjusted in accordance with multi-agents (Fang and Chen 2001).

Stochastic optimization and SA also work when the multi-agent system is equipped with a graph describing the information exchange among agents: each agent can exchange information only with its neighbors (Bianchi et al. 2013; Lei and Chen 2020; Lei et al. 2018). For distributed stochastic optimization subject to a global constraint, an SA-based distributed primal-dual algorithm is proposed (Lei et al. 2018) for a cost function being a sum of local functions. Each agent updates its estimate by using the local observations and the information derived from neighboring agents. The estimates generated by the algorithm at all agents pathwisely reach a consensus belonging to the optimal solution set.

Application of SA to Systems and Control

Parameter Estimation by SA

As mentioned in Introduction, the estimated parameter denoted by x^0 may be treated as a root

of an unknown regression function $f(\cdot)$, e.g., $f(x) = x - x^0$, $f(x) = \sin(\|x - x^0\|)$, $f(x) = e^x \|x - x^0\|^3$, etc. Since the selection of $\{a_k\}$ and $f(\cdot)$ is at the user's disposal, satisfaction of A1-A3 in 3.2 is not a problem when SAAWET is applied. The observation y_{k+1} at time $k + 1$ means the information available at time $k + 1$ about x^0 including the estimate x_k for x^0 at time k . It is clear that the observation y_{k+1} can always be written in the standard form (1) by setting $\epsilon_{k+1} \triangleq y_{k+1} - f(x_k)$.

Because of arbitrary selection of $f(\cdot)$, the error ϵ_{k+1} contains not only random noises but also structural errors, which cannot be expected to have a nice probabilistic property. However, the noise condition A4 can be verified for a large class of problems from systems and control, and the desired convergence results are obtained in the following cases:

In system identification the system coefficients and orders are estimated as parameters (Chen 2010, 2002; Mu and Chen 2014);

The values of a nonlinear function at any fixed points are treated as parameters to be estimated (Greblicki 2002; Mu and Chen 2013; Zhao et al. 2017; Chen and Zhao 2014);

For some control problems, in particular, for adaptive regulation, the optimal control performance is achieved at a time-invariant control, which serves as parameter to be estimated (Chen 2002; Chen and Zhao 2014).

This approach works well for some other problems described in Chen (2002) and Chen and Zhao (2014).

The key points of verification of A4 are to show that along any convergent subsequence $\{x_{n_k}\}$, $\{x_k\}$ in any small neighborhood of x_{n_k} suffer no truncation, and they are close to each other.

Other Applications

SA is well applied to solving problems from signal processing and communication. In fact, by using SA, the consensus is achieved for a networked nonlinear systems (Feng and Chen 2019); the adaptive filtering is realized by a sign algorithm (Chen and Yin 2003); the page

rank problem, adaptive pole assignment, blind identification, the principal component analysis (PCA) problem, and other problems are also appropriately solved (Chen 2002; Chen and Zhao 2014).

Summary and Future Directions

The RM algorithm and some of its modifications with their convergence properties are presented in the entry. They, in particular SAAWET, have successfully been applied to solving problems from systems and control and other related areas under reasonable conditions. It is natural to expect that more problems from various fields can be dealt with by using SAAWET. Concerning SAAWET itself, it is expected to consider how to prevent the algorithm from being stuck at an undesired point, how to reduce the number of truncations, and how to accelerate the convergence.

Cross-References

- [Distributed Estimation in Networks](#)
- [Nonlinear System Identification: An Overview of Common Approaches](#)
- [System Identification: An Overview](#)

Recommended Reading

Chen (2002), Kushner and Yin (2003) and Nevelson and Khasminskii (1976)

Bibliography

- Benaïm M (1996) A dynamical system approach to stochastic approximations. *SIAM J Control Optim* 34(2):437–472
- Benveniste A, Metivier M, Priouret P (1990) Adaptive algorithms and stochastic approximation. Springer, New York
- Bianchi P, Fort G, Hachem W (2013) Performance of a distributed stochastic approximation algorithm. *IEEE Trans Inf Theory* 59(11):405–418
- Borkar VS (1998) Asynchronous stochastic approximation. *SIAM J Control Optim* 36:840–851
- Chen HF (2002) Stochastic approximation and its applications. Kluwer, Dordrecht
- Chen HF (2010) New approach to recursive identification for ARMAX systems. *IEEE Trans Autom Control* 55(4):879–886
- Chen HF, Zhao WX (2014) Recursive identification and parameter estimation. CRC Press, Taylor & Francis, Boca Raton
- Chen HF, Duncan T, Pasik-Duncan B (1999) A Kiefer-Wolfowitz algorithm with randomized differences. *IEEE Trans Autom Control* 44(3):42–453
- Chen HF, Yin G (2003) Asymptotic properties of sign algorithms for adaptive filtering. *IEEE Trans Autom Control* 48(9):1545–1556
- Chen HF, Zhu YM (1986) Stochastic approximation procedures with randomly varying truncations. *Scientia Sinica (Series A)* 29(9):914–926
- Fang HT, Chen HF (2000) An a.s. convergent algorithm for global optimization with noise corrupted observations. *J Optim Appl* 104:43–376
- Fang HT, Chen HF (2001) Asymptotic behavior of asynchronous stochastic approximation. *Sci China (Series F)* 44:249–258
- Feng WH, Chen HF (2019) Output consensus of networked Hammerstein and Wiener systems. *SIAM J Control Optim* 57(2):1230–1254
- Goodfellow I, Bengio Y, Courville A (2016) Deep learning. The MIT Press, Cambridge
- Greblicki W (2002) Stochastic approximation in nonparametric identification of Hammerstein systems. *IEEE Trans Automat Contr* 47(11):1800–1810
- Kiefer J, Wolfowitz J (1952) Stochastic estimation of the maximum of a regression function. *Ann Math Stat* 23:462–466
- Kushner HJ, Clark DS (1978) Stochastic approximation for constrained and unconstrained systems. Springer, New York
- Kushner HJ, Yin G (2003) Stochastic approximation algorithms and applications. Springer, New York
- Lei JL, Chen HF (2020) Distributed stochastic approximation algorithm with expanding truncations: algorithm and applications. *IEEE Trans Autom Control*. Digital Object Identifier: 10.1109/TAC.2019.2912713
- Lei JL, Chen HF, Fang HT (2018) Asymptotic properties of primal-dual algorithm for distributed stochastic optimization over random networks with imperfect communications. *SIAM J Control Optim* 56(3):2157–2188
- Ljung L (1977) Analysis of recursive stochastic algorithms. *IEEE Trans Autom Control* 22:551–575
- Mu BQ, Chen HF (2013) Recursive identification of MIMO Wiener systems. *IEEE Trans Autom Control* 58(3):802–808
- Mu BQ, Chen HF (2014) Recursive identification of errors-in-variables Wiener-Hammerstein systems. *Eur J Control* 20:4–23
- Nevelson MB, Khasminskii RZ (1976) Stochastic approximation and recursive estimation. Translation of mathematical monographs, vol 47. American Mathematical Society, Providence

- Polyak BT, Judisky AB (1992) Acceleration of stochastic approximation by averaging. *SIAM J Control Optim* 30:838–855
- Robbins H, Monro S (1951) A stochastic approximation method. *Ann Math Stat* 22:400–407
- Spall JC (1992) Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans Autom Control* 37:31–341
- Tsitsiklis JN (1994) Asynchronous stochastic approximation and Q-learning. *Mach Learn* 16:185–202
- Wang JJ, Chong EKP, Kulkarni SR (1996) Equivalent necessary and sufficient conditions on noise sequences for stochastic approximation algorithms. *Adv Appl Probab* 28:784–801
- Zhao WX, Chen HF, Tempo R, Dabbene F (2017) Recursive nonparametric identification of nonlinear systems with adaptive binary sensors. *IEEE Trans Autom Control* 62(8):959–971

Stochastic Description of Biochemical Networks

João P. Hespanha¹ and Mustafa Khammash²

¹Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA, USA

²Department of Biosystems Science and Engineering, Swiss Federal Institute of Technology at Zurich (ETHZ), Basel, Switzerland

Abstract

Conventional deterministic chemical kinetics often breaks down in the small volume of a living cell where cellular species (e.g., genes, mRNAs, etc.) exist in discrete, low copy numbers and react through reaction channels whose timing and order is random. In such an environment, a stochastic chemical kinetics framework that models species abundances as discrete random variables is more suitable. The resulting models consist of continuous-time discrete-state Markov chains. Here we describe how such models can be formulated and numerically simulated, and we present some of the key analysis techniques for studying such reactions.

Keywords

Chemical master equation · Gillespie algorithm · Moment dynamics · Stochastic biochemical reactions · Stochastic models

Introduction

The time evolution of a spatially homogeneous mixture of chemically reacting molecules is often modeled using a stochastic formulation, which takes into account the inherent randomness of thermal molecular motion. This formulation is important when modeling complex reactions inside living cells, where small populations of key reactants can set the stage for significant stochastic effects. In this entry, we review the basic stochastic model of chemical reactions and discuss the most common techniques used to simulate and analyze this model.

Stochastic Models of Chemical Reactions

We start by considering a set of N molecular species (reactants) $S_1, \dots, S_N$ that are confined to a fixed volume Ω . These species react through M possible reactions $R_1, \dots, R_M$. In this formulation of chemical kinetics, we shall assume that the system is in thermal equilibrium and is well mixed. Thus, the reacting molecules move due to their thermal energy. The population of the different reactants is described by a random process $X(t) = (X_1(t) \dots X_N(t))^T$, where $X_i(t)$ is a random variable that models the abundance (in terms of the number of copies) of molecules of species S_i in the system at time t . For the allowable reactions, we shall only consider elementary reactions. These could either be monomolecular, $S_i \rightarrow$ products, or bimolecular, $S_i + S_j \rightarrow$ products. Upon the firing of reaction R_k , a transition occurs from some state $X = \mathbf{x}_i$ right before the reaction fires to some other state $X = \mathbf{x}_i + \mathbf{s}_k$, which reflects the change in the population immediately after the reaction has fired. $\mathbf{s}_k$ is referred to as the *stoichiometric vector*. The set

Stochastic Description of Biochemical Networks, Table 1 Propensity functions for elementary reactions. The constants c , c' , and c'' are related to k , k' , and k'' , the reaction rate constants from *deterministic* mass-action kinetics. Indeed it can be shown that $c = k$, $c' = k'/\Omega$, and $c'' = 2k''/\Omega$

Reaction type	Propensity function
$S_i \rightarrow \text{Products}$	$c x_i$
$S_i + S_j \rightarrow \text{Products} \quad (i \neq j)$	$c' x_i x_j$
$S_i + S_i \rightarrow \text{Products}$	$c'' x_i (x_i - 1)/2$

of allowable M reactions defines the so-called stoichiometry matrix:

$$S = [s_1 \cdots s_M].$$

To each reaction R_k , we associate a *propensity function*, $w_k(\mathbf{x})$ that describes the rate of that reaction. More precisely, $w_k(\mathbf{x})h$ is the probability that, given the system is in state $\mathbf{x}$ at time t , R_k fires once in the time interval $[t, t + h)$. The propensity functions for elementary reactions is given in Table 1.

Limiting to the Deterministic Regime

There is an important connection between the stochastic process $X(t)$, as represented by the continuous-time discrete-state Markov chain described above, and the solution of a related deterministic reaction rate equations obtained from mass-action kinetics. To see this, let $\Phi(t) = [\Phi_1(t), \dots, \Phi_N(t)]^T$ be the vector concentrations of species $S_1, \dots, S_N$. According to mass-action kinetics, $\Phi(\cdot)$ satisfies the ordinary differential equation:

$$\dot{\Phi} = Sf(\Phi(t)), \quad \Phi(0) = \Phi_0.$$

In order to compare the $\Phi(t)$ with $X(t)$, which represents molecular counts, we divide $X(t)$ by the reaction volume to get $X^\Omega(t) = X(t)/\Omega$. It turns out that $X^\Omega(t)$ *limits* to $\Phi(t)$: According to Kurtz (Ethier and Kurtz 1986), for every $t \geq 0$:

$$\lim_{\Omega \rightarrow \infty} \sup_{s \leq t} |X^\Omega(s) - \Phi(s)| = 0, \quad \text{almost surely.}$$

Hence, over any finite time interval, the stochastic model *converges* to the deterministic mass-action one in the thermodynamic limit. Note that this is only a large volume limit result. In practice, for a fixed volume, a stochastic description may differ considerably from the deterministic description.

Stochastic Simulations

Gillespie's stochastic simulation algorithm (SSA) constructs sample paths for the random process $X(t) = (X_1(t) \dots X_N(t))^T$ that are consistent with the stochastic model described above (Gillespie 1976). It consists of the following basic steps:

1. Initialize the state $X(0)$ and set $t = 0$.
2. Draw a random number $\tau \in (0, \infty)$ with exponential distribution and mean equal to $1/\sum_k w_k(X(t))$.
3. Draw a random number $k \in \{1, 2, \dots, M\}$ such that the probability of $k = i \in \{1, 2, \dots, M\}$ is proportional to $w_i(X(t))$.
4. Set $X(t + \tau) = X(t) + s_k$ and $t = t + \tau$.
5. Repeat from (2) until t reaches the desired simulation time.

By running this algorithm multiple times with independent random draws, one can estimate the distribution and statistical moments of the random process $X(t)$.

The Chemical Master Equation (CME)

The *chemical master equation* (CME), also known as the forward Kolmogorov equation, describes the time evolution of the probability that the system is in a given state $\mathbf{x}$. The CME can be derived based on the Markov property of chemical reactions. Suppose the system is in state $\mathbf{x}$ at time t . Within an error of order $\mathcal{O}(h^2)$, the following statements apply:

- The probability that an R_k reaction fires exactly once in the time interval $[t, t + h)$ is given by $w_k(\mathbf{x})h$.

- The probability that no reactions fire in the time interval $[t, t + h)$ is given by $1 - \sum_k w_k(\mathbf{x})dx$.
- The probability that more than one reaction fires in the time interval $[t, t + h)$ is zero.

Let $P(\mathbf{x}, t)$, denote the probability that the system is in state $\mathbf{x}$ at time t . We can express $P(\mathbf{x}, t + h)$ as follows:

$$P(\mathbf{x}, t + h) = P(\mathbf{x}, t) \left(1 - \sum_k w_k(\mathbf{x})h \right) + \sum_k P(\mathbf{x} - \mathbf{s}_k, t) w_k(\mathbf{x} - \mathbf{s}_k)h + \mathcal{O}(h^2).$$

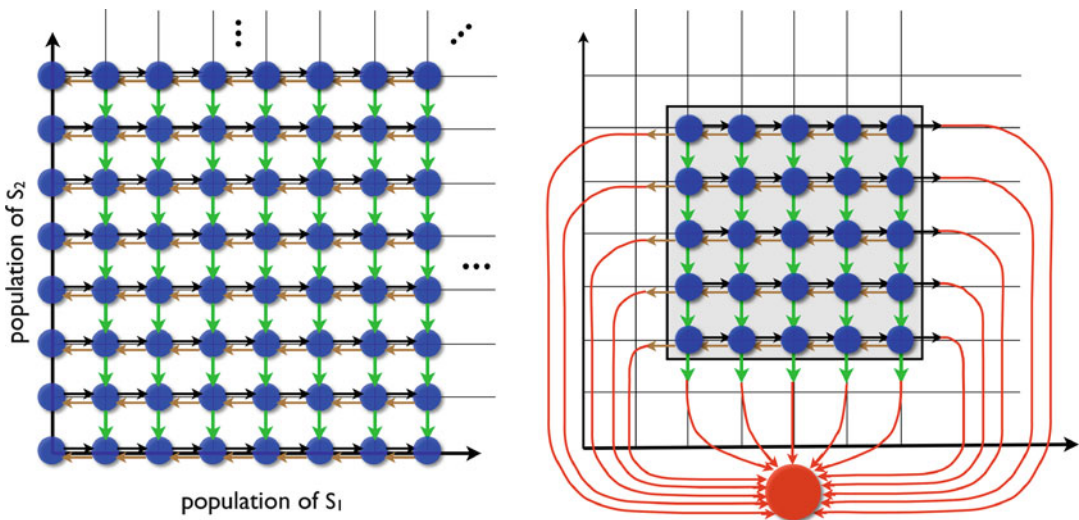
The first term on the right-hand side is the probability that the system is already in state $\mathbf{x}$ at time t , and no reactions occur in the next h . In the second term on the right-hand side, the k th term in the summation is the probability that the system at time t is an R_k reaction away from being at state $\mathbf{x}$ and that an R_k reaction takes place in the next h .

Moving $P(\mathbf{x}, t)$ to the left-hand side, dividing by h , and taking the limit as h goes to zero yields the chemical master equation (CME):

$$\frac{dP(\mathbf{x}, t)}{dt} = \sum_{k=1}^M \left(w_k(\mathbf{x} - \mathbf{s}_k) P(\mathbf{x} - \mathbf{s}_k, t) - w_k(\mathbf{x}) P(\mathbf{x}, t) \right). \quad (1)$$

The CME defines a linear dynamical system in the probabilities of the different states (each state is defined by a specific number of molecules of each of the species). However, there are generally an infinite number of states, and the resulting infinite linear system is not directly solvable. One approach to overcome this difficulty is to approximate the solution of the CME by truncating the states. A particular truncation procedure that gives error bounds is called the finite-state projection (FSP) (Munsky and Khammash 2006). The key idea behind the FSP approach is to keep those states that support the bulk of the probability distribution while projecting the remaining infinite states onto a single “absorbing” state. See Fig. 1.

The left panel in the figure shows the infinite states of a system with two species. The arrows indicate transitions among states caused by allowable chemical reactions. The underlying stochastic process is a continuous-time discrete-state Markov process. The right panel shows the projected (finite-state) system for a specific projection region (box). The projection is obtained



Stochastic Description of Biochemical Networks, Fig. 1 The finite-state projection

as follows: transitions within the retained states are kept, while transitions that emanate from these states and end at states outside the box are channeled to a single new absorbing state. Transitions into the box are deleted. The resulting projected system is a finite-state Markov process. The probability of each of its finite states can be computed exactly. It can be shown that the truncation, as defined here, gives a lower bound for the probability for the original full system. The FSP algorithm provides a way for constructing an approximation of the CME that satisfies any prespecified accuracy requirement.

Moment Dynamics

While the probability distribution $P(\mathbf{x}, t)$ provides great detail on the state $\mathbf{x}$ at time t , often statistical moments of the molecule copy numbers already provide important information about their variability, which motivates the construction of mathematical models for the evolution of such models over time.

Given a vector of integers $\mathbf{m} := (m_1, m_2, \dots, m_n)$, we use the notation $\mu^{(\mathbf{m})}$ to denote the following uncentered moment of X :

$$\mu^{(\mathbf{m})} := \mathbb{E}[X_1^{m_1} X_2^{m_2} \dots X_n^{m_n}].$$

Such moment is said to be of order $\sum_i m_i$. With N species, there are exactly N first-order moments $\mathbb{E}[X_i]$, $\forall i \in \{1, 2, \dots, N\}$, which are just the means; $N(N-1)/2$ second-order moments $\mathbb{E}[X_i^2]$, $\forall i$ and $\mathbb{E}[X_i X_j]$, $\forall i \neq j$, which can be used to compute variances and covariance; $N(N-1)(N-2)/6$ third-order moments; and so on.

Using the CME (1), one can show that

$$\begin{aligned} \frac{d\mu^{(\mathbf{m})}}{dt} = & \mathbb{E} \left[\sum_k w_k(X) \left((X_1 + s_{1,k})^{m_1} (X_2 - s_{2,k})^{m_2} \right. \right. \\ & \left. \left. \dots (X_N - s_{N,k})^{m_N} - X_1^{m_1} X_2^{m_2} \dots X_N^{m_N} \right) \right], \end{aligned}$$

and, because the propensity functions are all polynomials on $\mathbf{x}$ (cf. Table 1), the expected value in the right-hand side can actually be written as a

linear combination of other uncentered moments of X . This means that if we construct a vector μ containing all the uncentered moments of x up to some order k , the evolution of μ is determined by a differential equation of the form

$$\frac{d\mu}{dt} = A\mu + B\bar{\mu}, \quad \mu \in \mathbb{R}^K, \quad \bar{\mu} \in \mathbb{R}^{\bar{K}} \quad (2)$$

where A and B are appropriately defined matrices and $\bar{\mu}$ is a vector containing moments of order larger than k . The equation (2) is exact, and we call it the (*exact*) k -order moment dynamics, and the integer k is called the *order of truncation*. Note that the dimension K of (2) is always larger than k since there are many moments of each order. In fact, in general, K is of order n^k .

When all chemical reactions have only one reactant, the term $B\bar{\mu}$ does not appear in (2), and we say that the exact moment dynamics are *closed*. However, when at least one chemical reaction has two or more reactants, then the term $B\bar{\mu}$ appears, and we say that the moment dynamics are *open* since (2) depends on the moments in $\bar{\mu}$, which are not part of the state μ . When all chemical reactions are elementary (i.e., with at most two reactants), then all moments in $\bar{\mu}$ are exactly of order $k+1$.

Moment closure is a procedure by which one approximates the exact (but open) moment dynamics (2) by an approximate (but now closed) equation of the form

$$\dot{v} = Av + B\varphi(v), \quad v \in \mathbb{R}^K \quad (3)$$

where $\varphi(v)$ is a column vector that approximates the moments in $\bar{\mu}$. The function $\varphi(v)$ is called the moment closure function, and (3) is called the *approximate k th-order moment dynamics*. The goal of any moment closure method is to construct $\varphi(v)$ so that the solution v to (3) is close to the solution μ to (2).

There are three main approaches to construct the moment closure function $\varphi(\cdot)$:

1. *Matching-based methods* directly attempt to match the solutions to (2) and (3) (e.g., Singh and Hespanha 2011).

2. *Distribution-based methods* construct $\varphi(\cdot)$ by making reasonable assumptions on the statistical distribution of the molecule counts vector x (e.g., Gomez-Uribe and Verghese 2007).
3. *Large volume methods* construct $\varphi(\cdot)$ by assuming that reactions take place on a large volume (e.g., Van Kampen 2001).

It is important to emphasize that this classification is about methods to *construct* moment closure. It turns out that sometimes different methods lead to the same moment closure function $\varphi(\cdot)$.

Conclusion and Outlook

We have introduced complementary approaches to study the evolution of biochemical networks that exhibit important stochastic effects.

Stochastic simulations permit the construction of sample paths for the molecule counts, which can be averaged to study the ensemble behavior of the system. This type of approach scales well with the number of molecular species, but can be computationally very intensive when the number of reactions is very large. This challenge has led to the development of approximate stochastic simulation algorithms that attempt to simulate multiple reactions in the same simulation step (e.g., Rathinam et al. 2003).

Solving the CME provides the most detailed and accurate approach to characterize the ensemble properties of the molecular counts, but for most biochemical systems such solution cannot be found in closed form, and numerical methods scale exponentially with the number of species. This challenge has led to the development of algorithms that compute approximate solutions to the CME, e.g., by aggregating states with low probability, while keeping track of the error (e.g., Munsky and Khammash 2006).

Moment dynamics is attractive in that the number of k th-order moments only scales polynomially with the number of chemical species, but one only obtains closed dynamics for very simple biochemical networks. This limitation has led to the development of moment closure tech-

niques to approximate the open moment dynamics by a closed system of ordinary differential equations.

Cross-References

- [Deterministic Description of Biochemical Networks](#)
- [Robustness Analysis of Biological Models](#)
- [Spatial Description of Biochemical Networks](#)

Bibliography

- Ethier SN, Kurtz TG (1986) Markov processes: characterization and convergence. Wiley series in probability and mathematical statistics: probability and mathematical statistics. Hoboken, New Jersey
- Gillespie DT (1976) A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. J Comput Phys 22:403–434
- Gomez-Uribe CA, Verghese GC (2007) Mass fluctuation kinetics: capturing stochastic effects in systems of chemical reactions through coupled mean-variance computations. J Chem Phys 126(2):024109–024109–12
- Munsky B, Khammash M (2006) The finite state projection algorithm for the solution of the chemical master equation. J Chem Phys 124:044104
- Rathinam M, Petzold LR, Cao Y, Gillespie DT (2003) Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method. J Chem Phys 119(24):12784–12794
- Singh A, Hespanha JP (2011) Approximate moment dynamics for chemically reacting systems. IEEE Trans Autom Control 56(2):414–418
- Van Kampen NG (2001) Stochastic processes in physics and chemistry. Elsevier Science, Amsterdam

Stochastic Dynamic Programming

Qing Zhang

Department of Mathematics, The University of Georgia, Athens, GA, USA

Abstract

This article is concerned with one of the traditional approaches for stochastic control prob-

lems: Stochastic dynamic programming. Brief descriptions of stochastic dynamic programming methods and related terminology are provided. Two asset-selling examples are presented to illustrate the basic ideas. A list of topics and references are also provided for further reading.

Keywords

Asset-selling rule · Bellman equation · Hamilton-Jacobi-Bellman equation · Markov decision problem · Optimality principle · Stochastic control · Viscosity solution

Introduction

The term *dynamic programming* was introduced by Richard Bellman in the 1940s. It refers to a method for solving dynamic optimization problems by breaking them down into smaller and simpler subproblems.

To solve a given problem, one often needs to solve each part of the problem (subproblems) and then put together their solutions to obtain an overall solution. Some of these subproblems are of the same type. The idea behind the dynamic programming approach is to solve each subproblem only once in order to reduce the overall computation.

The cornerstone of dynamic programming (DP) is the so-called principle of optimality which is described by Bellman in his 1957 book (Bellman 1957):

Principle of Optimality: An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

This principle of optimality gives rise to DP (or optimality) equations, which are referred to as Bellman equations in discrete-time optimization problems or Hamilton-Jacobi-Bellman (HJB) equations in continuous-time ones. Such equations provide a necessary condition for optimality in terms of the value of the underlying decision problem. By and large, an optimal control policy

in most cases can be obtained by solving the associated Bellman (HJB) equation. In view of this, dynamic programming is a powerful tool for a broad range of control and decision-making problems. When the underlying system is driven by certain type of random disturbance, the corresponding DP approach is referred to as *stochastic dynamic programming*.

Terminology

The following concepts are often used in stochastic dynamic programming.

An **objective function** describes the objective of a given optimization problem (e.g., maximizing profits, minimizing cost, etc.) in terms of the states of the underlying system, decision (control) variables, and possible random disturbance.

State variables represent the information about the current system under consideration. For example, in a manufacturing system, one needs to know the current product inventory in order to decide how much to produce at the moment. In this case, the inventory level would be one of the state variables.

The variables chosen at any time are called the decision or **control variables**. For instance, the rate of production over time in the manufacturing system is a control variable. Typically, control variables are functions of state variables. They affect the future states of the system and the objective function.

In stochastic control problems, the system is also affected by random events (noise). Such noise is referred to system **disturbance**. The noise is often not available a priori. Only their probabilistic distributions are known.

The goal of the optimization problem is to choose control variables over time so as to either maximize or minimize the corresponding objective function. For example, in order to maximize the overall profits, a manufacturing firm has to decide how much to produce over time so as to maximize the revenue by meeting the product demand and minimize the costs associated with inventory. The best possible value of the objective

is called **value function**, which is given in terms of the state variables.

In the next two sections, we give two examples to illustrate how stochastic DP methods are used in discrete and continuous time.

An Asset-Selling Example (Discrete Time)

Consider a person wants to sell an asset (e.g., a car or a house). She is offered an amount of money every period (say, a day). Let $v_0, v_1, \dots, v_{N-1}$ denote the amount of these random offers. Assume they are independent and identically distributed. At the end of each period, the person has to decide whether to accept the offer or reject it. If she accepts the offer, she can put the money in a bank account and receive a fixed interest rate $r > 0$; if she rejects the offer, she waits till the next period. Rejected offers cannot be recycled. In addition, she has to sell her asset by the end of the N th period and accept the last offer v_{N-1} if all previous offers have been rejected. The goal is to decide when to accept an offer to maximize the overall return at the N th period.

In this example, for each k , v_k is the random disturbance. The control variables u_k take values in $\{\text{sell, hold}\}$. The state variables x_k are given by the equations

$$x_0 = 0; \quad x_{k+1} = \begin{cases} \text{sold} & \text{if } u_k = \text{sell} \\ v_k & \text{otherwise.} \end{cases}$$

Let

$$h_N(x_N) = \begin{cases} x_N & \text{if } x_N \neq \text{sold,} \\ 0 & \text{otherwise.} \end{cases}$$

$$h_k(x_k, u_k, v_k) = \begin{cases} (1+r)^{N-k}x_k & \text{if } x_k \neq \text{sold} \\ & \text{and } u_k = \text{sell} \\ 0 & \text{otherwise.} \end{cases}$$

for $k = 0, 1, \dots, N-1$.

Then, the payoff function is given by

$$E_{\{v_k\}} \left(h_N(x_N) + \sum_{k=0}^{N-1} h_k(x_k, u_k, v_k) \right).$$

Here, $E_{\{v_k\}}$ represents the expected value over $\{v_k\}$. The corresponding value functions $V_k(x_k)$ satisfy the following Bellman equations:

$$V_N(x_N) = \begin{cases} x_N & \text{if } x_N \neq \text{sold,} \\ 0 & \text{otherwise.} \end{cases}$$

$$V_k(x_k) = \begin{cases} \max \left((1+r)^{N-k}x_k, E V_{k+1}(v_k) \right) & \text{if } x_k \neq \text{sold} \\ 0 & \text{otherwise.} \end{cases}$$

for $k = 0, 1, \dots, N-1$.

The optimal selling rule can be given as (assuming $x_k \neq \text{sold}$) (see Bertsekas 1987):

accept the offer

$$v_{k-1} = x_k \text{ if } (1+r)^{N-k}x_k \geq E V_{k+1}(v_k),$$

reject the offer

$$v_{k-1} = x_k \text{ if } (1+r)^{N-k}x_k < E V_{k+1}(v_k).$$

Given the distribution for v_k , one can compute V_k backwards and solve the Bellman equations, which in turn leads to the above optimal selling rule.

Note that such backward iteration only works with finite horizon dynamic programming. When working with an infinite horizon (discounted or long-run average) payoff function, often used methods are value iteration (successive approxi-

mation) and policy iteration. The idea is to construct a sequence of functions recursively so that they converge pointwise to the value function. For description of these iteration methods, their convergence properties, and error bound analysis, we refer the reader to Bertsekas (1987).

Next, we consider a continuous-time asset-selling problem.

An Asset-Selling Example (Continuous Time)

Suppose a person wants to sell her asset. The price x_t at time $t \in [0, \infty)$ of her asset is given by a stochastic differential equation

$$\frac{dx_t}{x_t} = \mu dt + \sigma dw_t,$$

where μ and σ are known constants and w_t is the standard Brownian motion representing the disturbance. Suppose the transaction cost is K and the discount rate r . She has to decide when to sell her asset to maximize an expected return. In this example, the state variable is price x_t , control variable is a function of selling time τ , and the payoff function is given by

$$J(x, \tau) = Ee^{-r\tau}(x_\tau - K).$$

Let $V(x)$ denote the value function, i.e., $V(x) = \sup_\tau J(x, \tau)$. Then the associate HJB equation is given by

$$\min \left\{ rV(x) - x\mu \frac{dV(x)}{dx} - \frac{x^2\sigma^2}{2} \frac{d^2V(x)}{dx^2}, \right. \\ \left. V(x) - K \right\} = 0. \quad (1)$$

Let

$$x^* = \frac{K\beta}{\beta - 1},$$

where

$$\beta = \frac{1}{\sigma^2} \left(\frac{\sigma^2}{2} - \mu + \sqrt{\left(\mu - \frac{\sigma^2}{2} \right)^2 + 2r\sigma^2} \right).$$

Then the optimal selling rule can be given as (see Øksendal 2007):

$$\begin{cases} \text{sell} & \text{if } x_t \geq x^*, \\ \text{hold} & \text{if } x_t < x^*. \end{cases}$$

In general, to solve an optimal control problem via the DP approach, one first needs to solve the associate Bellman (HJB) equations. Then, these solutions can be used to come up with an optimal control policy. For example, in the above case, given the value function $V(x)$, one should hold if

$$rV(x) - x\mu \frac{dV(x)}{dx} - \frac{x^2\sigma^2}{2} \frac{d^2V(x)}{dx^2} = 0$$

and sell when $V(x) - K = 0$. The threshold level x^* is the exact dividing point between the first part equals zero and the second part vanishes. In addition, one can also provide a theoretical justification in terms of a verification theorem to show that the solution obtained this way is indeed optimal (see Fleming and Rishel (1975), Fleming and Soner (2006), or Yong and Zhou (1999)).

HJB Equation Characterization and Computational Methods

In continuous-time optimal control problem, one major difficulty that arises in solving the associated HJB equations (e.g., (1)) is the characterization of the solutions. In most cases, there is no guarantee that the derivatives or partial derivatives exist. In this connection, the concept of viscosity solutions developed by Crandall and Lions in the 1980s can often be used to characterize the solutions and their uniqueness. We refer the reader to Fleming and Soner (2006)

for related literature and applications. In addition, we would like to point out that closed-form solutions are rare in stochastic control theory and difficult to obtain in most cases. In many applications, one needs to resort to computational methods. One typical way to solve an HJB equation is the finite difference methods. An alternative is Kushner's Markov chain approximation methods; see Kushner and Dupuis (1992).

Summary and Future Directions

In this article, we have briefly stated stochastic DP methods, showed how they work in two simple examples, and discussed related issues. One serious limitation of the DP approach is the so-called curse of dimensionality. In other words, the DP does not work for problems with high dimensionality. Various efforts have been devoted to search for approximate solutions. One approach developed in recent years is the multi-time-scale approach. The idea is to classify random events according to the frequency of their occurrence. Frequent occurring events are grouped together and treated as a single "state" to achieve the reduction of dimensionality. We refer the reader to Yin and Zhang (2005, 2013) for related literature and theoretical development. Finally, we would like to mention that stochastic DP has been used in many applications in economics, engineering, management science, and finance. Some applications can be found in Sethi and Thompson (2000). Additional references are also provided at the end for further reading.

Cross-References

- [Backward Stochastic Differential Equations and Related Control Problems](#)
- [Numerical Methods for Continuous-Time Stochastic Control Problems](#)

- [Risk-Sensitive Stochastic Control](#)
- [Stochastic Adaptive Control](#)
- [Stochastic Linear-Quadratic Control](#)
- [Stochastic Maximum Principle](#)

Bibliography

- Bellman RE (1957) Dynamic programming. Princeton University Press, Princeton
- Bertsekas DP (1987) Dynamic programming. Prentice Hall, Englewood Cliffs
- Davis MHA (1993) Markov models and optimization. Chapman & Hall, London
- Elliott RJ, Aggoun L, Moore JB (1995) Hidden Markov models: estimation and control. Springer, New York
- Fleming WH, Rishel RW (1975) Deterministic and stochastic optimal control. Springer, New York
- Fleming WH, Soner HM (2006) Controlled Markov processes and viscosity solutions, 2nd edn. Springer, New York
- Hernandez-Lerma O, Lasserre JB (1996) Discrete-time Markov control processes: basic optimality criteria. Springer, New York
- Kushner HJ, Dupuis PG (1992) Numerical methods for stochastic control problems in continuous time. Springer, New York
- Kushner HJ, Yin G (1997) Stochastic approximation algorithms and applications. Springer, New York
- Øksendal B (2007) Stochastic differential equations, 6th edn. Springer, New York
- Pham H (2009) Continuous-time stochastic control and optimization with financial applications. Springer, New York
- Sethi SP, Thompson GL (2000) Optimal control theory: applications to management science and economics, 2nd edn. Kluwer, Boston
- Sethi SP, Zhang Q (1994) Hierarchical decision making in stochastic manufacturing systems. Birkhäuser, Boston
- Yin G, Zhang Q (2005) Discrete-time Markov chains: two-time-scale methods and applications. Springer, New York
- Yin G, Zhang Q (2013) Continuous-time Markov chains and applications: a two-time-scale approach, 2nd edn. Springer, New York
- Yin G, Zhu C (2010) Hybrid switching diffusions: properties and applications. Springer, New York
- Yong J, Zhou XY (1999) Stochastic control: Hamiltonian systems and HJB equations. Springer, New York

Stochastic Fault Detection

Aiping Wang¹ and Hong Wang^{2,3}

¹Department of Electronic and Information Engineering, Bozhou University, Bozhou, Anhui, PR China

²Energy and Transportation Science, Oak Ridge National Laboratory, Oak Ridge, TN, USA

³The University of Manchester, Manchester, UK

Abstract

This entry describes the state-of-the-art and future perspectives on stochastic fault detection, namely, stochastic fault detection and diagnosis (FDD). Both model-based and data-driven FDD methods for stochastic signals and systems have been included, where the use of hypothesis testing, Kalman filtering, system estimation, principal component analysis (PCA), and stochastic distribution control has been discussed for the construction of effective FDD algorithms. Indeed, stochastic FDD constitute an important and integrated part in developing fault-tolerant controls (FTC) for guaranteed safe operation of control systems, of which increased penetration of random factors is inevitable nowadays.

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the US Department of Energy. The US government retains, and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript or allow others to do so, for US government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>.)

Keywords

Hypothesis testing · Fault detection and diagnosis (FDD) · System identification · Multivariable statistics · Stochastic distribution controls

Introduction

Stochastic systems are everywhere. Examples are most of industrial processes, power grid, transportation systems, etc. For example, due to an increased penetration of renewables (wind, solar, and distributed energy storages, etc.) that are random in nature, control systems in power grid operation are now subjected to ever-increased degrees of randomness and uncertainties. This makes some grid key variables such as frequency, voltage, and power flow random and mostly even non-Gaussian. For industrial processes, the increased use of low-cost raw materials leads to large random variations and uncertainties in their compositions. This, together with variations of environments under which control systems operate, increases the degree of randomness for the whole system. During the operation of such systems, the effect of these random factors would inevitably affect their operational performance. As a result, system dynamics and most of the variables in these control systems exhibit stochastic nature.

For these systems, one of the important issues is its operational safety, and the challenge has always been how faults can be effectively detected and diagnosed so that proper actions can be taken to prevent the system from severe consequences caused by faults. In this regard, FDD play a key role. An earlier and accurate FDD can allow us to realize effective fault-tolerant control that ensures safe operation of the system after fault has occurred. In this context, FDD for stochastic systems have been studied for a long time and are still an important topic of research in the future.

Objective and General Structure

The objective of fault detection is to use available information such as various measured signals (data) from the system to detect the fault in the system, where the fault is regarded as unexpected changes during the system operation. Fault can happen in various parts of the system. It can be actuator, sensor, and system faults (Chen and Patton 2012; Isermann and Munchhof 2010). Also, it can be unexpected changes of statistics (mean and variance) of a signal or some system parameters (Basseville and Nikiforov 1993). Once fault is detected, fault diagnosis can be performed to find out its location and size. This defines the functionalities of FDD, which are applicable to any systems rather than just to stochastic systems. Here only stochastic systems and signals will be considered.

Since the system is stochastic in nature and the available information used for FDD are represented as random processes, tools such as hypothesis testing, filtering, system estimation, multivariable statistics, stochastic estimation theory, and stochastic distribution control have been developed in the past decades. These methods have been widely and successfully applied to perform FDD for stochastic systems.

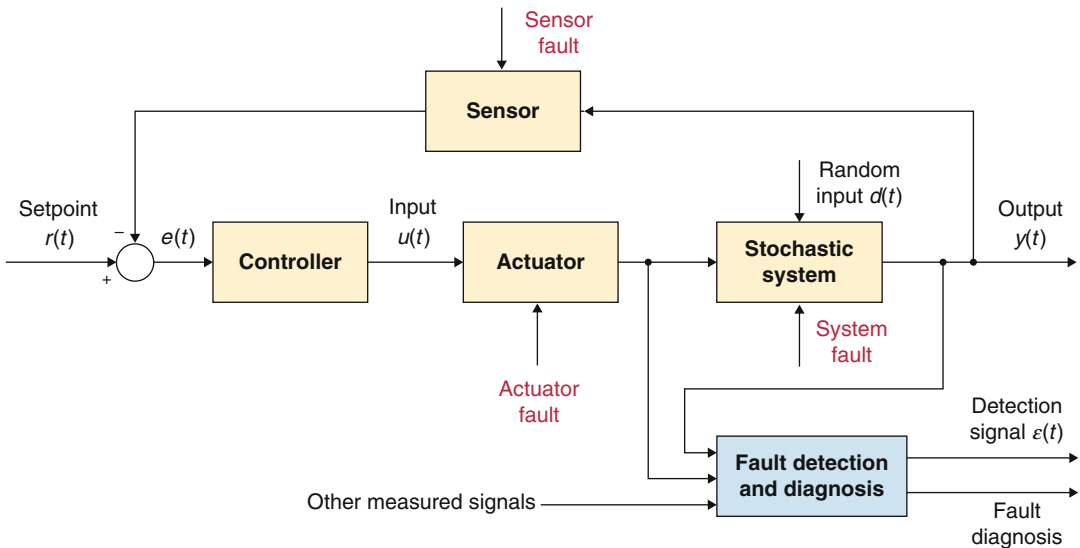
In this regard, the following figure shows a generic structure of FDD for stochastic systems, where all the signals are function of time and some of them are random processes.

Using the notations in Fig. 1, the FDD for stochastic systems means to perform the following task:

$$\begin{aligned} & \text{Use } \{u(t), y(t), \text{other measured signals}\} \\ & \rightarrow \text{fault detection and diagnosis} \quad (1) \end{aligned}$$

The Existing Approaches

The earlier days of research in this area have largely been focused on developing algorithms for unexpected change detection of random signals and filtering-based fault detection and diagnosis for dynamic stochastic systems. The well-known Kalman filtering algorithms have been used to generate fault detection signals which can be used to activate an alarm when fault occurs. The idea is to use time-domain signals such as system inputs and outputs to constitute FDD algorithms (Fig. 1). Recently, further development has been seen for stochastic distribution systems, where FDD are performed using available system measurements on the probability



Stochastic Fault Detection, Fig. 1 A general structure for fault detection and diagnosis for stochastic systems

density functions (PDFs) of relevant system variables such as output PDFs of the concerned systems. In this section, survey on FDD for stochastic systems will be covered.

Model-Based Fault Detection for Stochastic Systems

To realize fast FDD, model-based approaches have been developed (Chen and Patton 2012; Isermann and Munchhof 2010) for systems of which their dynamic models can be established using first principles such as mass and energy balance laws. For the purpose of FDD, the availability of a healthy system dynamic model (i.e., fault-free model) is generally assumed, say, for example, the state space models subjected to random inputs or random parameter changes. Since the system is stochastic in nature, the dynamic model is either represented as a set of continuous stochastic differential equations (such as *Ito* differential equations) or discrete-time equations, and the model can be either in the state space or the input-output format.

State Space Model-Based Fault Detection and Diagnosis

When the system is in a state space form and is linear time-invariant, then the well-known Kalman filter techniques can be readily used to perform fault detection. In this case, one needs to construct a standard Kalman filter based upon the healthy state space model of the system and then monitors the changes of the “*residual signal*” of the filtering. Such a residual is taken as a testing signal for fault detection. Since the Kalman filter is constructed using healthy system parameters, if there is no fault occurring in the system, then the residual signal would quickly converge to a zero-mean Gaussian noise with minimum variance. This intuitively indicates that if the system has a fault, then the mean value of the residual of the Kalman filter will not be close to zero (Wang and Shang 2015; Villez et al. 2011). Therefore, to perform fault detection, the Kalman filter needs to operate online in parallel to the system. The monitoring of changes in the residual signal is required in line with the progression of time. For this purpose, a small threshold value is assigned

to monitor the residual. If the magnitude of the residual signal exceeds the threshold, then it can be concluded that there is a fault in the system. With reference to Fig. 1, this means that the following fault detection mechanism should apply

$$\begin{aligned} & \text{If } \|\varepsilon(t)\| > \delta \\ & \rightarrow \text{there is a fault in the system} \quad (2) \end{aligned}$$

where $\delta > 0$ is a pre-specified threshold. Once the system is claimed of faulty, fault diagnosis can be performed in order to find out where the fault has occurred and what its size is.

However, fault diagnosis is of much more complicated problem as its first stage is to estimate changes in model parameters. Therefore, in some case, the extended state space model is used by representing certain form of dynamics for unknown system parameters. In this way, the extended state vector is formed that consists of the original state vector and the model parameters to be estimated. As a result, the well-known Kalman filter can still be used for the extended state space model to simultaneously estimate the original state vector and the system parameters. Even so, there is also an issue to diagnose original faults in the system as the model parameters can be of a set of complicated nonlinear functions of the actual physical parameters whose unexpected changes reflect the actual faults in the system. Indeed, sometimes one physical parameter changes can lead to variations of several model parameters (Isermann and Munchhof 2010). Online inverse mapping between estimated model parameters and physical parameters is required in order to further estimate unexpected changes of physical parameters.

The Kalman filtering-based method constitutes a main structure in FDD for state space model represented stochastic systems. In case when system is nonlinear, the extended Kalman filtering or other types of filtering such as minimum entropy filtering (Guo et al. 2009) or particle filtering algorithms (Kadirkamanathan et al. 2002) can be used to formulate fault detection signals and fault diagnosis.

Input-Output Model-Based Fault Detection and Diagnosis

Rather than using the state space models, one can also use input-output models to detect and diagnose the fault in the system. For example, for linear time-invariant systems represented by autoregression and moving average with exogenous (ARMAX) models, one can directly use recursive system identification algorithms to perform simultaneous fault detection and diagnosis (Isermann and Munchhof 2010). Since the parameter identification directly estimates the changes of the model parameters in the system, it can give FDD results simultaneously. Of course, when the system is nonlinear, then the rather rich tools on nonlinear system estimation (i.e., maximum likelihood estimation, Bayesian estimation, neural networks, etc.) can be readily used to estimate system parameters from input and output measurements alone (Isermann and Munchhof 2010; Abreu and Gemund 2010). Again, fault diagnosis requires the use of the mapping between model and physical parameters, and the final fault diagnosis result is to obtain the estimation on the unexpected changes of physical parameters from the estimated model parameters.

Fault Detection and Diagnosis for Unknown Systems

When the healthy model of the system is not available, one needs to use system estimation techniques to establish a healthy model to start with from the measurements of the system input and output when there is no fault in the system. This requires the use of the first principle methods and data-driven techniques or a combination of both (i.e., gray box model or semi-physical modeling) to obtain healthy models for the system without faults. In many cases, neural networks and machine learning are used for such purposes. Once healthy models are obtained, FDD can be performed in the same way as we described before in sections “[State Space Model-Based Fault Detection and Diagnosis](#)” and “[Input-Output Model-Based Fault Detection and Diagnosis](#)”. In some cases, the learning of system healthy model and FDD needs to take place either at the same time or alternatively in the progression of

time in order to reduce the false alarm caused by either normal operational deterioration of system components or nominal external disturbances.

Data-Driven Fault Detection and Diagnosis for Stochastic Systems

Change Detection for Random Signals

Given a record of a signal embedded with random noises from a system, abrupt change detection is to find out whether there are some unexpected changes in such a signal in terms of its statistics such as mean value and variance (Basseville and Nikiforov 1993). This has been performed using hypothesis tests in statistics, where a testing signal is generated from the original signal. The hypothesis is either on *there is a change* or on *there is no change*. A threshold value is normally assigned to such an exercise. If the absolute value of testing signal is larger than the threshold, then it can be concluded that there is a change (or a fault) taking place in the system. This group of approaches come from a direct use of statistics theory and has shown to be effective in detecting faults reflected by a signal alone, rather than system models and dynamics. The algorithm is simple yet is slow in coping with fast fault detection for stochastic systems.

The Use of Principal Component Analysis (PCA) for Fault Detection and Diagnosis

Another group of FDD for stochastic systems is based upon a method in multivariable statistics theory called principal component analysis (PCA). The primary objective of PCA is to reduce the dimension of high-dimensional data column mostly seen in complex industrial processes (i.e., chemical plant, paper-making, steel-making, mineral processing plant, etc.). This requires effective analysis of “*big data*” so that operational characteristics of the process can be obtained and monitored. PCA can address challenges in terms of (1) analyzing directly a high-dimensional data (normally several hundreds to thousand channels of data) embedded with various random noises and (2) performing the required fault detection (or sometimes called “condition monitoring”) for the

process. Using PCA, such a high-dimensional data can be reduced into a small number of principal components – leading to the reduction of dimensionality in the sense that this small number of principal components can reflect most of the information and variabilities of the original fault-free high-dimensional data set. In this context, FDD can be carried out by monitoring “*residual vector*” changes when a new data set is collected using Hotelling’s and Q statistics (Bakshi 1998; Hong et al. 2011). This approach is powerful in fault detection and can be difficult sometimes in performing relevant fault diagnosis. Therefore, in some cases, a combination of PCA-based fault detection with model-based approach (such as filtering- or observer-based fault diagnosis) has been used.

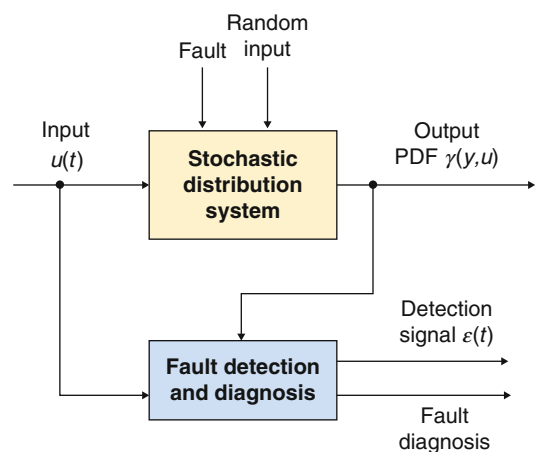
Total Probabilistic Fault Detection and Diagnosis

Random processes are fully characterized by their probability density functions (PDFs) which evolves dynamically both in time and space for dynamic stochastic systems. With the fast development of sensing technology, nowadays, PDFs of some process variables can be measured online for some control processes seen in paper-making, combustion, transportation, and chemical plant. In this context, the research question is how one can use measured PDFs of the system variables (such as the output PDFs) to detect and diagnose the faults in the system. Different from traditional FDD where only time-domain signals are used, here, one needs to use input and output PDFs, which are functions of both time and space, to perform the required FDD. The following figure shows the structure of the FDD for this kind of systems, where $u(t)$ is the input, $\gamma(y, u)$ is the output PDFs, and y is the space variable that define the time-varying shape of output PDFs (Fig. 2).

This constitutes recent advances on FDD for stochastic distribution control systems. In this case, the output of such systems is represented by its time-varying PDFs, and the input is still a function of time.

This is largely a data-driven approach, where the system model is established when no fault

occurs using measured inputs and outputs PDFs (Wang 2000; Guo et al. 2009). Then by representing the output PDFs as a B-spline approximation with fixed set of basis functions, the approximation weights associated with basis functions can be represented as a standard state space form in the time domain. In this way, the parameter matrices of such a dynamical system can be estimated online using a series of measured output PDFs together with control inputs (Wang 2000). As a result, fault detection can be carried out using observer-based fault detection approaches well studied in the control systems research community. The only difference from the standard observer-based fault detection is that the residual for fault detection is formed using the functional distance concept with weighted integration of output PDFs. Once the fault is detected, it has been shown that fault diagnosis can be carried out using adaptive diagnostic observer in the same way as FDD for deterministic systems (Wang and Daley 1996; Chen and Patton 2012). In addition, FDD for stochastic distribution systems provide a theoretical foundation for designing image-based FDD as images captured from real systems (such as combustion flames images and traffic flow images) can be characterized by time-varying PDFs.



Stochastic Fault Detection, Fig. 2 Fault detection and diagnosis for stochastic distribution systems using output PDFs

Conclusions and Future Perspectives

In this entry, existing techniques on stochastic fault detection and diagnosis have been briefly described by generally grouping them as model-based and data-driven approaches. These approaches reflect the state of the art in this area and have been successfully applied to many control systems such as those seen in mechatronics systems, process industries, aerospace systems, etc. With the ever-increased complexity of control systems, stochastic FDD will continue to be an important topic of research for enhanced safe operation of systems. As a result, we have at least the following future perspectives:

(1) **Reliable fault detection and diagnosis:**

Since the system exhibits stochastic nature (where the signals used for FDD are random processes), highly reliable fault detection needs to be investigated to see how the fault can be detected and distinguished from normal variations of random variables. This will help to design a proper threshold that minimizes false alarm yet can still detect faults of small magnitudes in time. Moreover, fault diagnosis gives the location and size of the faults, which are calculated from measured samples of random signals such as system inputs and outputs. This indicates that the fault diagnosis that generates fault estimation signal is in fact a random process itself. Therefore, it is imperative that reliable fault diagnosis is produced with minimum randomness and high level of confidence. This also applies to fault prediction, where novel algorithms should be developed that can produce highly accurate prediction of the faults so that proactive fault-tolerant controls can be obtained.

(2) **Issues with large-scale systems:** For large-scale systems, FDD need to be carried out in a distributed way. This has been a subject of study in recent years and will still be one of the focused areas for stochastic FDD (Keliris et al. 2015; Boem et al.

2019). The key is to leverage the computing and communication capabilities for complex systems to construct distributed FDD. Once local faults are detected and diagnosed, it will be communicated to relevant system levels so that distributed and collaborative fault-tolerant control can be established.

- (3) **Resilience and cyber-security perspectives:** Resilience means a kind of “recovering” capability when the system is subjected to severe faults. It is therefore a special case of fault-tolerant controls. In parallel, by taking cyberattacks as faults to the system (e.g., unexpected changes in the set points for the control systems), then it also belongs to the scope of FDD for stochastic systems. Therefore, research into resilience and cyber-security issues need to use tools developed or to be developed for stochastic FDD. Novel methods are still needed to handle the faults in terms of enhanced resilience and cyber-security for issues on model representation and fast FDD.

Cross-References

- [Estimation, Survey on](#)
- [Fault-Tolerant Control](#)
- [Kalman Filters](#)
- [Randomized Methods for Control of Uncertain Systems](#)
- [Stochastic Adaptive Control](#)

Bibliography

- Abreu R, Gemund AJC (2010) Diagnosing multiple intermittent failures using maximum likelihood estimation. *Artif Intell* 174(18): 1481–1497
- Basseville M, Nikiforov I (1993) Detection of abrupt changes: theory and application. Prentice Hall information and system sciences series. Prentice Hall, London
- Bakshi R (1998) Multiscale PCA with application to multivariate statistical process monitoring. *AICHE J* 46(7):1596–1610
- Boem F, Rivero S, Ferrari-Trecate G, Parisini T (2019) Plug-and-play fault detection and isolation for

- large-scale nonlinear systems with stochastic uncertainties. *IEEE Trans Autom Control* 64(1):4–19
- Chen J, Patton RJ (2012) Robust model-based fault diagnosis for dynamic systems. Kluwer Academic Publishers, Boston/Dordrecht/London
- Guo L, Li P, Wang H, Chai T (2009) Entropy optimization filtering for fault isolation of nonlinear non-Gaussian stochastic systems. *IEEE Trans Autom Control* 54(6):804–810
- Hong J, Zhang J, Morris J (2011) Fault localization in batch processes through progressive principal component analysis modeling. *Ind Eng Chem Res* 50(13):8153–8162
- Isermann R, Munchhof M (2010) Parameter estimation for nonlinear systems. Identification of dynamic systems. Springer, Berlin/Heidelberg
- Kadirkamanathan V, Li P, Jaward M, Fabri S (2002) Particle filtering-based fault detection in non-linear stochastic systems. *Int J Syst Sci* 33(4):259–265
- Keliris C, Polycarpou M, Parisini T (2015) Distributed fault diagnosis for process and sensor faults in a class of interconnected input-output nonlinear discrete-time systems. *Int J Control* 88(8):1472–1489
- Villez K, Srinivasan B, Rengaswamy R, Narasimhan S, Venkatasubramanian V (2011) Kalman-based strategies for fault detection and identification (FDI): extensions and critical evaluation for a buffer tank system. *Comput Chem Eng* 35(6):806–816
- Wang H (2000) Bounded dynamic stochastic distributions: modelling and control. Springer, London
- Wang H, Daley S (1996) Actuator fault diagnosis: an adaptive observer based approach. *IEEE Trans Autom Control* 41(7):1073–1077
- Wang Z, Shang H (2015) Kalman filter based fault detection for two-dimensional systems. *J Process Control* 28(1):83–94

Stochastic Games and Learning

Krzysztof Szajowski
Faculty of Pure and Applied Mathematics,
Wrocław University of Science and Technology,
Wrocław, Poland

Abstract

A stochastic game was introduced by Lloyd Shapley in the early 1950s. It is a dynamic game with *probabilistic transitions* played by one or more players. The game is played in a sequence of stages. At the beginning of

each stage, the game is in a certain *state*. The players select actions, and each player receives a *payoff* that depends on the current state and the chosen actions. The game then moves to a new random state whose distribution depends on the previous state and the actions chosen by the players. The procedure is repeated at the new state, and the play continues for a finite or infinite number of stages. The total payoff to a player is often taken to be the discounted sum of the stage payoffs or the limit inferior of the averages of the stage payoffs.

A learning problem arises when the agent does not know the reward function or the state transition probabilities. If an agent directly learns about its optimal policy without knowing either the reward function or the state transition function, such an approach is called *model-free reinforcement learning*. *Q-learning* is an example of such a model.

Q-learning has been extended to a noncooperative multi-agent context, using the framework of general-sum stochastic games. A learning agent maintains *Q*-functions over joint actions and performs updates based on assuming Nash equilibrium behavior over the current *Q*-values. The challenge is convergence of the learning protocol.

Keywords

Asynchronous dynamic programming ·
Dynamic programming · Equilibrium ·
Markov decision process · *Q-learning* ·
Reinforcement learning · Repeated game

Introduction

A Stochastic Game

Definition 1 (Stochastic games) A stochastic game is a dynamic game with probabilistic transitions played by one or more players. The game is played in a sequence of stages. At the beginning of each stage, the game is in a certain state. The players select *actions*, and each player receives a *payoff* that depends on the current state and the chosen actions. The game then moves to

a new random state whose distribution depends on the previous state and the actions chosen by the players. The process is repeated at the new state, and the play continues for a finite or infinite number of stages.

The *total payoff* to a player can be defined in various ways. It depends on the payoffs at each stage and strategies chosen by players. The aim of the players is to control their total payoffs in the game by appropriate actions.

The notion of a stochastic game was introduced by Lloyd Shapley (1953) in the early 1950s. Stochastic games generalize both Markov decision processes (see also MDP) and repeated games. A repeated game is equivalent to a stochastic game with a single state. The stochastic game is played in discrete time with past history as common knowledge for all the players. An *individual strategy* for a player is a map which associates with each given history a probability distribution on the set of actions available to the players. The players' actions at stage n determine the players' payoffs at this stage and the state $s \in \mathcal{S}$ at stage $n + 1$.

Learning

Learning is acquiring new, or modifying and reinforcing existing, knowledge, behaviors, skills, values, or preferences and may involve synthesizing different types of information. The ability to learn is possessed by humans, animals, and some machines which will be later called *agents*. In the context of this entry, learning refers to a particular class of stochastic game theoretical models.

Definition 2 (Learning in stochastic games) A learning problem arises when an agent does not know the reward function or the state transition probabilities. If the agent directly learns about its optimal policy without knowing either the reward function or the state transition function, such an approach is called *model-free reinforcement learning*. Q -learning is an example of such a model.

Learning models constitute a branch of larger literature. Players follow a form of behavioral rule, such as imitation, regret minimization,

or reinforcement. Learning models are most appropriate in settings where players have a good understanding of their strategic environment and where the stakes are high enough to make forecasting and optimization worthwhile. The known approaches are formulated as *minimax- Q* (Littman 1994), *Nash- Q* (Hu and Wellman 1998), tinkering with learning rates ("Win or Learn Fast"-WoLF, Bowling and Veloso 2001), and multiple timescale Q -learning (Leslie and Collins 2005).

Model of Stochastic Game

Let us assume that the environment is modeled by the probability space $(\Omega, \mathcal{F}, \mathbf{P})$. An N -person stochastic game is described by the objects $(\mathfrak{N}, \mathcal{S}, X_k, A_k, r_k, q)$ with the interpretation that:

1. $\mathfrak{N}$ is a set of players, with $|\mathfrak{N}| = N \in \mathbb{N}$.
2. $\mathcal{S}$ is the *set of states* of the game, and it is finite.
3. $\vec{X} = X_1 \times X_2 \times \dots \times X_N$ is the *state of actions*, where X_k is a nonempty, finite space of actions for player k .
4. A_k 's are correspondences from $\mathcal{S}$ into nonempty subsets of X_k . For each $s \in \mathcal{S}$, $A_k(s)$ represents the *set of actions* available to player k in state s . For $s \in \mathcal{S}$, denote $\vec{A}(s) = A_1(s) \times A_2(s) \times \dots \times A_N(s)$.
5. $r_k : \mathcal{S} \times \vec{X} \rightarrow \mathbb{R}$ is a payoff function for player k .
6. q is a transition probability from $\mathcal{S} \times \vec{X}$ to $\mathcal{S}$, called the *law of motion* among states. If s is a state at a certain stage of the game and the players select $\vec{x} \in \vec{A}(s)$, then $q(\cdot | s, \vec{x})$ is the probability distribution of the next state of the game.

The stochastic game generates two processes:

1. $\{\sigma_n\}_{n=1}^T$ with values in $\mathcal{S}$
2. $\{\alpha_n\}_{n=1}^T$ with values in $\vec{X}$

Strategies

Let $\mathfrak{H} = \mathfrak{S}_1 \times \overrightarrow{X}_1 \times \mathfrak{S}_2 \times \cdots$ be the space of all infinite histories of the game and $\mathfrak{H}_n = \mathfrak{S}_1 \times \overrightarrow{X}_1 \times \mathfrak{S}_2 \times \overrightarrow{X}_2 \times \cdots \times \mathfrak{S}_n$ the histories up to stage n .

Definition 3 A player's *strategy* $\pi = \{\alpha_n\}_{n=1}^T$ consists of random maps $\alpha_n : \Omega \times \mathfrak{H}_n \rightarrow X$. In other words, the strategy associates with each given history a probability distribution dependent on the set of actions available to the player. If α_n is dependent on the history only, it is called *deterministic*.

The mathematical description of the strategies can be made as follows:

1. For player $i \in \mathbb{N}$, a deterministic strategy specifies a choice of actions for the player at every stage of every possible history.
2. A mixed strategy is a probability distribution over deterministic strategies.
3. Restricted classes of strategies:
 - (a) A behavioral strategy – a mixed strategy in which the mixing takes place at each history independently.
 - (b) A Markov strategy – a behavioral strategy such that for each time t , the distribution over actions depends only on the current state, but the distribution may be different at time t than at time $t' \neq t$.
 - (c) A stationary strategy – a Markov strategy in which the distribution over actions depends only on the current state (not on the time t).

The Total Payoff Types

For any profile of strategies $\pi = (\pi_1, \dots, \pi_N)$ of the players and every initial state $s_1 = s \in \mathfrak{S}$, a probability measure P_s^π and a stochastic process $\{\sigma_n, \alpha_n\}$ are defined on $\mathfrak{H}$ in a canonical way, where the random variables σ_n and α_n describe the state and the actions chosen by the players, respectively, on the n th stage of the game. Let us define E_s^π the expectation operator with respect to the probability measure P_s^π . For each profile of strategies $\pi = (\pi_1, \dots, \pi_N)$ and every initial state $s \in \mathfrak{S}$, the following are considered:

1. The *expected T -stage payoff* to player k , for any finite horizon T , defined as

$$\Phi_k^T(\pi)(s) = E_s^\pi \left(\sum_{n=1}^T r_k(\sigma_n, \alpha_n) \right)$$

2. The β -discounted expected payoff to player k , where $\beta \in (0, 1)$ is called the *discount factor*, defined as

$$\Phi_k^\beta(\pi)(s) = E_s^\pi \left(\sum_{n=1}^{\infty} \beta^{n-1} r_k(\sigma_n, \alpha_n) \right)$$

3. The *average payoff per unit time* for player k defined as

$$\Phi_k(\pi)(s) = \limsup_T \frac{1}{T} \Phi_k^T(\pi)(s)$$

Equilibria

Let $\pi^* = (\pi_1^*, \dots, \pi_N^*) \in \Pi$ be a fixed profile of the players' strategies. For any strategy $\pi_k \in \Pi_k$ of player k , we write (π_{-k}^*, π_k) to denote the strategy profile obtained from π^* by replacing π_k^* with π_k .

Definition 4 (A Nash equilibrium) A strategy profile $\pi^* = (\pi_1^*, \dots, \pi_N^*) \in \Pi$ is called a *Nash equilibrium* (in Π) for the average payoff stochastic game if no unilateral deviations from it are profitable, that is, for each $s \in \mathfrak{S}$

$$\Phi_k(\pi^*)(s) \geq \Phi_k(\pi_{-k}^*, \pi_k)(s)$$

for every player k and any strategy π_k .

Definition 5 (An ε -Nash equilibrium) A strategy profile $\pi^* = (\pi_1^*, \dots, \pi_N^*)$ is called an ε -(Nash) *equilibrium* of the average payoff stochastic game if for every $k \in \mathfrak{N}$, we have

$$\Phi_k(\pi^*)(s) \geq \Phi_k(\pi_{-k}^*, \pi_k)(s) - \varepsilon,$$

for the given $\varepsilon > 0$ and all π_k .

Nash equilibria and ε -Nash equilibria are analogously defined for the T -stage stochastic games,

β -discounted stochastic games, and the average payoff per unit time stochastic games.

Construction of an Equilibrium

For stochastic games with a finite state space and finite action spaces, the existence of a stationary equilibrium has been shown (cf. Herings and Peeters 2004). The stationary strategies at time t do not depend on the entire history of the game up to that time. This allows reduction of the problem of finding discounted stationary equilibria in a general n -person stochastic game to that of finding a global minimum in a nonlinear program with linear constraints. Solving this nonlinear program is equivalent to solving a certain nonlinear system for which it is known that the objective value in the global minimum is zero (cf. Filar et al. 1991). However, as is noted by Breton (1991), the convergence of an optimization algorithm to the global optimum is not guaranteed.

The solution of the finite horizon finite stochastic game can be construct by dynamic programming (see, e.g., Nowak and Szajowski 1998; Tijms 2012). For discounted games, the solution construction is based on an equivalence (the two-person case is presented here for simplicity):

1. (π_1^*, π_2^*) is an equilibrium point in the discounted stochastic game with equilibrium payoffs $(\Phi_1^\beta(\vec{\pi}^*), \Phi_2^\beta(\vec{\pi}^*))$.
2. For each $s \in \mathfrak{S}$, the pair $(\pi_1^*(s), \pi_2^*(s))$ constitutes an equilibrium point in the static bimatrix game $(B_1(s), B_2(s))$ with equilibrium payoffs $(\Phi_1^\beta(s, \vec{\pi}^*), \Phi_2^\beta(s, \vec{\pi}^*))$, where for players $k = 1, 2$, and pure actions $(a_1, a_2) \in A_1(s) \times A_2(s)$, an admissible action space at state s , the elements of $B_k(s)$ related to (a_1, a_2)

$$b_k(s, a_1, a_2) := (1 - \beta)r_k(s, a_1, a_2) + \beta E_s^{(a_1, a_2)} \Phi_k^\beta(\vec{\pi}^*) \quad (1)$$

An algorithm for recursive computation of stationary equilibria in stochastic games can be derived from (1). It starts with bimatrix

games with $\beta = 0$, and then a careful equilibrium selection process guarantees its convergence under mild assumptions on the model (see, e.g., Herings and Peeters 2004).

A Brief History of the Research on Stochastic Games

The notion of a stochastic game was introduced by Shapley (1953) in the early 1950s. It is a dynamic game with *probabilistic transitions* played by one or more players. The game is played in a sequence of stages. At the beginning of each stage, the game is in a certain *state*. The players select actions, and each player receives a *payoff* that depends on the current state and the chosen actions. The game then moves to a new random state whose distribution depends on the previous state and the actions chosen by the players. The process is repeated at the new state, and the play continues for a finite or an infinite number of stages. The total payoff to a player is often taken to be the discounted sum of the stage payoffs or the limit inferior of the averages of the stage payoffs. Stochastic games generalize both Markov decision processes (cf. Bellman 1957; Howard 1960) and repeated games (cf. Stearns 1967; Aumann 1985).

The theory of nonzero-sum stochastic games with the average payoffs per unit time for the players started with the papers by Rogers (1969) and Sobel (1971). They considered finite state spaces only and assumed that the transition probability matrices induced by any stationary strategies of the players are irreducible. Until now, only special classes of nonzero-sum average payoff stochastic games have been shown to possess Nash equilibria (or ε -equilibria).

The study of an important subclass of stochastic games began the work of Dynkin (1969). In its wording, the players had the right to stop the observed process by getting a reward dependent on the state of stopped process. Significant results and to Kiefer (1971), Neveu (1975), and Ohtsubo (1987).

A review of various cases and results for generalization to infinite state spaces can be found in the survey papers by Nowak and Szajowski

(1998), Vieille (2002) and Jaśkiewicz and Nowak (2018).

Learning in Stochastic Game

The problem of an agent learning to act in an unknown world is both challenging and interesting. Reinforcement learning has been successful at finding optimal control policies for a single agent operating in a stationary environment, specifically a Markov decision process. Learning to act in multi-agent systems offers additional challenges (see the following surveys: Shoham and Leyton-Brown 2009, Chap. 7; Weiß and Sen 1996; Buşoniu et al. 2010). We provide here, an overview of a general idea of learning for single and multi-agent systems:

1. Goals of single-agent reinforcement learning are to determine the optimal value and a control policy which maximizes the payoff. The model of such a system can be built based on the framework of Markov decision processes with discounted payoff. Suppose the policy is stationary and defined by a function $h : \mathcal{S} \rightarrow X$. Such a policy defines what action should be taken in each state: $\alpha_n(\cdot) := h(\cdot)$. There are various ways to learn the optimal policy. The most straightforward way is based on the Q -values: $Q^h(s, a) = \sum_{j=0}^{\infty} \beta_j^{jr}$. The greedy action is $a = \arg \max_{a' \in A(s)} Q^h(s, a')$ (see the article on Q -learning in Reinforcement learning).
2. Multi-agent reinforcement learning can be employed to solve a single task, or an agent may be required to perform a task in an environment with other agents, either human, robot, or software ones. In either case, from an agent's perspective, the world is not stationary. In particular, the behavior of the other agents may change as they also learn to better perform their tasks. This type of a multi-agent nonstationary world creates a difficult problem for learning to act in these environments. Such a nonstationary scenario can be viewed as a game with multiple players. In game theory, in

the study of such problems, there is generally an underlying assumption that the players have similar adaptation and learning abilities. Therefore, the actions of each agent affect the task achievement of the other agents. It allows to build the value of the game and an equilibrium strategy profile in following steps.

Stochastic games can be seen as an extension of the single-agent Markov decision process framework to include multiple agents whose actions all impact the resulting rewards and the next state. They can also be viewed as an extension of the framework of matrix games. Such a view emphasizes the difficulty of finding the optimal behavior in stochastic games since the optimal behavior of any one agent depends on the behavior of other agents. A comprehensive study of the multi-agent learning techniques for stochastic games does not yet exist. For the interested reader, there are monographs by Fudenberg and Levine (1998) and Shoham and Leyton-Brown (2009) and the special issue of the journal *Artificial Intelligence* (Vohra and Wellman 2007), which could be consulted.

Despite its interesting properties, Q -learning is a very slow method that requires a long period of training for learning an acceptable policy. In practice, to reduce the problem, there are parallel computing implementation models of Q -learning.

Summary and Future Directions

Details concerning solution concepts for stochastic games can be found in Filar and Vrieze (1997). The refinements of the Nash equilibrium concept have been known in the economic dynamic games (see Myerson 1978). The Nash equilibrium concept may be extended gradually when the rules of the game are interpreted in a broader sense, so as to allow preplay or even intraplay communication. A well-known extension of the Nash equilibrium is Aumann's correlated equilibrium (see Aumann 1987), which depends only on

the normal form of the game. Two other solution concepts for multistage games have been proposed by Forges (1986): the extensive form correlated equilibrium, where the players can observe private exogenous signals at every stage, and the communication equilibrium, where the players are furthermore allowed to transmit inputs to an appropriate device at every stage. An application of the notion of correlated equilibria for stochastic games can be found in Nowak and Szajowski (1998).

In economics, in the context of economic growth problems, Ramsey (1928) has introduced an *overtaking optimality* and independently Rubinstein (1979) for repeated games. The criterion has been investigated for some stochastic games by Carlson and Haurie (1995) and Nowak (2008) and others. The existence of overtaking optimal strategies is a subtle issue, and there are counterexamples showing that one has to be careful with making statements on overtaking optimality.

Regarding a stochastic game and learning, let us mention that the first idea can be found in the papers by Brown (1951) and Robinson (1951). Some convergence results for a fictitious play have been given by Shoham and Leyton-Brown (2009) in Theorem 7.2.5. An important example showing non-convergence was given by Shapley (1964). In multi-person stochastic games and learning, convergence to equilibria is a basic stability requirement (see, e.g., Greenwald and Hall 2003; Hu and Wellman 2003). This means that the agents' strategies should eventually converge to a coordinated equilibrium. Nash equilibrium is most frequently used, but their usefulness is suspected. For instance, in Shoham and Leyton-Brown (2009), there is an argument that the link between stage-wise convergence to Nash equilibria and the performance in stochastic games is unclear.

The research on repeated game with incomplete information formulated by Harsanyi (1967) (cf. Harsanyi 1968a, b; Kohlberg 1975; Gensbittel and Renault 2015) stimulate the investigation of stopping game with asymmetric and incomplete information (e.g., Grün 2013). A general treatment of stochastic filtering, crucial tool for

these question in stopping game, can be found in Liptser and Shiryaev (1977).

Cross-References

- Auctions
- Dynamic Noncooperative Games
- Evolutionary Games
- Game Theory: A General Introduction and a Historical Overview
- Iterative Learning Control
- Learning in Games
- Learning Theory: The Probably Approximately Correct Framework
- Option Games: The Interface Between Optimal Stopping and Game Theory
- Stochastic Adaptive Control

Bibliography

- Aumann RJ (1985) Repeated games. In: Feiwel GR (ed) Issues in contemporary microeconomics and welfare. Palgrave Macmillan, London. https://doi.org/10.1007/978-1-349-06876-0_5
- Aumann RJ (1987) Correlated equilibrium as an expression of Bayesian rationality. *Econometrica* 55:1–18. <https://doi.org/10.2307/1911154>
- Bellman R (1957) A Markovian decision process. *J Math Mech* 6:679–684
- Bowling M, Veloso M (2001) Rational and convergent learning in stochastic games. In: Proceedings of the 17th international joint conference on artificial intelligence (IJCAI), Seattle, pp 1021–1026
- Breton M (1991) Algorithms for stochastic games. In: Raghavan TES, Ferguson TS, Parthasarathy T, Vrieze OJ (eds) Stochastic games and related topics: in honor of Professor L. S. Shapley, vol 7. Springer Netherlands, Dordrecht, pp 45–57. https://doi.org/10.1007/978-94-011-3760-7_5
- Brown GW (1951) Iterative solution of games by fictitious play. In: Koopmans TC (ed) Activity analysis of production and allocation. Wiley, New York, Chap XXIV, pp 374–376
- Buşoniu L, Babuška R, Schutter BD (2010) Multi-agent reinforcement learning: an overview. In: Srinivasan D, Jain LC (eds) Innovations in multi-agent systems and application–1. Springer, Berlin, pp 183–221
- Carlson D, Haurie A (1995) A turnpike theory for infinite horizon open-loop differential games with decoupled controls. In: Olsder GJ (ed) New trends in dynamic games and applications. Annals of the international society of dynamic games, vol 3. Birkhäuser, Boston, pp 353–376

- Dynkin EB (1969) The game variant of a problem on optimal stopping. *SovMath Dokl* 10: 270–274
- Filar J, Vrieze K (1997) *Competitive Markov decision processes*. Springer, New York
- Filar JA, Schultz TA, Thuijsman F, Vrieze OJ (1991) Nonlinear programming and stationary equilibria in stochastic games. *Math Program* 50(2, Ser A):227–237. <https://doi.org/10.1007/BF01594936>
- Forges F (1986) An approach to communication equilibria. *Econometrica* 54:1375–1385. <https://doi.org/10.2307/1914304>
- Fudenberg D, Levine DK (1998) *The theory of learning in games*, vol 2. MIT, Cambridge
- Gensbittel F, Renault J (2015) The value of Markov chain games with incomplete information on both sides. *Math Oper Res* 40(4):820–841. <https://doi.org/10.1287/moor.2014.0697>
- Greenwald A, Hall K (2003) Correlated-Q learning. In: *Proceedings 20th international conference on machine learning (ISML-03)*, Washington, DC, 21–24 Aug 2003, pp 242–249
- Grün Ch (2013) On Dynkin games with incomplete information. *SIAM J Control Optim* 51(5):4039–4065. <https://doi.org/10.1137/120891800>
- Harsanyi JC (1967) Games with incomplete information played by “Bayesian” players. I. The basic model. *Manag Sci* 14:159–182. <https://doi.org/10.1287/mnsc.14.3.159>
- Harsanyi JC (1968a) Games with incomplete information played by “Bayesian” players. II. Bayesian equilibrium points. *Manag Sci* 14:320–334. <https://doi.org/10.1287/mnsc.14.5.320>
- Harsanyi JC (1968b) Games with incomplete information played by “Bayesian” players. III. The basic probability distribution of the game. *Manag Sci* 14:486–502. <https://doi.org/10.1287/mnsc.14.7.486>
- Herings PJ-J, Peeters RJAP (2004) Stationary equilibria in stochastic games: structure, selection, and computation. *J Econ Theory* 118(1):32–60. <https://doi.org/10.1016/j.jet.2003.10.001>
- Howard RA (1960) *Dynamic programming and markov processes*. The MIT Press, Cambridge
- Hu J, Wellman MP (1998) Multiagent reinforcement learning: theoretical framework and an algorithm. In: *Proceedings of the 15th international conference on machine learning*, New Brunswick, pp 242–250.
- Hu J, Wellman MP (2003) Nash Q-learning for general-sum stochastic games. *J Mach Learn Res* 4:1039–1069
- Jaśkiewicz A, Nowak AS (2018) Non-zero-sum stochastic games. In: Basar T, Zaccour G (eds) *Handbook of dynamic game theory*. Springer, Cham, pp 1–64. ISBN:978-3-319-27335-8, https://doi.org/10.1007/978-3-319-27335-8_33-3
- Kiefer YI (1971) Optimal stopped games. *Theory Probab Appl* 16:185–189
- Kohlberg E (1975) Optimal strategies in repeated games with incomplete information. *Int J Game Theory* 4(1–2):7–24
- Leslie DS, Collins EJ (2005) Individual Q -learning in normal form games. *SIAM J Control Optim* 44(2):495–514. <https://doi.org/10.1137/S0363012903437976>
- Liptser RS, Shiryaev AN (1977) *Statistics of random processes*, Volume 2. Applications of mathematics, vol 6. Springer, New York
- Littman ML (1994) Markov games as a framework for multi-agent reinforcement learning. In: *Proceedings of the 13th international conference on machine learning*, New Brunswick, pp 157–163
- Mertens J-F, Zamir S (1971/1972) The value of two-person zero-sum repeated games with lack of information on both sides. *Int J Game Theory* 1:39–64
- Myerson RB (1978) Refinements of the nash equilibrium concept. *Int J Game Theory* 7(2):73–80. <https://doi.org/10.1007/BF01753236>
- Neveu J (1975) *Discrete-parameter martingales*. North-Holland, Amsterdam
- Nowak AS (2008) Equilibrium in a dynamic game of capital accumulation with the overtaking criterion. *Econ Lett* 99(2):233–237. <https://doi.org/10.1016/j.econlet.2007.05.033>
- Nowak AS, Szajowski K (1998) Nonzerosum stochastic games. In: Bardi M, Raghavan TES, Parthasarathy T (eds) *Stochastic and differential games: theory and numerical methods*. Annals of the international society of dynamic games, vol 4. Birkhäuser, Boston, pp 297–342. https://doi.org/10.1007/978-1-4612-1592-9_7
- Ohtsubo Y (1987) A nonzero-sum extension of Dynkin’s stopping problem. *Math Oper Res* 12(2):277–296. <https://doi.org/10.1287/moor.12.2.277>
- Ramsey F (1928) A mathematical theory of savings. *Econ J* 38:543–559
- Robinson J (1951) An iterative method of solving a game. *Ann Math* 2(54):296–301. <https://doi.org/10.2307/1969530>
- Rogers PD (1969) *Nonzero-sum stochastic games*, Ph.D. thesis, University of California, Berkeley. ProQuest LLC, Ann Arbor
- Rubinstein A (1979) Equilibrium in supergames with the overtaking criterion. *J Econ Theory* 21:1–9. [https://doi.org/10.1016/0022-0531\(79\)90002-4](https://doi.org/10.1016/0022-0531(79)90002-4)
- Shapley L (1953) Stochastic games. *Proc Natl Acad Sci USA* 39:1095–1100. <https://doi.org/10.1073/pnas.39.10.1095>
- Shapley L (1964) Some topics in two-person games. *Ann Math Stud* 52:1–28
- Shoham Y, Leyton-Brown K (2009) *Multiagent systems: algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, Cambridge. <https://doi.org/10.1017/CBO9780511811654>
- Sobel MJ (1971) Noncooperative stochastic games. *Ann Math Stat* 42:1930–1935. <https://doi.org/10.1214/aoms/1177693059>
- Stearns RE (1967) A formal information concept for games with incomplete information. Report of the US arms control and disarmament agency/ST-116, 405, Washington, DC, ch 4
- Tijms H (2012) Stochastic games and dynamic programming. *Asia Pac Math Newsl* 2(3):6–10

- Vieille N (2002) Stochastic games: recent results. In: Handbook of game theory. Elsevier Science, Amsterdam, pp 1833–1850. ISBN 0-444-88098-4
- Vohra R, Wellman M (eds) (2007) Foundations of multi-agent learning. Artif Intell 171:363–452
- Weiß G, Sen S (eds) (1996) Adaption and learning in multi-agent Systems. In: Proceedings of the IJCAI'95 workshop, Montréal, 21 Aug 1995, vol 1042. Springer, Berlin. <https://doi.org/10.1007/3-540-60923-7>
- Zamir S (1971/1972) On the relation between finitely and infinitely repeated games with incomplete information. Int J Game Theory 1:179–198

Stochastic Linear-Quadratic Control

Shanjian Tang
Fudan University, Shanghai, China

Abstract

In this short article, we briefly review some major historical studies and recent progress on continuous-time stochastic linear-quadratic (SLQ) control and related mean-variance (MV) hedging.

Keywords

Bellman's quasilinearization · BMO-martingale · Mean-variance hedging · Monotone convergence · Quadratic backward stochastic differential equations · Riccati equation

Introduction

A stochastic linear-quadratic (SLQ) control problem is the optimal control of a linear stochastic dynamic equation subject to an expected quadratic cost functional of the system state and control. As shown in Athans (1971), it is a typical case of optimal stochastic control both in theory and application. Due to the linearity of the

system dynamics and the quadratic feature of the cost functions, the optimal control law is usually synthesized into a feedback (also called closed) form of the optimal state, and the corresponding proportional coefficients are specified by the associated Riccati equation. In what follows, we restrict our exposition within the continuous-time SLQ problem, and further, mainly for the finite-horizon case.

The initial study on the continuous-time SLQ problem seems to be due to Florentin (1961). However, his linear stochastic control system is assumed to be Gaussian. That is, the system noise is additive and has neither multiplication with the state nor with the control. Such a case is usually termed as the linear-quadratic Gaussian (LQG) problem, and in the case of complete observation, the optimal feedback law remains to be invariant when the white noise vanishes. The continuous-time partially observable case was first discussed by Potter (1964) and a more general formulation was later given by Wonham (1968a). It is proved that the optimal control can be obtained by the following two separate steps: (1) generate the conditional mean estimate of the current state using a Kalman filter and (2) optimally feed back as if the conditional mean state estimate was the true state of the system. This result is referred to as the certainty equivalence principle or the strict separation theorem. Different assumptions were discussed by Tse (1971) for the separation of control and state estimation.

Wonham (1967, 1968b, 1970) investigated the SLQ problem in a fairly general systematic framework. In the first two papers, his stochastic system is able to admit a state-dependent noise. Finally, Wonham (1970) considered the following very general (admitting both state- and control-dependent noise) linear stochastic differential system driven by a d -dimensional Brownian motion $W = (W^1, W^2, \dots, W^d)$:

$$X_t = x + \int_0^t (A_s X_s + B_s u_s) dt + \int_0^t \sum_{i=1}^d (C_s^i X_s + D_s^i u_s) dW_s^i, \quad t \in [0, T];$$

and the following cost functional:

$$J(u) = E\langle MX_T, X_T \rangle + E \int_0^T [\langle Q_t X_t, X_t \rangle + \langle N_t u_t, u_t \rangle] dt.$$

Here, $T > 0$, $X_t \in \mathbb{R}^n$ is the state at time t , and $u_t \in \mathbb{R}^m$ is the control at time t . Assume

$$\begin{cases} -\dot{K}_t = A_t^* K_t + K_t A_t + C_t^{i*} K_t C_t^i - \Gamma_t(K_t)(N_t + D_t^{i*} K_t D_t^i) \Gamma_t(K_t), & t \in [0, T]; \\ K_T = M. \end{cases} \quad (1)$$

Here, the asterisk stands for transpose, the repeated superscripts imply summation from 1 to d , and the function Γ is defined by

$$\Gamma_t(K) := -(N_t + D_t^{i*} K D_t^i)^{-1} (K B_t + C_t^{i*} K D_t^i)^*$$

for time $t \in [0, T]$ and any $K \in \mathcal{S}_+^n := \{\text{all nonnegative } n \times n \text{ matrices}\}$. This Riccati equation is a nonlinear ordinary differential equation (ODE). Since the nonlinear term $\Gamma_t(K)(N_t + D_t^{i*} K D_t^i) \Gamma_t(K)$ in the right-hand side is not uniformly Lipschitz in K in general, the standard existence and uniqueness theorem of ODEs does not directly tell whether this Riccati equation has a unique continuous solution in $\mathcal{S}_+^n$. To solve this issue, Wonham (1970) used Bellman's principle of quasilinearization and constructed the following sequence of successive linear approximating matrix-valued ODEs.

Define for $(t, K, \tilde{\Gamma}) \in [0, T] \times \mathbb{R}^{n \times n} \times \mathbb{R}^{m \times n}$,

$$\begin{aligned} F_t(K, \tilde{\Gamma}) &:= [A_t + B_t \tilde{\Gamma}]^* K + K [A_t + B_t \tilde{\Gamma}] \\ &\quad + [C_t^i + D_t^{i*} \tilde{\Gamma}]^* K [C_t^i + D_t^{i*} \tilde{\Gamma}] \\ &\quad + Q_t + \tilde{\Gamma}^* N_t \tilde{\Gamma}. \end{aligned} \quad (2)$$

For $K \in \mathcal{S}_+^n$, the matrix $F_t(K, \tilde{\Gamma}) - F_t(K, \Gamma_t(K))$ is nonnegative, that is,

$$F_t(K, \tilde{\Gamma}) \geq F_t(K, \Gamma_t(K)), \quad \forall \tilde{\Gamma} \in \mathbb{R}^{m \times n}. \quad (3)$$

Riccati equation (1) can then be written into the following form:

that all the coefficients $A, B; C^i, D^i, i = 1, 2, \dots, d; Q, N$ are piecewisely continuous matrix-valued (of suitable dimensions) functions of time, and M, Q_t are nonnegative matrices and N_t is uniformly positive. Wonham (1970) gave the following Riccati equation:

$$\begin{cases} -\dot{K}_t = F_t(K_t, \Gamma_t(K_t)), & t \in [0, T]; \\ K_T = M. \end{cases} \quad (4)$$

The iterating linear approximations are therefore structured as follows: Set $K^0 \equiv M$ and for $l = 1, 2, \dots$,

$$\begin{cases} -\dot{K}_t^l = F_t(K_t^l, \Gamma_t(K_t^{l-1})), & t \in [0, T]; \\ K_T^l = M. \end{cases} \quad (5)$$

Using the above minimal property (3) of $F_t(K, \cdot)$ at $\Gamma_t(K)$, Wonham showed that the unique nonnegative solution K^l of ODE (5) is monotonically decreasing in the sequential number $l = 1, 2, \dots$. Using the method of monotone convergence, the sequence of solutions $\{K^l\}$ is shown to converge to some $K \in \mathcal{S}_+^n$, which turns out to solve Riccati equation (1).

The Case of Random Coefficients and Backward Stochastic Riccati Equation

Bismut (1976, 1978) are the first studies on the SLQ problem with random coefficients. Let $\{\mathcal{F}_t, t \in [0, T]\}$ be the completed natural filtration of W . When the coefficients $A, B; C^i, D^i, i = 1, 2, \dots, d; Q, N$ and M may be random, with $A, B; C^i, D^i, i = 1, 2, \dots, d; Q, N$ being $\mathcal{F}_t$ -adapted and essentially bounded and M being $\mathcal{F}_T$ -measurable and essentially bounded, Bismut (1976, 1978) used the stochastic maximum principle for

optimal control and derived the following Riccati equation:

$$\left\{ \begin{array}{l} -dK_t = [A_t^* K_t + K_t A_t + C_t^{i*} K_t C_t^i + C_t^{i*} L_t^i + L_t^i C_t^i \\ \quad - \Psi_t(K_t, L_t)(N_t + D_t^{i*} K_t D_t^i) \Psi_t(K_t, L_t)] dt - L^i dW_t^i, \quad t \in [0, T]; \\ K_T = M \end{array} \right. \quad (6)$$

where the function Ψ_t for $t \in [0, T]$ is defined as follows:

$$\begin{aligned} \Psi_t(K, L) &:= -(N_t + D_t^{i*} K D_t^i)^{-1} (K B_t + C_t^{i*} K D_t^i + L^i D_t^i)^*, \forall K \in \mathcal{S}_+^n, \forall L \\ &:= (L^1, \dots, L^d) \in (R^{n \times n})^d. \end{aligned}$$

Peng (1992b) used his stochastic Hamilton-Jacobi-Bellman equation to the SLQ problem and also derived the above equation. They both established the existence and uniqueness of an adapted solution of backward stochastic Riccati equation (6) when the function $\Psi_t(K, L)$ does not contain L . However, Bismut used the fixed-point method, and Peng (1992b) used Bellman's principle of quasilinearization and the method of monotone convergence. Neither methodology works for the general case of quadratic growth in the second unknown variable L in the drift of the stochastic equation. Bismut (1976, 1978) and Peng (1999) stated the general case as an open problem. By considering the stochastic equation for the inverse of K_t , Kohlmann and Tang (2003a) solved some particular cases where the function $\Psi_t(K, L)$ can depend on L . Tang (2003) finally solved the general case, using the method of stochastic flows.

In the general case, the optimal feedback coefficient $\Psi_t(K_t, L_t)$ at time t depends on L_t in a linear manner, which is in general not essentially bounded with respect to (t, ω) . Kohlmann and Tang (2003b) observed that the stochastic integral process $\int_0^t L_t^i dW_t^i$ is a BMO-martingale.

Indefinite SLQ Problem

Chen (1985) contains a theory of singular (the control weighting matrix vanishing in the quadratic cost functional) LQG control, which is a particular type of indefinite SLQ problems. In the deterministic linear-quadratic (LQ) control theory, the well posedness (i.e., the value function is finite on $[0, T] \times R^n$) of the problem suggests that the control weighting matrix N in the quadratic cost functional be positive definite. In the stochastic case, when N_t is slightly negative, the SLQ may still be well posed if the control could also increase the intensity of the system noise. Peng (1992a) used an indefinite but well-posed SLQ problem to illustrate his new second-order stochastic maximum principle. Chen et al. (1998) gave a deeper study on this feature of the SLQ problem. Yong and Zhou (1999) gave a systematic account of the progress around in the indefinite SLQ problem.

Mean-Variance Hedging

In the theory of finance, Duffie and Richardson (1991) introduced the SLQ control model to hedge a contingent claim in an incomplete

market. Schweizer (1992) developed a first framework for MV hedging, and then it was extended to a very general setting in Gouriéroux et al. (1998). Before 2000, the martingale method was used to solve the MV hedging problem. Kohlmann and Zhou (2000) began to use the standard SLQ theory to derive the optimal hedging strategy for a general contingent claim in a financial market of deterministic coefficients, and such a SLQ methodology was subsequently extended to very general settings for financial markets by Kohlmann and Tang (2002, 2003b), Bobrovnytska and Schweizer (2004), and Jeanblanc et al. (2012). See more detailed surveys on the literature by Pham (2000), Schweizer (2010), and Jeanblanc et al. (2012).

Summary and Future Directions

In comparison to the continuous-time deterministic LQ theory, the continuous-time SLQ theory has the following two striking features: An indefinite SLQ problem may be well posed, and the optimal feedback coefficient may be unbounded due to its linear dependence on the martingale part L of the stochastic solution of the Riccati equation. Due to the second feature, the convergence of the sequence of successive approximations constructed via Bellman's quasilinearization still remains to be solved in the general case. This problem partially motivates Delbaen and Tang (2010) to study the regularity of unbounded stochastic differential equations and also may help to explain the necessity of rich studies on mean-variance hedging and closedness of stochastic integrals with respect to semimartingales (as in Delbaen et al. 1994, 1997) in various general settings.

Cross-References

- [Backward Stochastic Differential Equations and Related Control Problems](#)
- [Stochastic Maximum Principle](#)

Recommended Reading

The theory of SLQ control in various contexts is available in textbooks, monographs, or papers. Anderson and Moore (1971, 1989), Bensoussan (1992), and Chen (1985) include good accounts of the LQG control theory. Wonham (1970) includes a full introduction to the SLQ problem with deterministic piecewise continuous-time coefficients. Bismut (1978) gives a systematic and readable French introduction to SLQ problem with random coefficients. Yong and Zhou (1999) include an extensive discussion on the well-posed indefinite SLQ problem. Tang (2003) gives a complete solution of a general backward stochastic Riccati equation.

Bibliography

- Anderson BDO, Moore JB (1971) Linear optimal control. Prentice-Hall, Englewood Cliffs
- Anderson BDO, Moore JB (1989) Optimal control: linear quadratic methods. Prentice-Hall, Englewood Cliffs
- Athans M (1971) The role and use of the stochastic linear-quadratic-Gaussian problem in control system design. *IEEE Trans Autom Control* AC-16(6):529–552
- Bensoussan A (1992) Stochastic control of partially observable systems. Cambridge University Press, Cambridge
- Bismut JM (1976) Linear quadratic optimal stochastic control with random coefficients. *SIAM J Control Optim* 14:419–444
- Bismut JM (1978) Contrôle des systèmes linéaires quadratiques: applications de l'intégrale stochastique. In: Delachérie C, Meyer PA, Weil M (eds) *Séminaire de probabilités XII. Lecture Notes in Math* 649. Springer, Berlin, pp 180–264
- Bobrovnytska O, Schweizer M (2004) Mean-variance hedging and stochastic control: beyond the Brownian setting. *IEEE Trans Autom Control* 49:396–408
- Chen H (1985) Recursive estimation and control for stochastic systems. Wiley, New York, pp 302–335
- Chen S, Li X, Zhou X (1998) Stochastic linear quadratic regulators with indefinite control weight costs. *SIAM J Control Optim* 36:1685–1702
- Delbaen F, Tang S (2010) Harmonic analysis of stochastic equations and backward stochastic differential equations. *Probab Theory Relat Fields* 146:291–336
- Delbaen F et al (1994) Weighted norm inequalities and closedness of a space of stochastic integrals. *C R Acad Sci Paris Sér I Math* 319:1079–1081
- Delbaen F et al (1997) Weighted norm inequalities and hedging in incomplete markets. *Financ Stoch* 1: 181–227

- Duffie D, Richardson HR (1991) Mean-variance hedging in continuous time. *Ann Appl Probab* 1:1–15
- Florentin JJ (1961) Optimal control of continuous-time, Markov, stochastic systems. *J Electron Control* 10:473–488
- Gouriéroux C, Laurent JP, Pham H (1998) Mean-variance hedging and numéraire. *Math Financ* 8:179–200
- Jeanblanc M et al (2012) Mean-variance hedging via stochastic control and BSDEs for general semimartingales. *Ann Appl Probab* 22:2388–2428
- Kohlmann M, Tang S (2002) Global adapted solution of one-dimensional backward stochastic Riccati equations, with application to the mean-variance hedging. *Stoch Process Appl* 97: 255–288
- Kohlmann M, Tang S (2003a) Multidimensional backward stochastic Riccati equations and applications. *SIAM J Control Optim* 41:1696–1721
- Kohlmann M, Tang S (2003b) Minimization of risk and linear quadratic optimal control theory. *SIAM J Control Optim* 42:1118–1142
- Kohlmann M, Zhou XY (2000) Relationship between backward stochastic differential equations and stochastic controls: a linear-quadratic approach. *SIAM J Control Optim* 38:1392–1407
- Peng S (1992a) New developments in stochastic maximum principle and related backward stochastic differential equations. In: *Proceedings of the 31st conference on decision and control, Tucson, Dec 1992*. IEEE, pp 2043–2047
- Peng S (1992b) Stochastic Hamilton-Jacobi-Bellman equations. *SIAM J Control Optim* 30: 284–304
- Peng S (1999) Open problems on backward stochastic differential equations. In: Chen S, Li X, Yong J, Zhou XY (eds) *Control of distributed parameter and stochastic systems*, IFIP, Hangzhou. Kluwer, pp 267–272
- Pham H (2000) On quadratic hedging in continuous time. *Math Methods Oper Res* 51:315–339
- Potter JE (1964) A guidance-navigation separation theorem. *Experimental Astronomy Laboratory, Massachusetts Institute of Technology, Cambridge, Rep. RE-11*, 1964
- Schweizer M (1992) Mean-variance hedging for general claims. *Ann Appl Probab* 2:171–179
- Schweizer M (2010) Mean-variance hedging. In: Cont R (ed) *Encyclopedia of quantitative finance*. Wiley, New York, pp 1177–1181
- Tang S (2003) General linear quadratic optimal stochastic control problems with random coefficients: linear stochastic Hamilton systems and backward stochastic Riccati equations. *SIAM J Control Optim* 42:53–75
- Tse E (1971) On the optimal control of stochastic linear systems. *IEEE Trans Autom Control* AC-16(6): 776–785
- Wonham WM (1967) Optimal stationary control of a linear system with state-dependent noise. *SIAM J Control* 5:486–500
- Wonham WM (1968a) On the separation theorem of stochastic control. *SIAM J Control* 6:312–326
- Wonham WM (1968b) On a matrix Riccati equation of stochastic control. *SIAM J Control* 6:681–697. Erratum (1969); *SIAM J Control* 7:365
- Wonham WM (1970) Random differential equations in control theory. In: Bharucha-Reid AT (ed) *Probabilistic methods in applied mathematics*. Academic, New York, pp 131–212
- Yong JM, Zhou XY (1999) *Stochastic controls: Hamiltonian systems and HJB equations*. Springer, New York

Stochastic Maximum Principle

Ying Hu

CNRS, IRMAR – UMR6625, Université
Rennes, Rennes, France

Abstract

The stochastic maximum principle (SMP) gives some necessary conditions for optimality for a stochastic optimal control problem. We give a summary of well-known results concerning stochastic maximum principle in finite-dimensional state space as well as some recent developments in infinite-dimensional state space.

Keywords

Stochastic optimal control · Necessary condition · Stochastic maximum principle

Introduction

The problem of finding necessary conditions for optimality for a stochastic optimal control problem with finite-dimensional state equation had been well studied since the pioneering work of Bismut (1976, 1978). In particular, Bismut introduced linear backward stochastic differential equations (BSDEs) which have become an active domain of research since the seminal paper of Pardoux and Peng in 1990 concerning (nonlinear) BSDEs in Pardoux and Peng (1990).

The first results on SMP concerned only the stochastic systems where the control domain is convex or the diffusion coefficient does not contain control variable. In this case, only the first-

order expansion is needed. This kind of SMP was developed by Bismut (1976, 1978), Kushner (1972), Haussmann (1986), ... It is important to note that Bismut in Bismut (1976, 1978) introduced linear BSDE to represent the first-order adjoint process.

Peng made a breakthrough by establishing the SMP for the general stochastic optimal control problem where the control domain needs not to be convex and the diffusion coefficient can contain the control variable. He solved this general case by introducing the second-order expansion and second-order BSDE. We refer to the book (Yong and Zhou 1999) for the account of the theory of SMP in finite-dimensional spaces and describe Peng's SMP in the next section. We note that recently, SMP has found wide applications in probabilistic theory of mean field games; see the recent book (Carmona and Delarue 2018).

Despite the fact that the problem has been solved in complete generality more than 20 years ago, the infinite-dimensional case still has important open issues both on the side of the generality of the abstract model and on the side of its applicability to systems modeled by stochastic partial differential equations (SPDEs). Last section is devoted to the recent development of SMP in infinite-dimensional space.

Statement of SMP

Formulation of Problem

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space, on which an m -dimensional Brownian motion W is given. Let $\{\mathcal{F}_t\}_{t \geq 0}$ be the natural completed filtration of W .

We consider the following stochastic controlled system:

$$\begin{aligned} dx(t) &= b(x(t), u(t))dt + \sigma(x(t), u(t))dW(t), \\ x(0) &= x_0, \end{aligned} \quad (1)$$

with the cost functional

$$J(u(\cdot)) = \mathbb{E} \left\{ \int_0^T f(x(t), u(t))dt + h(x(T)) \right\}. \quad (2)$$

In the above, b, σ, f, h are given functions with appropriate dimensions. (U, d) is a separable metric space.

We define

$$\begin{aligned} \mathcal{U} &= \{u : [0, T] \times \Omega \rightarrow U \mid u \text{ is} \\ &\quad \{\mathcal{F}_t\}_{t \geq 0} - \text{adapted}\}. \end{aligned} \quad (3)$$

The optimal problem is: Minimize $J(u(\cdot))$ over $\mathcal{U}$.

Any $\bar{u} \in \mathcal{U}$ satisfying

$$J(\bar{u}) = \inf_{u \in \mathcal{U}} J(u) \quad (4)$$

is called an optimal control. The corresponding $\bar{x}$ and $(\bar{x}, \bar{u})$ are called an optimal state process/trajectory and optimal pair, respectively.

In this section, we assume the following standard hypothesis.

Hypothesis 2.1

1. The functions $b : \mathbb{R}^n \times U \mapsto \mathbb{R}^n$, $\sigma = (\sigma^1, \dots, \sigma^m) : \mathbb{R}^n \times U \mapsto \mathbb{R}^{n \times m}$, $f : \mathbb{R}^n \times U \mapsto \mathbb{R}$ and $h : \mathbb{R}^n \mapsto \mathbb{R}$ are measurable functions.
2. For $\varphi = b, \sigma^j, j = 1, \dots, m, f$, the functions $x \mapsto \varphi(x, u)$ and $x \mapsto h(x)$ are C^2 , denoted φ_x and φ_{xx} (h_x and h_{xx} , respectively), which are also continuous functions of (x, u) .
3. There exists a constant $K > 0$ such that

$$|\varphi_x| + |\varphi_{xx}| + |h_x| + |h_{xx}| \leq K,$$

and

$$|\varphi| + |h| \leq K(1 + |x| + |u|).$$

Adjoint Equations

Let us first introduce the following backward stochastic differential equations (BSDEs).

$$dp(t) = -\{b_x(\bar{x}(t), \bar{u}(t))^T p(t)$$

$$\begin{aligned}
& + \sum_{j=1}^m \sigma_x^j(\bar{x}(t), \bar{u}(t))^T q_j(t) \\
& - f_x(\bar{x}(t), \bar{u}(t))\} dt + q(t) dW(t), \\
p(T) & = -h_x(\bar{x}(T)). \tag{5}
\end{aligned}$$

The solution (p, q) to the above BSDE (first-order BSDE) is called the first-order adjoint process.

$$\begin{aligned}
dP(t) & = -\{b_x(\bar{x}(t), \bar{u}(t))^T P(t) + P(t)b_x(\bar{x}(t), \bar{u}(t)) + \sum_{j=1}^m \sigma_x^j(\bar{x}(t), \bar{u}(t))^T P(t) \sigma_x^j(\bar{x}(t), \bar{u}(t)) \\
& + \sum_{j=1}^m \{\sigma_x^j(\bar{x}(t), \bar{u}(t))^T Q_j(t) + Q_j(t) \sigma_x^j(\bar{x}(t), \bar{u}(t)) \\
& + H_{xx}(\bar{x}(t), \bar{u}(t), p(t), q(t))\} dt + \sum_{j=1}^m Q_j(t) dW^j(t), \tag{6} \\
P(T) & = -h_{xx}(\bar{x}(T)),
\end{aligned}$$

where the Hamiltonian H is defined by

$$\begin{aligned}
H(x, u, p, q) & = \langle p, b(x, u) \rangle \\
& + \text{tr}[q^T \sigma(x, u)] - f(x, u). \tag{7}
\end{aligned}$$

The solution (P, Q) to the above BSDE (second-order BSDE) is called second-order adjoint process.

Stochastic Maximum Principle

Let us now state the stochastic maximum principle.

Theorem 1 *Let $(\bar{x}, \bar{u})$ be an optimal pair of problem. Then there exist a unique couple (p, q) satisfying (5) and a unique couple (P, Q) satisfying (6), and the following maximum condition holds:*

$$\begin{aligned}
& H(\bar{x}(t), \bar{u}(t), p(t), q(t)) - H(\bar{x}(t), u, p(t), q(t)) \\
& - \frac{1}{2} \text{tr}(\{\sigma(\bar{x}(t), \bar{u}(t)) \\
& - \sigma(\bar{x}(t), u)\}^T P(t) \{\sigma(\bar{x}(t), \bar{u}(t)) \\
& - \sigma(\bar{x}(t), u)\}) \geq 0. \tag{8}
\end{aligned}$$

SMP in Infinite-Dimensional Space

The problem of finding necessary conditions for optimality for a stochastic optimal control problem with infinite-dimensional state equation, along the lines of the Pontryagin maximum principle, was already addressed in the early 1980s in the pioneering paper (Bensoussan 1983).

Whereas the Pontryagin maximum principle for infinite-dimensional stochastic control problems is a well-known result as far as the control domain is convex (or the diffusion does not depend on the control), see Bensoussan (1983) and Hu and Peng (1990); for the general case (i.e., when the control domain needs not be convex and the diffusion coefficient can contain a control variable), existing results are limited to abstract evolution equations under assumptions that are not satisfied by the large majority of concrete SPDEs.

The technical obstruction is related to the fact that (as it was pointed out in Peng 1990) if the control domain is not convex, the optimal control has to be perturbed by the so-called spike variation. Then if the control enters the diffusion, the irregularity in time of the Brownian trajectories imposes to take into account a second variation process. Thus the stochastic maximum principle has to involve an adjoint process for the second variation. In the finite-dimensional case, such a process can be characterized as the solution

of a matrix-valued backward stochastic differential equation (BSDE), while in the infinite-dimensional case, the process naturally lives in a non-Hilbertian space of operators, and its characterization is much more difficult. Moreover the applicability of the abstract results to concrete controlled SPDEs is another delicate step due to the specific difficulties that they involve such as the lack of regularity of Nemytskii-type coefficients in L^p spaces.

Concerning results on the infinite-dimensional stochastic Pontryagin maximum principle, as we already mentioned, in Bensoussan (1983) and Hu and Peng (1990), the case of diffusion independent on the control is treated (with the difference that in Hu and Peng (1990) a complete characterization of the adjoint to the first variation as the unique mild solution to a suitable BSDE is achieved).

The paper Tang and Li (1994) is the first one in which the general case is addressed with, in addition, a general class of noises possibly with jumps. The adjoint process of the second variation $(P_t)_{t \in [0, T]}$ is characterized as the solution of a BSDE in the (Hilbertian) space of Hilbert-Schmidt operators. This forces to assume a very strong regularity on the abstract state equation and control functional that prevent application of the results in Tang and Li (1994) to SPDEs.

Then in the papers by Fuhrman et al. (2012, 2013), the state equation is formulated, only in a semiabstract way in order, on one side, to cope with all the difficulties carried by the concrete non-linearities and on the other to take advantage of the regularizing properties of the leading elliptic operator.

In Lü and Zhang (2014), P_t was characterized as “transposition solution” of a backward stochastic evolution equation in $\mathcal{L}(L^2(\mathcal{O}))$. Coefficients are required to be twice Fréchet-differentiable as operators in $L^2(\mathcal{O})$. And in Du and Meng (2013), the process P_t is characterized in a similar way as it is in Fuhrman et al. (2012, 2013). Roughly speaking it is characterized as a suitable stochastic bilinear form. As it is the case in Lü and Zhang (2014) and in Du and Meng (2013) as well, the regularity assumptions on the coefficients are too restrictive to apply directly

the results in Lü and Zhang (2014) and Du and Meng (2013) to controlled SPDEs.

Finally, a stochastic maximum principle for the optimal control of a stochastic partial differential equation driven by white noise in the case when the set of control actions is convex is proved in Fuhrman et al. (2018). Particular attention is paid to well-posedness of the adjoint backward stochastic differential equation and the regularity properties of its solution with values in infinite-dimensional spaces.

Cross-References

- [Backward Stochastic Differential Equations and Related Control Problems](#)
- [Optimal Control and Pontryagin's Maximum Principle](#)
- [Stochastic Dynamic Programming](#)
- [Stochastic Linear-Quadratic Control](#)

Bibliography

- Bensoussan A (1983) Stochastic maximum principle for distributed parameter systems. *J Franklin Inst* 315 (5–6):387–406
- Bismut JM (1976) Linear quadratic optimal stochastic control with random coefficients. *SIAM J Control Optim* 14(3):419–444
- Bismut JM (1978) An introductory approach to duality in optimal stochastic control. *SIAM Rev* 20(1):62–78
- Carmona R, Delarue F (2018) Probabilistic theory of mean field games with applications. I. Mean field FBSDEs, control, and games. Probability theory and stochastic modelling, vol 83. Springer, Cham
- Du K, Meng Q (2013) A maximum principle for optimal control of stochastic evolution equations. *SIAM J Control Optim* 51(6):4343–4362
- Fuhrman M, Hu Y, Tessitore G (2012) Stochastic maximum principle for optimal control of SPDEs. *C R Math Acad Sci Paris* 350(13–14):683–688
- Fuhrman M, Hu Y, Tessitore G (2013) Stochastic maximum principle for optimal control of SPDEs. *Appl Math Optim* 68(2):181–217
- Fuhrman M, Hu Y, Tessitore G (2018) Stochastic maximum principle for optimal control of partial differential equations driven by white noise. *Stoch Partial Differ Equ Anal Comput* 6(2):255–285
- Haussmann UG (1986) A stochastic maximum principle for optimal control of diffusions. Pitman research notes in mathematics series, vol 151. Longman Scientific & Technical, Harlow/Wiley, New York

- Hu Y, Peng S (1990) Maximum principle for semilinear stochastic evolution control systems. *Stoch Stoch Rep* 33(3–4):159–180
- Kushner HJ (1972) Necessary conditions for continuous parameter stochastic optimization problems. *SIAM J Control* 10:550–565
- Lü Q, Zhang X (2014) General Pontryagin-type stochastic maximum principle and backward stochastic evolution equations in infinite dimensions. *Springer briefs in mathematics*. Springer, Cham
- Pardoux E, Peng S (1990) Adapted solution of a backward stochastic differential equation. *Syst Control Lett* 14(1):55–61
- Peng S (1990) A general stochastic maximum principle for optimal control problems. *SIAM J Control Optim* 28(4):966–979
- Tang S, Li X (1994) Maximum principle for optimal control of distributed parameter stochastic systems with random jumps. In: Markus L, Elworthy KD, Everitt WN, Bruce Lee E (eds) *Differential equations, dynamical systems, and control science*. Lecture notes in pure and applied mathematics, vol 152. Dekker, New York, pp 867–890
- Yong J, Zhou XY (1999) *Stochastic controls. Hamiltonian systems and HJB equations*. Applications of mathematics, vol 43. Springer, New York

Stochastic Model Predictive Control

Basil Kouvaritakis and Mark Cannon
Department of Engineering Science, University of Oxford, Oxford, UK

Abstract

Stochastic model predictive control is a form of model predictive control that takes account of the stochastic nature of uncertain parameters and disturbances affecting the system model. This information may be used in the definition of performance indices, constraints, or both. We discuss cost functions, constraints, closed-loop properties, and implementation issues.

Keywords

Stochastic lyapunov function · Mean-square stability · Recursive feasibility

Introduction

Stochastic model predictive control (MPC) refers to a family of optimal control strategies for stochastic systems. Future performance is quantified using a cost function evaluated along predicted trajectories for system states and control inputs, taking into account the distribution of stochastic model uncertainty. This leads to a stochastic optimal control problem, which is solved numerically to determine an optimal open-loop control sequence or alternatively a sequence of feedback control laws. Only the first element of this optimal sequence is applied to the controlled system, and the optimal control problem is solved again at the next sampling instant on the basis of updated information on the system state. The numerical nature of the approach makes it applicable to systems with nonlinear dynamics and constraints on states and inputs, while the repeated optimization of predicted trajectories using updated measurements introduces feedback to compensate for the effects of uncertainty in the model.

Robust MPC tackles problems with hard state and input constraints that must be satisfied for all realizations of model uncertainty. However, robust MPC strategies can be overly conservative in many applications. Stochastic MPC provides a less conservative control strategy by expressing the predicted performance index and constraints in terms of expectations or probabilistic bounds. Constraints limit performance, and an advantage of MPC is that it can allow systems to operate close to constraint boundaries. Likewise, stochastic MPC provides advantages by expressing performance and constraints in terms statistical quantities rather than worst-case realizations of model uncertainty.

Applications of stochastic MPC have been reported in diverse fields, including finance and portfolio management, risk management, sustainable development policy assessment, chemical and process industries, electricity generation and distribution, building climate control, and telecommunications network traffic control. This entry aims to summarize the

theoretical framework underlying stochastic MPC algorithms.

Stochastic MPC

Consider a system with discrete time model

$$x^+ = f(x, u, w) \quad (1)$$

$$z = g(x, u, v) \quad (2)$$

where $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$ are the system state and control input and x^+ is the successor state (i.e., if x_i is the state at time i , then $x^+ = x_{i+1}$ is the state at time $i + 1$). Inputs $w \in \mathbb{R}^{n_w}$ and $v \in \mathbb{R}^{n_v}$ are exogenous disturbances with unknown current and future values but known probability distributions, and $z \in \mathbb{R}^{n_z}$ is a vector of output variables that are subject to constraints.

The optimal control problem that is solved online at each time step in stochastic MPC is defined in terms of a performance index $J_N(x, \hat{\mathbf{u}}, \hat{\mathbf{w}})$ evaluated over a future horizon of N time steps. Typically in stochastic MPC, $J_N(x, \hat{\mathbf{u}}, \hat{\mathbf{w}})$ is a quadratic function of the following form (in which $\|x\|_Q^2 = x^T Q x$)

$$J_N(x, \hat{\mathbf{u}}, \hat{\mathbf{w}}) = \sum_{i=0}^{N-1} (\|\hat{x}_i\|_Q^2 + \|\hat{u}_i\|_R^2) + V_f(\hat{x}_N) \quad (3)$$

for positive definite matrices Q and R , and a terminal cost $V_f(x)$ defined as discussed in section “[Stability and Convergence](#).” Here $\hat{\mathbf{u}} := \{\hat{u}_0, \dots, \hat{u}_{N-1}\}$ is a postulated sequence of control inputs, and $\hat{\mathbf{x}}(x, \hat{\mathbf{u}}, \hat{\mathbf{w}}) := \{\hat{x}_0, \dots, \hat{x}_N\}$ is the corresponding sequence of states such that $\hat{x}_i$ is the solution of (1) at time i with initial state $\hat{x}_0 = x$, for a given sequence of disturbance inputs $\hat{\mathbf{w}} := \{\hat{w}_0, \dots, \hat{w}_{N-1}\}$. Since $\hat{\mathbf{w}}$ is a random sequence, $J_N(x, \hat{\mathbf{u}}, \hat{\mathbf{w}})$ is a random variable, and the optimal control problem is therefore formulated as the minimization of a cost $V_N(x, \hat{\mathbf{u}})$ derived from $J_N(x, \hat{\mathbf{u}}, \hat{\mathbf{w}})$ under

specific assumptions on $\hat{\mathbf{w}}$. Common definitions of $V_N(x, \hat{\mathbf{u}})$ are as follows:

- (a) Expected value cost:

$$V_N(x, \hat{\mathbf{u}}) := \mathbb{E}_x(J(x, \hat{\mathbf{u}}, \hat{\mathbf{w}}))$$

where $\mathbb{E}_x(\cdot)$ denotes the conditional expectation of a random variable $(\cdot)$ given the model state x .

- (b) Worst-case cost, assuming $\hat{w}_i \in \mathcal{W}$ for all i with probability 1, for some compact set $\mathcal{W} \subset \mathbb{R}^{n_w}$:

$$V_N(x, \hat{\mathbf{u}}) := \max_{\hat{\mathbf{w}} \in \mathcal{W}^N} J(x, \hat{\mathbf{u}}, \hat{\mathbf{w}}).$$

- (c) Nominal cost, assuming $\hat{w}_i$ is equal to some nominal value, e.g., if $\hat{w}_i = 0$ for all i , then

$$V_N(x, \hat{\mathbf{u}}) := J(x, \hat{\mathbf{u}}, \mathbf{0}),$$

where $\mathbf{0} = \{0, \dots, 0\}$.

The minimization of $V_N(x, \hat{\mathbf{u}})$ is performed subject to constraints on the sequence of outputs $\hat{z}_i := g(\hat{x}_i, \hat{u}_i, \hat{v}_i)$, $i \geq 0$. These constraints may be formulated in various ways, summarized as follows, where for simplicity we assume $n_z = 1$:

- (A) Expected value constraints: for all i ,

$$\mathbb{E}_x(\hat{z}_i) \leq 1.$$

- (B) Probabilistic constraints pointwise in time:

$$\Pr_x(\hat{z}_i \leq 1) \geq p,$$

for all i and for a given probability p .

- (C) Probabilistic constraints over a future horizon:

$$\Pr_x(\hat{z}_i \leq 1, i = 0, 1, \dots, N) \geq p$$

for a given probability p .

In (B) and (C), $\Pr_x(\mathcal{A})$ represents the conditional probability of an event $\mathcal{A}$ that depends on the sequence $\hat{\mathbf{x}}(x, \hat{\mathbf{u}}, \hat{\mathbf{w}})$, given that the initial model state is $\hat{x}_0 = x$; for example, the probability $\Pr_x(\hat{z}_i \leq 1)$ depends on the distribution of $\{\hat{w}_0, \dots, \hat{w}_{i-1}, \hat{v}_i\}$.

The important special case of state constraints can also be handled by (A)–(C) through appropriate choice of the function $g(x, u, v)$. For example, the constraint $\Pr_x(h(x) \leq 1) \geq p$, for a given function $h : \mathbb{R}^n \rightarrow \mathbb{R}$, can be expressed in the form (B) with $z = g(x, u, v) := h(f(x, u, w))$ and $v := w$ in (2).

In common with other receding horizon control strategies, stochastic MPC is implemented via the following algorithm. At each discrete time step:

- (i) Minimize the cost index $V_N(x, \hat{\mathbf{u}})$ over $\hat{\mathbf{u}}$ subject to the constraints on $\hat{z}_i, i \geq 0$, given the current system state x .
- (ii) Apply the control input $u = \hat{u}_0^*(x)$ to the system, where $\hat{\mathbf{u}}^*(x) = \{\hat{u}_0^*(x), \dots, \hat{u}_{N-1}^*(x)\}$ is the minimizing sequence given x .

If the system dynamics (1) are unstable, then performing the optimization in step (i) directly over future control sequences can result in a small set of feasible states x . To avoid this difficulty the elements of the control sequence $\hat{\mathbf{u}}$ are usually expressed in the form $\hat{u}_i = u_T(\hat{x}_i) + s_i$, where $u_T(x)$ is a locally stabilizing feedback law and $\{s_0, \dots, s_{N-1}\}$ are optimization variables in step (i).

Constraints and Recursive Feasibility

The constraints in (B) and (C) include hard constraints ($p = 1$) as a special case, but in general the conditions (A)–(C) represent soft constraints that are not required to hold for all realizations of model uncertainty. However, these constraints can only be satisfied if the state belongs to a subset of state space, and the requirement (common in MPC) that the optimization in step (i) of the stochastic MPC algorithm should remain feasible if it is initially feasible therefore implies

additional constraints. For example, the condition $\Pr_x(\hat{z}_0 \leq 1) \geq p$ can be satisfied only if x belongs to the set for which there exists $\hat{u}_0$ such that $\Pr_x(g(x, \hat{u}_0, \hat{v}_0) \leq 1) \geq p$. Hence, soft constraints implicitly impose hard constraints on the model state.

Stochastic MPC algorithms typically handle the conditions relating to feasibility of constraint sets in one of two ways. Either the stochastic MPC optimization is allowed to become infeasible (often with penalties on constraint violations included in the cost index) or conditions ensuring robust feasibility of the stochastic MPC optimization at all future times are imposed as extra constraints in the MPC optimization.

The first of these approaches has been used in the context of constraints (C) imposed over a horizon, for which conditions ensuring future feasibility are generally harder to characterize in terms of algebraic conditions on the model state than (A) or (B). A disadvantage of this approach is that the closed-loop system may not satisfy the required soft constraints, even if these constraints are feasible when applied to system trajectories predicted at initial time.

The second approach treats conditions for feasibility as hard constraints and hence requires a guarantee of recursive feasibility, namely, that the MPC optimization must remain feasible for the closed-loop system if it is feasible initially. This can be achieved by requiring, similarly to robust MPC, that the conditions for feasibility of the optimization problem should be satisfied for all realizations of the sequence $\hat{\mathbf{w}}$. For example, for given $\hat{x}_0 = x$, there exists $\hat{\mathbf{u}}$ satisfying that the conditions of (B) if

$$\Pr_{\hat{x}_i}(g(\hat{x}_i, \hat{u}_i, \hat{v}_i) \leq 1) \geq p, i = 0, 1, \dots \quad (4a)$$

$$\hat{x}_i \in X \quad \forall \{\hat{w}_0, \dots, \hat{w}_{i-1}\} \in \mathcal{W}^i, i = 1, 2, \dots \quad (4b)$$

where X is the set

$$X = \{x : \exists u \text{ such that } \Pr_x(g(x, u, v) \leq 1) \geq p\}.$$

Furthermore, a stochastic MPC optimization that includes the constraints of (4) will remain

feasible at subsequent times (since (4b) ensures the existence of $\hat{\mathbf{u}}^+$ such that each element of $\hat{\mathbf{x}}(f(x, \hat{u}_0, \hat{w}_0), \hat{\mathbf{u}}^+, \hat{\mathbf{w}}^+)$ lies in X for all $\hat{w}_0 \in \mathcal{W}$ and all $\hat{\mathbf{w}}^+ \in \mathcal{W}^N$).

Satisfaction of (4) at each time step i on the infinite horizon $i \geq N$ can be ensured through a finite number of constraints by introducing constraints on the N -step-ahead state $\hat{x}_N$. This approach uses a fixed feedback law, $u_T(x)$, to define a postulated input sequence after the initial N -step horizon via $\hat{u}_i = u_T(\hat{x}_i)$ for all $i \geq N$. The constraints of (4) are necessarily satisfied for all $i \geq N$ if a constraint

$$\hat{x}_N \in X_T$$

is imposed, where X_T is robustly positively invariant with probability 1 under $u_T(x)$, i.e.,

$$f(x, u_T(x), w) \in X_T, \quad \forall x \in X_T, \quad \forall w \in \mathcal{W}, \quad (5)$$

and furthermore the constraint $\Pr_x(z \leq 1) \geq p$ is satisfied at each point in X_T under $u_T(x)$, i.e.,

$$\Pr_x(g(x, u_T(x), v) \leq 1) \geq p, \quad \forall x \in X_T.$$

Although the recursively feasible constraints (4) account robustly for the future realizations of the unknown parameter w in (1), the key difference between stochastic and robust MPC formulations is that the conditions in (4) depend on the probability distribution of the parameter v in (2). It also follows from the necessity of hard constraints for feasibility that the distribution of w must in general have finite support in order that feasibility can be guaranteed recursively. On the other hand, the support of v in the definition of z may be unbounded (an important exception being the case of state constraints in which $v = w$).

Stability and Convergence

This section outlines the stability properties of stochastic MPC strategies based on cost indices (a)–(c) of section “Stochastic MPC” and related variants. We use $V_N^*(x) = V_N(x, \hat{\mathbf{u}}^*(x))$ to denote the optimal value of the cost index, and

X_T denotes a subset of state space satisfying the robust invariance condition (5). We also denote the solution at time i of the system (1) with initial state $x_0 = x$ and under a given feedback control law $u = \kappa(x)$ and disturbance sequence $\mathbf{w} = \{w_0, w_1, \dots\}$ as $x_i(x, \kappa, \mathbf{w})$.

The expected value cost index in (a) results in mean-square stability of the closed-loop system provided the terminal term $V_f(x)$ in (3) satisfies

$$\begin{aligned} \mathbb{E}_x V_f(f(x, u_T(x), w)) &\leq V_f(x) - \|x\|_Q^2 \\ &\quad - \|u_T(x)\|_R^2 \end{aligned}$$

for all x in the terminal set X_T . The optimal cost is then a stochastic Lyapunov function satisfying

$$\begin{aligned} \mathbb{E}_x V_N^*(f(x, \hat{u}_0^*(x), w)) &\leq V_N^*(x) - \|x\|_Q^2 \\ &\quad - \|\hat{u}_0^*(x)\|_R^2. \end{aligned}$$

For positive definite Q , this implies the closed-loop system under the stochastic MPC law is mean-square stable, so that $x_i(x, \hat{u}_0^*, \mathbf{w}) \rightarrow 0$ as $i \rightarrow \infty$ with probability 1 for any feasible initial condition x . For the case of systems (1) subject to additive disturbances, the modified cost

$$\begin{aligned} V_N(x, \hat{\mathbf{u}}) &:= \mathbb{E}_x \left[\sum_{i=0}^{N-1} (\|\hat{x}_i\|_Q^2 + \|\hat{u}_i\|_R^2 - l_{ss}) \right. \\ &\quad \left. + V_f(\hat{x}_N) \right] \end{aligned}$$

where $l_{ss} := \lim_{i \rightarrow \infty} \mathbb{E}_x(\|x_i(x, u_T, \mathbf{w})\|_Q^2 + \|u_i\|_R^2)$ under $u_i = u_T(x_i)$ results in the asymptotic bound

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \mathbb{E}_x(\|x_i(x, \hat{u}_0^*, \mathbf{w})\|_Q^2 + \|u_i\|_R^2) \leq l_{ss}$$

along the closed-loop trajectories of (1) under the MPC law $u_i = \hat{u}_0^*(x_i)$, for any feasible initial condition x .

Considering the worst-case cost (b), if $V_f(x)$ is designed as a control Lyapunov function for (1), with

$$V_f(f(x, u_T(x), w)) \leq V_f(x) - \|x\|_Q^2 - \|u_T(x)\|_R^2$$

for all $w \in \mathcal{W}$ and all $x \in X_T$, then $V_N^*(x)$ is a Lyapunov function satisfying

$$V_N^*(f(x, \hat{u}_0^*(x), w)) \leq V_N^*(x) - \|x\|_Q^2 - \|\hat{u}_0^*(x)\|_R^2$$

for all $w \in \mathcal{W}$, implying $x = 0$ is an asymptotically stable equilibrium of (1) under the MPC law $u = \hat{u}_0^*(x)$. Clearly the system model (1) cannot be subject to unknown additive disturbances in this case. However, for the case in which the system (1) is subject to additive disturbances, a variant of this approach uses a modified cost which is equal to zero inside some set of states, leading to asymptotic stability of this set rather than an equilibrium point. Also in the context of additive disturbances, an alternative approach uses an $\mathcal{H}_\infty$ -type cost,

$$V_N(x, \hat{\mathbf{u}}) := \max_{\hat{\mathbf{w}} \in \mathcal{W}^N} \left[\sum_{i=0}^{N-1} (\|\hat{x}_i\|_Q^2 + \|\hat{u}_i\|_R^2 - \gamma^2 \|\hat{w}_i\|^2) + V_f(\hat{x}_N) \right]$$

for which the closed-loop trajectories of (1) under the associated MPC law $u_i = \hat{u}_0^*(x_i)$ satisfy

$$\sum_{i=0}^{\infty} (\|x_i(x, \hat{u}_0^*, \mathbf{w})\|_Q^2 + \|u_i\|_R^2) \leq \gamma^2 \sum_{i=0}^{\infty} \|w_i\|^2 + V_N^*(x_0)$$

provided $V_f(f(x, u_T(x), w)) \leq V_f(x) - (\|x\|_Q^2 + \|u_T(x)\|_R^2 - \gamma^2 \|w\|^2)$ for all $w \in \mathcal{W}$ and $x \in X_T$.

Algorithms employing the nominal cost (c) typically rely on the existence of a feedback law $u_T(x)$ such that the system (1) satisfies, in the absence of constraints and under $u_i = u_T(x_i)$, an input-to-state stability (ISS) condition of the form

$$\sum_{i=0}^{\infty} (\|x_i(x, u_T, \mathbf{w})\|_Q^2 + \|u_i\|_R^2) \leq \gamma^2 \sum_{i=0}^{\infty} \|w_i\|^2 + \beta \quad (6)$$

for some γ and $\beta > 0$. If $V_f(x)$ satisfies

$$V_f(f(x, u_T(x), 0)) \leq V_f(x) - (\|x\|_Q^2 + \|u_T(x)\|_R^2)$$

for all $x \in X_T$, then the closed-loop system under stochastic MPC with the nominal cost (c) satisfies an ISS condition with the same gain γ as the unconstrained case (6) but a different constant β .

Implementation Issues

In general stochastic MPC algorithms require more computation than their robust counterparts because of the need to determine the probability distributions of future states. An important exception is the case of linear dynamics and purely additive disturbances, for which the model (1)–(2) becomes

$$x^+ = Ax + Bu + w \quad (7)$$

$$z = Cx + Du + v \quad (8)$$

where A, B, C, D are known matrices. In this case the expected value constraints (A) and probabilistic constraints (B), as well as hard constraints that ensure future feasibility of the stochastic MPC optimization in each case, can be invoked non-conservatively through tightened constraints on the expectations of future states. Furthermore, the required degree of tightening can be computed off-line using numerical integration of probability distributions or using random sampling techniques, and the online computational load is similar to MPC with no model uncertainty.

The case in which the matrices A, B, C, D in the model (7)–(8) depend on unknown stochastic parameters is more difficult because the predicted states then involve products of random variables. An effective approach to this problem uses a sequence of sets (known as a tube) to recursively bound the sequence of predicted states via one step-ahead set inclusion conditions. By using polytopic bounding sets that are defined as the

intersection of a fixed number of half-spaces, the complexity of these tubes can be controlled by the designer, albeit at the expense of conservative inclusion conditions. Furthermore, an application of Farkas' lemma allows these sets to be computed online through linear conditions on optimization variables.

Random sampling techniques developed for general stochastic programming problems provide effective means of handling the soft constraints arising in stochastic MPC. These techniques use finite sets of samples or scenarios to represent the probability distributions of model states and parameters. Furthermore bounds are available on the number of samples that are needed in order to meet specified confidence levels on the satisfaction of constraints. Probabilistic and expected value constraints can be imposed using random sampling, and this approach has also been applied to the case of probabilistic constraints over a horizon (C) through a scenario-based optimization approach.

Summary and Future Directions

This entry describes how the ideas of robust MPC can be extended to the case of stochastic model uncertainty. Crucial in this development are the assumptions on uncertainty distributions that allow the assertion of recursive feasibility of the stochastic MPC optimization problem. For simplicity of presentation, we have considered the case of full-state feedback. However, stochastic MPC can also be applied to the output feedback case using a state estimator if the probability distributions of measurement and estimation noise are known.

An area of future development is optimization over sequences of feedback policies. Although an observer at initial time cannot know the future realizations of random uncertainty, information on $\hat{x}_i$ will be available to the controller i -steps ahead, and, as mentioned in section “[Stochastic MPC](#)” in the context of feasible initial condition sets, $\hat{u}_i$ must therefore depend on $\hat{x}_i$. In general the optimal control decision is of the form $\hat{u}_i = \mu_i(\hat{x}_i)$ where $\mu_i(\cdot)$ is a feedback policy.

This implies optimization over arbitrary feedback policies, which is considered to be intractable in general since the required online computation grows exponentially with the horizon N . However, approximate approaches to this problem have been suggested which optimize over restricted classes of feedback laws and further developments in this respect are expected in the future.

Stochastic MPC has a central role to play in the development of adaptive model predictive control. This is a variant of model predictive control in which online measurements are used to construct improved estimates of uncertain parameters or structural elements of system models while simultaneously controlling the system. In this context stochastic MPC has the potential to optimize the design of control inputs for the purposes of model identification while respecting state and input constraints through the inclusion of additional constraints or terms in the performance index.

Cross-References

- [Distributed Model Predictive Control](#)
- [Economic Model Predictive Control](#)
- [Nominal Model-Predictive Control](#)
- [Robust Model Predictive Control](#)
- [Tracking Model Predictive Control](#)

Recommended Reading

A historical perspective on stochastic MPC is provided by Åström and Wittenmark (1973), Charnes and Cooper (1963), and Schwarm and Nikolaou (1999). A treatment of constraints stated in terms of expected values can be found, for example, in Primbs and Sung (2009). Probabilistic constraints and the conditions for recursive feasibility can be found in Kouvaritakis et al. (2010) for the additive case, whereas the general case of multiplicative and additive uncertainty is described in Fleming and Cannon (2019), which uses random sampling techniques. Random sampling techniques were developed

for random convex programming (Calafiore and Campi 2005) and were used in a scenario-based approach to predictive control in Calafiore and Fagiano (2013). An output feedback stochastic MPC strategy incorporating state estimation is described in Cannon et al. (2012).

The use of the expectation of a quadratic cost and associated mean-square stability results are discussed in Lee and Cooley (1998). Robust stability results for MPC based on worst-case costs are given by Lee and Yu (1997) and Mayne et al. (2005). Input-to-state stability of MPC based on a nominal cost is discussed in Marruedo et al. (2002).

Descriptions of stochastic MPC based on closed-loop optimization can be found in Lee and Yu (1997) and Batina (2004). These algorithms are computationally intensive, and approximate solutions can be found by restricting the class of closed-loop predictions as discussed, for example, in van Hessem and Bosgra (2002) and Primbs and Sung (2009).

Bibliography

- Åström KJ, Wittenmark B (1973) On self-tuning regulators. *Automatica* 9(2):185–199
- Batina I (2004) Model predictive control for stochastic systems by randomized algorithms. Eindhoven: Technische Universiteit Eindhoven. <https://doi.org/10.6100/IR573294>
- Calafiore GC, Campi MC (2005) Uncertain convex programs: randomized solutions and confidence levels. *Math Program* 102(1):25–46
- Calafiore GC, Fagiano L (2013) Robust model predictive control via scenario optimization. *IEEE Trans Autom Control* 58(1):219–224
- Cannon M, Cheng Q, Kouvaritakis B, Raković SV (2012) Stochastic tube MPC with state estimation. *Automatica* 48(3):536–541
- Charnes A, Cooper WW (1963) Deterministic equivalents for optimizing and satisficing under chance constraints. *Oper Res* 11(1):19–39
- Fleming J, Cannon M (2019) Stochastic MPC for additive and multiplicative uncertainty using sample approximations. *IEEE Trans Autom Control* 64(9):3883–3888. <https://doi.org/10.1109/TAC.2018.2887054>
- Kouvaritakis B, Cannon M, Raković SV, Cheng Q (2010) Explicit use of probabilistic distributions in linear predictive control. *Automatica* 46(10):1719–1724
- Lee JH, Cooley BL (1998) Optimal feedback control strategies for state-space systems with stochastic parameters. *IEEE Trans Autom Control* 43(10):1469–1475
- Lee JH, Yu Z (1997) Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica* 33(5):763–781
- Marruedo DL, Alamo T, Camacho EF (2002) Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In: *IEEE conference on decision and control, Las Vegas*, pp 4619–4624
- Mayne DQ, Seron MM, Raković SV (2005) Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica* 41(2):219–224
- Primbs JA, Sung CH (2009) Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise. *IEEE Trans Autom Control* 54(2):221–230
- Schwarm AT, Nikolaou M (1999) Chance-constrained model predictive control. *AIChE J* 45(8):1743–1752
- van Hessem DH, Bosgra OH (2002) A conic reformulation of model predictive control including bounded and stochastic disturbances under state and input constraints. In: *IEEE conference on decision and control, Las Vegas*, pp 4643–4648

Stock Trading via Feedback Control Methods

B. Ross Barmish¹ and James A. Primbs²

¹ECE Department, Boston University, Boston, MA, USA

²Department of Finance, California State University Fullerton, Fullerton, CA, USA

Abstract

This article covers stock trading from a feedback control point of view. To this end, mechanics and practical considerations associated with the use of feedback algorithms are explained for both real-world trading and scenarios involving numerical simulation.

Keywords

Finance · Stock trading · Feedback control · Simulation

This paper is a updated version of an article which appeared in the previous edition of this encyclopedia.

Introduction

Stock trading involves the purchase and sale of *shares* of ownership in public companies by an individual or entity such as a pension fund, mutual fund, hedge fund, or endowment. These shares are typically traded in many markets through major exchanges, such as those in New York, London, and Shanghai, to name a few, with the trader's goal generally being to increase wealth. In pursuit of this goal, investors are typically unwilling to use a trading scheme which they deem to be excessively risky vis-a-vis the level of returns being sought. Although attention in this paper is restricted to the practicalities associated with the mechanics of trading and related simulation, it should be mentioned that risk considerations are paramount in a more full-blown analysis which includes a decision whether to go forward or not; e.g., variance, drawdown, and value-at-risk are frequently considered. The authors recommend the textbook by Luenberger (2013) as an entry point for this topic.

The words *feedback control* in the title of this article refer to the use of information which becomes available to the trader over time and is used to make buy and sell decisions on the stock according to some set of rules. That is, the size of the stock position being held varies with time. The mapping from this information to the investment level is called the *feedback law* and is typically described with a closed-loop configuration and classical algorithms which come from the body of research called control theory; e.g., see Astrom and Murray (2008). In this paper, to keep the scope narrowly confined to stock-trading mechanics, numerical simulation, and practical considerations associated with the use of feedback-based algorithms, we only describe the most basic of trading schemes here. We refer the reader to our work, (Barmish and Primbs 2016) and its bibliography, as one of many possible entry points to the voluminous body of theoretical literature covering many more aspects of stock trading beyond the introductory material covered here.

For simplicity, in this article, we restrict attention to trading a single stock while noting that the

concepts described herein are readily modified to address the multi-stock case, i.e., a *portfolio*. The basic idea of viewing portfolios in a control-theoretic setting goes back at least as far as Merton (1969) and Samuelson (1969) where optimal control concepts are explicitly used. Whereas the theoretical foundations in these papers rely on idealized assumptions such as “frictionless markets” and “continuous trading,” the main objective in this article is to describe the practical considerations and complexities which arise in a real-world scenario. Accordingly, we drop idealizing assumptions used in the theoretical literature and instead emphasize implementation issues and constraints which are encountered by the practitioner.

Feedback Versus Open-Loop Control

We first elaborate on the distinction between trading a stock via feedback control and its alternative, *open-loop control*. This is done via simple examples: Suppose an investor buys \$1,000 of stock at time $t = 0$ with the a plan to make no changes in this position until some pre-specified future time $t = T$. Then, this *buy-and-hold* trading strategy falls within the realm of open-loop control. If instead, this same investor adds \$1,000 to the position every month, then this type of *dollar-cost averaging* strategy would still fall into the open-loop category. That is, in both scenarios, no information is being used to modify the stock position over time. Finally, suppose this same investor makes a \$1,000 purchase only at the end of those months over which the stock price has decreased. Then this type of *buy-low* investor is now using a simple feedback control strategy because price information is being used to modify the stock position over time. The ability of feedback to compensate for market uncertainty is the *raison d'être* for its use in trading.

Closed-Loop Feedback Configuration

To describe stock trading via feedback control in a more formal manner, the first step involves

creation of a closed-loop feedback configuration involving the trader and the broker; see Fig. 1. In the figure, the feedback controller resides inside the block-labeled “trader,” and there is a wide diversity of possible algorithms which can be used to modify the investment level over time. In some cases, a model for future stock prices is central to the trading algorithm. Oftentimes, no stock price model is used at all, and trading signals are generated based on observed “price patterns.” This falls under the umbrella of “technical analysis” in its purest form; e.g., see the books by Kirkpatrick and Dahlquist (2007) and Lo and Hasanhodzic (2010) for further details. In any event, regardless of the trading method used, the time-varying control signal is the investment level $I(k)$, where the discrete index k represents the k -th possible trading time. The classical case familiar to most investors is $I(k) > 0$, which corresponds to the trader being “long.” That is, stock is purchased in hope of a price increase; see the section to follow on “short selling” for the case $I(k) < 0$.

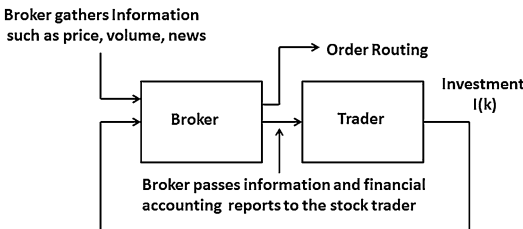
Note that an alternative, but equivalent, specification of the investment level $I(k)$ is in terms of the number of shares of stock it represents. That is, an investment level of $I(k)$ in a stock at price $S(k)$ corresponds to $N(k) = I(k)/S(k)$ shares. In the sequel, for simplicity, we allow $I(k)$ to represent a fractional number of shares. In practice, this type of fractional holding is only allowed in some restricted situations such as reinvestment of dividends or dollar allocations to buy shares of a mutual fund. However, in cases where a

significant number of shares are being bought or sold, the use of fractional shares is a good approximation for all practical purposes.

Letting Δt denote the time between trading opportunities, we note that this quantity can vary greatly depending on the strategy used and the speed of communication between the trader and the broker. For example, Δt corresponding to 24 h might be synonymous with a strategy that trades at the open or close each day. In contrast, for a so-called high-frequency trader, Δt might be on the order of milliseconds or even much less. Even further, one may take the limit as Δt approaches zero, which leads to the purely theoretical situation in which trading occurs in continuous time. Since this article aims to describe real-world stock-trading mechanics as opposed to theoretical results, we work exclusively in discrete index k .

Short Selling: Use of Borrowed Shares

As previously mentioned, we also allow for the possibility that $I(k) < 0$. In this case, the trader is called a *short seller* and the following is meant: At stage k , shares valued at $I(k)$ are borrowed from the broker and immediately sold in the market in the hope that the price will decline. When the trader arrives at $k' > k$, if such a decline has occurred, the short seller can “cover” the position and realize a profit. This is accomplished by buying back the stock and returning the borrowed shares to the broker. Alternatively, if the stock price has increased, the short seller can still cover the position. However, in this case, a realized loss is the result when the shares are returned to the broker. Yet, a third possibility is that the short seller continues to hold the position with a “paper loss” in the hope that the stock price will eventually fall. This being the case, the position can be covered at some future $k'' > k'$ with the end results being either a smaller loss or a gain.



Stock Trading via Feedback Control Methods,
Fig. 1 Feedback loop involving trader and broker

Historical and Model-Generated Prices

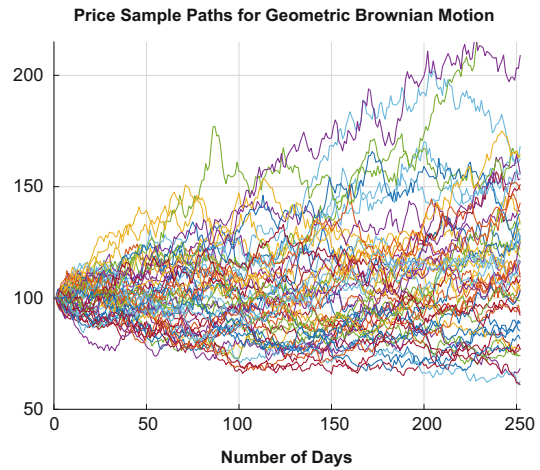
A trading system, be it described by a simulation or real-money implementation, involves the use of sequential price data $S(k)$. This can be obtained either in real time or can be historical stock market data which is available from various recognized sources. For example, data consisting of end-of-day closing prices adjusted for dividends and splits can be downloaded for free from Yahoo! Finance. Another possibility, available from the Wharton Research Data Services for a subscription fee, is the comprehensive database of historical prices at time scales from monthly to tick-by-tick. Finally, we mention the so-called “ITCH” price data associated with messages sent to the NASDAQ market, which is time-stamped to the order of a microsecond or less.

The trader, in testing various strategies, also has the option to generate synthetic data for simulation purposes. For example, one of the most common ways to accomplish this is via the use of a geometric Brownian motion process. That is, a process *drift* μ and a *volatility* $\sigma > 0$, say on an annualized basis, are provided to the simulator, and prices are generated sequentially in time via the recursion

$$S(k+1) = \exp \left(\left[\mu - \frac{1}{2} \sigma^2 \right] \Delta t + \sigma \epsilon(k) \sqrt{\Delta t} \right) S(k)$$

where Δt is measured in years and $\epsilon(k)$ is a zero-mean normally distributed random variable with unit standard deviation. To illustrate sample paths generated along the lines above, we used 252 time steps corresponding to the number of days in a trading year, and generated 50 price paths using $\mu = 0.15$, $\sigma = 0.3$ with initial price $S(0)$ being \$100; see Fig. 2. As an alternative to the above, in many papers, a so-called binomial lattice is used for simulation by many authors.

The reader is referred to the textbooks by Oksendal (2014) and Luenberger (2013) for a detailed description of this celebrated stochastic price model.



Stock Trading via Feedback Control Methods, Fig. 2 Geometric Brownian motion price paths

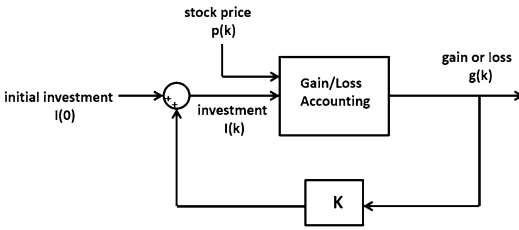
The Feedback Control Law for Trading

In this section, we discuss the previously mentioned mapping taking the information available to the trader to the amount invested $I(k)$. This feedback law is the “heart” of the controller and enables adaptation to uncertain and changing market conditions. One of the simplest examples of a stock-trading feedback control law is obtained using a classical linear time-invariant controller. In this case, at stage k , the level of investment $I(k)$ is obtained using the cumulative gains or losses from trading $g(k)$ according to the formula

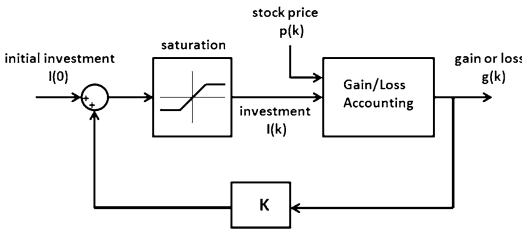
$$I(k) = I_0 + K g(k),$$

where K is a constant feedback gain which is chosen by the trader based on various considerations of risk versus return; see Fig. 3. For example, a trader using a stochastic model such as geometric Brownian motion described above might select K to optimize some risk metric.

Beginning with $g(0) = 0$ and using the feedback law above, the trader initially invests $I(0) = I_0$ in the stock and then begins to monitor the cumulative gain or loss $g(k)$ associated with this investment. Subsequently, $I(k)$ is changed when the position begins to either make or lose



Stock Trading via Feedback Control Methods, Fig. 3 Stock trading via linear feedback



Stock Trading via Feedback Control Methods, Fig. 4 Feedback loop with saturation

money due to the movement of the stock. The constant of proportionality K above, the so-called *feedback gain*, is used to scale the investment level. When I_0 and K are positive, $I(k)$ is positive, and the trade is long. Alternatively, when I_0 and K are negative, $I(k)$ is negative; hence, the trader is a short seller. This type of classical linear feedback is an example of a strategy which falls within the well-known class of “trend followers.”

As a second example, consider a long trade with $I_0, K > 0$ and an investor who wishes to limit the trade to some level $I_{\max} > I_0$. In this case, the feedback loop includes a nonlinear saturation block, see Fig. 4, and the update equation for the investment is

$$I(k) = \min\{I_0 + Kg(k), I_{\max}\}.$$

There are also variations of this scheme involving the notion of “reset,” which assures that excessive time is not spent in the saturation regime when the stock price is falling after a long period of increase resulting in $I(0) + Kg(k) >> I_{\max}$.

To conclude this section, it is important to note that the two examples given above involve controllers which, for simplicity, were taken to be “memoryless.” That is, at stage k , the investment

level $I(k)$ is determined by the current value of the gain-loss function $g(k)$ with no regard for its previous history $g(0), g(1), \dots, g(k-1)$. More generally, $I(k)$ can depend not only on the history of g but also on the history of other system variables. For example, at stage k , a trader might use the price history $S(0), S(1), \dots, S(k-1), S(k)$ to predict the extent to which an ongoing trend will continue. A simple example involving the use of history is the celebrated PI controller. This is a generalization of the previously described linear time-invariant controller; i.e., the investment is now

$$I(k) = I_0 + Kg(k) + K_I \sum_{i=0}^k g(i)$$

where K_I is a so-called *integral gain*. A researcher interested in going beyond the confines of this paper can consult Malekpour et al. (2018) to see the PI controller in action in a more abstract research setting involving continuous-time processes.

Order-Filling Mechanics

In practice, at stage k , the trader specifies the desired investment update to the broker who is responsible for providing a “fill” via interaction with the stock exchange. The way this step is carried out depends on a number of factors: If the stock being purchased is not heavily traded, there may be “liquidity” issues which arise. To provide an example illustrating this liquidity issue, imagine a trader who decides to increment the most recent investment level $I(k-1)$ by \$10,000 to reach the desired level $I(k)$ as prescribed by the feedback law. If there are only 75 shares available at an ask price of \$100, then \$7,500 is initially spent to acquire these shares. This leaves \$2,500 still to be invested. Now suppose there are 300 shares available at the next higher ask price of \$102. Then the \$2,500 expenditure would result in an acquisition of approximately 24.51 shares. Overall, the trader achieves the desired investment with two “partial fills”

and ends up with an average acquisition cost of approximately \$100.49 per share. This type of “book walking” frequently arises for large traders such as hedge funds. For example, if millions of shares are being purchased at time k , the acquisition price of the final shares may be significantly higher than the initial shares. Generalizing on the scenario above, suppose a trader arrives at investment level $I(k)$ via two trades: the first is investment $I_a(k)$ to purchase shares at price $S_a(k)$, and the second is an investment $I_b(k)$ to purchase shares at price $S_b(k)$. Then, the average cost, call it $\bar{S}(k)$, to acquire these shares is readily calculated to be

$$\bar{S}(k) = \frac{S_a(k)S_b(k)(I_a(k) + I_b(k))}{S_a(k)I_b(k) + S_b(k)I_a(k)}.$$

The scenario above corresponds to the steps one might see when a large trader, such as a hedge fund, submits a so-called market order and there is insufficient supply at the current stock price for it to be filled. When this multiple-transaction issue arises in real trading, it will not be possible to predict in advance what price $S(k)$ will result without using a model of the limit order book. Although not covered here, there are a variety of other possible order types which could be considered in a more detailed exposition of trading mechanics. A trader using a broker’s platform can place orders with various contingencies.

In the case when a stock trades with large daily volume, if large “market movers” are not transacting, it may be reasonable to assume in simulations that the trader is a *price taker*. That is, one assumes the bid-ask spread is zero, and trading is said to be “highly liquid.” This being the case, the trader obtains as many shares as desired at the current stock price $S(k)$.

Gain-Loss Accounting

A broker generally provides frequent updates on the account value $V(k)$, gains and losses $g(k)$, and *transaction costs* $T(k)$. The update of the *gain-loss* function $g(k)$ from time k to $k + 1$ is given by

$$g(k + 1) = g(k) + \rho(k)I(k) - T(k)$$

with

$$\rho(k) \doteq \frac{S(k + 1) - S(k)}{S(k)}$$

being the *percentage change* in the stock price from k to $k + 1$.

As far as transaction costs $T(k)$ are concerned, these consist primarily of the broker’s commission, exchange fees, and taxes. Nowadays, the broker’s commission is much smaller than it was decades ago. For example, in the 1990s, the commission on a trade could easily exceed \$20 with surcharges based on the number of shares. In contrast, at present, using a discount broker, one can easily obtain commission rates significantly less than \$5 per trade, even when a large number of shares are being transacted.

Margin Charges and Interest Accumulation

In a brokerage account, a trader who uses *leverage* borrows funds from the broker to purchase stock which would otherwise be unaffordable with the cash in the account. This is referred to as *trading on margin*. For such cases, the broker will charge an interest rate, known as the *margin rate*, on the borrowed funds. While in practice there is a limit on how much money can be borrowed, it can be quite large; e.g., a hedge fund can often obtain access to many multiples of its account value, whereas small traders might typically be able to borrow up to 100% of the account value. In terms of modeling a simulation, at stage k , if a trader is going long, it would be typical for the broker to impose *leverage constraint* $I(k) \leq 2V(k)$.

When a trader is not fully invested and the account contains “idle cash,” it is typical to receive interest from the broker on this money. On a per-period basis, letting m be the margin rate and r_f the interest rate, often referred to as being risk free, it is generally the case that $m > r_f$. At the writing of this paper, the spread $m - r_f$ is quite large compared to most other periods in history. Nowadays, on an annualized basis, the

margin rate is typically over 7%, and the interest rate is less than 1%.

To cover both interest received and margin charges to be paid, we work with the *interest accrual*, $A(k)$, be it positive or negative. For the period Δt corresponding to the transition from k to $k + 1$, this quantity is given by

$$A(k) = r_f \max\{V(k) - I(k), 0\} \\ + m \min\{V(k) - I(k), 0\}.$$

which either increments or decrements the account value $V(k)$. For the case of a short seller with $I(k) < 0$, the formula above should only be used for traders with sufficiently large accounts who have enough “clout” with the broker so as to be allowed to capitalize on the proceeds of a short sale. For a small- to medium-size trading account, the short-sale proceeds are generally “held aside,” and the account is “marked to market” on a daily basis. As a result, for this trader, the $A(k)$ equation above needs to be revised to account for “cash in reserve” and turns out to provide smaller interest rate accruals.

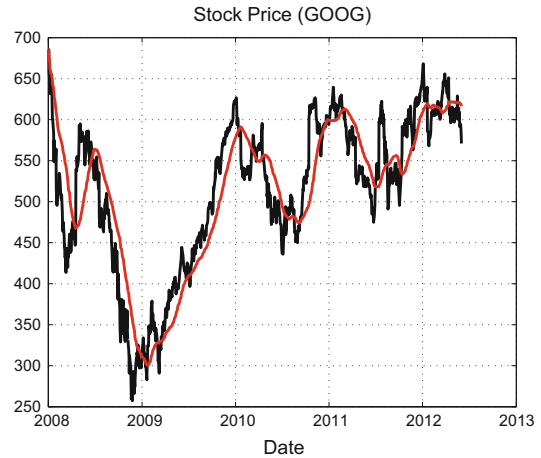
Finally, the broker’s reporting, as previously mentioned, generally includes the entire value of the account $V(k)$. This number is made up of the stock positions, either idle or borrowed cash and “dividends” $D(k)$ paid by the company to the trader holding a long position during this period. Alternatively, in the case of a short seller, $D(k)$ becomes a liability and is treated as a decrement to the account value. In either event, the account value update is given by

$$V(k+1) = V(k) + \rho(k)I(k) + A(k) + D(k) - T(k)$$

which is generally updated in real time.

Leverage Constraints and Margin Calls

As previously mentioned, it is important to take account of the fact that the broker limits the size of the trader’s investment $I(k)$ by imposing a leverage constraint. When a long stock position falls dramatically, a “crisis” of sorts can ensue



Stock Trading via Feedback Control Methods,
Fig. 5 Google closing prices

for the trader. To illustrate, recalling the example above with leverage constraint $I(k) \leq 2V(k)$, when this inequality is violated, the account is viewed by the broker as being inadequately collateralized, and a “margin call” is triggered. Subsequently, any new orders by the trader are “stopped.” To continue trading and avoid forced liquidation of current positions, the account must be brought back into compliance with the leverage constraint by a deposit of new assets or cash within a short pre-specified time period.

Simulation Example

We now provide a simulation example to illustrate some of the concepts in this paper. Figure 5 shows the daily closing prices from January 1, 2008, to June 1, 2012, of Google (GOOG), traded on the NASDAQ stock exchange. The figure also includes the 50-day simple moving average $S_{av}(k)$ which will be used in a feedback controller with investments triggered by sign changes in $S(k) - S_{av}(k)$. This is a moving average crossing strategy; e.g., see Brock et al. (1992).

To initiate the strategy, there is no trading during the first 50 days while the moving average is first established. Subsequently, a trade is triggered at the first instant $k = k^*$ when the moving average $S_{av}(k)$ has been crossed by the

stock price $S(k)$. For $k \geq k^*$, the feedback law for the investment level is given by

$$I(k) = I_0 \text{sign}\{S(k) - S_{av}(k)\}$$

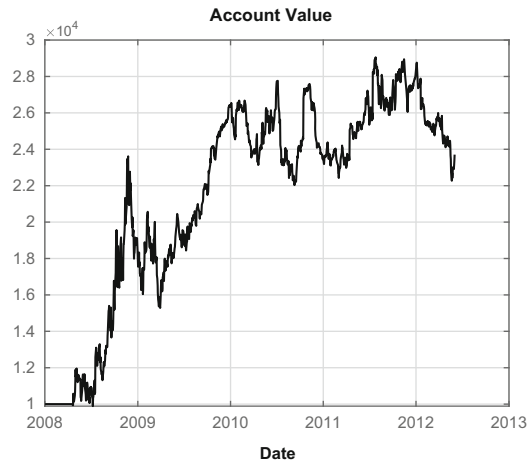
with I_0 being \$20,000. Furthermore, assuming the initial account value $V(0)$ is \$10,000, the issues of margin and interest are immediately in play. In the simulation, we use a risk-free rate of 1.5% per annum and a margin rate of 3% per annum, corresponding approximately to per-period daily rates of $r_f = 6 \times 10^{-5}$ and $m = 12 \times 10^{-5}$. It is assumed that interest may be obtained on the proceeds of short sales at the risk-free rate. This corresponds to the use of the $A(k)$ equation above without modification. During this period, Google did not pay a dividend; hence, $D(k) = 0$ for all k .

A transaction cost of \$3 per trade is charged and occurs every day of trading because the position is being adjusted to achieve the target investment level $I(k) = \pm 20,000$ above. Finally, we assume in our simulation that each day a “market-on-close” order is executed at the closing price.

Figure 4 shows the evolution of the account value $V(k)$ over time. While the Google price trends up and down over the simulation window, the controller based on the moving average adapts its investment level to capture the major trends. This leads to a significant overall gain of approximately 140% on the initial account value, despite the fact that Google is slightly down over the simulation period.

Summary and Future Directions

A primary objective of this article was to concentrate almost entirely on trading mechanics and the equations which are needed for simulation in a control-theoretic framework (Fig. 6). As discussed in the introduction, a variety of risk metrics and theoretical considerations could be brought into play in a more full-blown analysis. It should also be noted that there is a growing body of literature on stock trading and financial markets with a control-theoretic flavor.



Stock Trading via Feedback Control Methods,
Fig. 6 Account value from feedback trading of Google

It is convenient to subdivide this literature into two categories: The first category, called *model-based* approaches, involves an underlying parameterized model for stock prices which may or may not be completely specified. In the second category of papers, called *model-free* approaches, the stock price is viewed as an external input with no predictive model for its evolution. In addition, no parameter estimation is involved, and feedback trade signals are generated based on some observed “pattern” of prices or trading gains. For the uninitiated reader, one entry point into the literature would be the tutorial paper by Barmish et al. (2013).

Cross-References

- [Financial Markets Modeling](#)
- [Option Games: The Interface Between Optimal Stopping and Game Theory](#)
- [Risk-Sensitive Stochastic Control](#)

Bibliography

- Artzner P, Delbaen F, Eber J, Heath D (1999) Coherent measures of risk. *J Math Finance* 9:203–208
- Astrom KJ, Murray RM (2008) *Feedback systems, an introduction for scientists and engineers*. Princeton University Press, Princeton

- Barmish BR, Primbs JA (2016) On a new paradigm for stock trading via a model-free feedback controller. *IEEE Trans Autom Control* 61(3):662–676
- Barmish BR, Primbs JA, Malekpour S, Warnick S (2013) On the basics for simulation of feedback-based stock trading strategies, an invited tutorial. In: *Proceedings of IEEE conference on decision and control*. IEEE
- Brock W, Lakonishok J, LeBaron, B (1992) Simple technical trading rules and the stochastic properties of stock returns. *J Finance* 47:1731–1764
- Kirkpatrick CD, Dahlquist JR (2007) *Technical analysis: the complete resource for financial market technicians*. Financial Times Press, New York
- Lo AW, Hasanhodzic J (2010) *The evolution of technical analysis: financial prediction from Babylonian tablets to Bloomberg terminals*. Bloomberg Press, New York
- Luenberger DG (2013) *Investment science*, 2nd edn. Oxford, London
- Malekpour S, Primbs JA, Barmish BR (2018) A generalization of simultaneous long-short stock trading to PI controllers. *IEEE Trans Autom Control* 63(10):3531–3536
- Merton RC (1969) Lifetime portfolio selection under uncertainty: the continuous time case. *Rev Econ Stat* 51:247–257
- Oksendal B (2014) *Stochastic differential equations: an introduction with applications*, 6th edn. Springer, New York
- Samuelson PA (1969) Lifetime portfolio selection by dynamic stochastic programming. *Rev Econ Stat* 51:239–246

Strategic Form Games and Nash Equilibrium

Asuman Ozdaglar
Laboratory for Information and Decision
Systems, Department of Electrical Engineering
and Computer Science, Massachusetts Institute
of Technology, Cambridge, MA, USA

Abstract

This chapter introduces strategic form games, which provide a framework for the analysis of strategic interactions in multi-agent environments. We present the main solution concept in strategic form games, *Nash equilibrium*, and provide tools for its systematic study. We present fundamental results for existence and uniqueness of Nash equilibria and discuss

their efficiency properties. We conclude with current research directions in this area.

Keywords

Efficiency · Existence · Nash equilibrium · Strategic form games · Uniqueness

Synonyms

Nash Equilibrium

Introduction

Many problems in communication, decision, and technological networks as well as in social and economic situations depend on human choices, which are made in anticipation of the behavior of the others in the system. Examples include how to map your drive over a road network, how to use the communication medium, and how to choose strategies for resource use and more conventional economic, financial, and social decisions such as which products to buy, which technologies to invest in, or who to trust. The defining feature of all of these interactions is the dependence of an agent's objective (payoff, utility, or survival) on others' actions. Game theory focuses on formal analysis of such strategic interactions. Here, we will review strategic form games, which focus on static game-theoretic interactions and present the relevant solution concept.

Strategic Form Games

A *strategic form* game is a model for a static game in which all players act simultaneously without knowledge of other players' actions.

Definition 1 (Strategic Form Game) A strategic form game is a triplet $\langle \mathcal{I}, (S_i)_{i \in \mathcal{I}}, (u_i)_{i \in \mathcal{I}} \rangle$ where:

1. $\mathcal{I}$ is a finite set of players, $\mathcal{I} = \{1, \dots, I\}$.

2. S_i is a nonempty set of available actions for player i .
3. $u_i : S \rightarrow \mathbb{R}$ is the utility (payoff) function of player i where $S = \prod_{i \in \mathcal{I}} S_i$.

We will use the terms *action* and (*pure*) *strategy* interchangeably. (We will later use the term “mixed strategy” to refer to randomizations over actions.) We denote by $s_i \in S_i$ an action for player i , and by $s_{-i} = [s_j]_{j \neq i}$ a vector of actions for all players *except* i . We refer to the tuple $(s_i, s_{-i}) \in S$ as an *action (strategy) profile* or *outcome*. We also denote by $S_{-i} = \prod_{j \neq i} S_j$ the set of actions (strategies) of all players except i . Our convention throughout will be that each player i is interested in action profiles that “maximize” his utility function u_i .

The next two examples illustrate strategic form games with finite and infinite strategy sets.

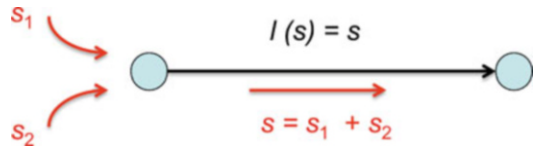
Example (Finite Strategy Sets) We consider a two-player game with finite strategy sets. Such a game can be represented in matrix form, where the rows correspond to the actions of player 1 and columns represent the actions of player 2. The cell indexed by row x and column y contains a pair (a, b) , where a is the payoff to player 1 and b is the payoff to player 2, i.e., $a = u_1(x, y)$ and $b = u_2(x, y)$. This class of games is sometimes referred to as *bimatrix games*. For example, consider the following game of “Matching Pennies.”

	HEADS	TAILS
HEADS	-1, 1	1, -1
TAILS	1, -1	-1, 1

Matching Pennies

This game represents “pure conflict” in the sense that one player’s utility is the negative of the utility of the other player, i.e., the sum of the utilities for both players at each outcome is “zero.” This class of games is referred to as *zero-sum games* (or *constant-sum games*) and has been studied extensively in the game theory literature (Basar and Olsder 1995).

Example (Infinite Strategy Sets) We next present a game with infinite strategy sets. We consider



Strategic Form Games and Nash Equilibrium, Fig. 1
A network game with two players

a simple network game where two players send data or information flows over a communication network represented by a single link. Each player i derives a value for sending s_i units of flow over the link given by

$$v_i(s_i) = \begin{cases} a_i s_i - \frac{s_i^2}{2} & \text{if } s_i \leq a_i, \\ \frac{a_i^2}{2} & \text{if } s_i \geq a_i, \end{cases}$$

where $a_i \in [0, 1]$ is a player-specific scalar. Each player also incurs a per-flow delay or latency cost, due to congestion on the link, represented by the function $l(s) = s$, where s is the total flow on the link, i.e., $s = s_1 + s_2$ (see Fig. 1). The resulting interactions can be represented by the strategic form game $\langle \mathcal{I}, (S_i), (u_i) \rangle$, which consists of:

1. A set of two players, $\mathcal{I} = 1, 2$
2. A strategy set $S_i = [0, 1]$ for each player i , where $s_i \in S_i$ represents the amount of flow player i sends over the link
3. A utility function u_i for each player i given by value derived from sending s_i units of flow minus the total latency cost, i.e.,

$$u_i(s_1, s_2) = v_i(s_i) - s_i l(s_1 + s_2).$$

Nash Equilibrium

We next introduce the fundamental solution concept for strategic form games, *Nash equilibrium*. A Nash equilibrium captures a steady state of the play in a strategic form game such that each player acts optimally given their “correct” conjectures about the behavior of the other players.

Definition 2 (Nash Equilibrium) A (*pure strategy*) *Nash equilibrium* of a strategic form game $\langle \mathcal{I}, (S_i), (u_i)_{i \in \mathcal{I}} \rangle$ is a strategy profile $s^* \in S$ such that for all $i \in \mathcal{I}$, we have

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) \quad \text{for all } s_i \in S_i.$$

Hence, a Nash equilibrium is a strategy profile s^* such that no player i can profit by unilaterally deviating from his strategy s_i^* , assuming every other player j follows his strategy s_j^* . The definition of a Nash equilibrium can be restated in terms of best-response correspondences.

Definition 3 (Nash Equilibrium – Restated)

Let $\langle \mathcal{I}, (S_i), (u_i)_{i \in \mathcal{I}} \rangle$ be a strategic form game. For any $s_{-i} \in S_{-i}$, consider the best-response correspondence of player i , $B_i(s_{-i})$, given by

$$B_i(s_{-i}) = \{s_i \in S_i \mid u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i}) \text{ for all } s'_i \in S_i\}.$$

We say that an action profile s^* is a *Nash equilibrium* if

$$s_i^* \in B_i(s_{-i}^*) \quad \text{for all } i \in \mathcal{I}.$$

Thus, if we define the best-response correspondence $B(s) = [B_i(s_{-i})]_{i \in \mathcal{I}}$, the set of Nash equilibria is given by the set of fixed points of $B(s)$. Below, we give two examples of games with pure strategy Nash equilibria.

Example (Battle of the Sexes) Consider a two-player game with the following payoff structure:

	BATTLE SOCCER	
BATTLE	2, 1	0, 0
SOCCER	0, 0	1, 2

Battle of the Sexes

This game, referred to as the Battle of the Sexes game, represents a scenario in which the two players wish to coordinate their actions but have different preferences over their actions. This game has two pure strategy Nash equilibria,

i.e., the strategy profiles (BATTLE, BATTLE) and (SOCCER, SOCCER).

Example Recall the network game given in Example 132. To simplify the computations, let us assume without loss of generality that $a_1 \geq a_2 \geq \frac{a_1}{3}$. It can be seen that the best-response functions (single-valued in this case) of the players are given by

$$B_i(s_{-i}) = \max \left\{ 0, \frac{a_i - s_{-i}}{3} \right\} \quad \text{for } i = 1, 2.$$

The unique pure strategy Nash equilibrium of this game is the fixed point of these functions given by

$$(s_1^*, s_2^*) = \left(\frac{3a_1 - a_2}{8}, \frac{3a_2 - a_1}{8} \right).$$

Mixed Strategy Nash Equilibrium

Consider the two-player “penalty kick” game between a penalty taker and a goalkeeper that has the same payoff structure as the matching pennies:

	LEFT	RIGHT
LEFT	1, -1	-1, 1
RIGHT	-1, 1	1, -1

Penalty kick game

This game does not have a pure strategy Nash equilibrium. It can be verified that if the penalty taker (column player) commits to a pure strategy, e.g., chooses LEFT, then the best response of the goalkeeper (row player) would be to choose the same side leading to a payoff of -1 for the penalty taker. In fact, the penalty taker would be better off following a strategy which randomizes between LEFT and RIGHT, ensuring that the goalkeeper cannot perfectly match his action. This is the idea of “randomized” or mixed strategies which we will discuss next.

We first introduce some notation. Let Σ_i denote the set of probability measures over the pure strategy (action) set S_i . We use $\sigma_i \in \Sigma_i$ to denote the *mixed strategy* of player i . When S_i is a finite set, a mixed strategy is a finite-dimensional probability vector, i.e., a vector

whose elements denote the probability with which a particular action will be played. For example, if S_i has two elements, the set of mixed strategies Σ_i is the one-dimensional probability simplex, i.e., $\Sigma_i = \{(x_1, x_2) \mid x_i \geq 0, x_1 + x_2 = 1\}$. We use $\sigma \in \Sigma = \prod_{i \in \mathcal{I}} \Sigma_i$ to denote a *mixed strategy profile*. Note that this implicitly assumes that players randomize independently. We similarly denote $\sigma_{-i} \in \Sigma_{-i} = \prod_{j \neq i} \Sigma_j$.

Following von Neumann-Morgenstern expected utility theory, we extend the payoff functions u_i from S to Σ by

$$u_i(\sigma) = \int_S u_i(s) d\sigma(s),$$

i.e., the payoff of a mixed strategy σ is given by the expected value of pure strategy payoffs under the distribution σ .

We are now ready to define the mixed strategy Nash equilibrium.

Definition 4 (Mixed Strategy Nash Equilibrium) A mixed strategy profile σ^* is a *mixed strategy Nash equilibrium* if for each player i ,

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_i, \sigma_{-i}^*) \quad \text{for all } \sigma_i \in \Sigma_i.$$

Note that since $u_i(\sigma_i, \sigma_{-i}^*) = \int_{S_i} u_i(s_i, \sigma_{-i}^*) d\sigma_i(s_i)$, it is sufficient to check only *pure* strategy “deviations” when determining whether a given profile is a Nash equilibrium. This leads to the following characterization of a mixed strategy Nash equilibrium.

Proposition 4 A mixed strategy profile σ^* is a *mixed strategy Nash equilibrium* if and only if for each player i ,

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s_i, \sigma_{-i}^*) \quad \text{for all } s_i \in S_i.$$

We also have the following useful characterization of a mixed strategy Nash equilibrium in finite strategy set games.

Proposition 5 Let $G = \langle \mathcal{I}, (S_i)_{i \in \mathcal{I}}, (u_i)_{i \in \mathcal{I}} \rangle$ be a strategic form game with finite strategy sets. Then, $\sigma^* \in \Sigma$ is a Nash equilibrium if and only

if for each player $i \in \mathcal{I}$, every pure strategy in the support of σ_i^* is a best response to σ_{-i}^* .

Proof. Let σ^* be a mixed strategy Nash equilibrium, and let $E_i^* = u_i(\sigma_i^*, \sigma_{-i}^*)$ denote the expected utility for player i . By Proposition 4, we have

$$E_i^* \geq u_i(s_i, \sigma_{-i}^*) \quad \text{for all } s_i \in S_i.$$

We first show that $E_i^* = u_i(s_i, \sigma_{-i}^*)$ for all s_i in the support of σ_i^* (combined with the preceding relation, this proves one implication). Assume to arrive at a contradiction that this is not the case, i.e., there exists an action s_i' in the support of σ_i^* such that $u_i(s_i', \sigma_{-i}^*) < E_i^*$. Since $u_i(s_i, \sigma_{-i}^*) \leq E_i^*$ for all $s_i \in S_i$, this implies that

$$\sum_{s_i \in S_i} \sigma_i^*(s_i) u_i(s_i, \sigma_{-i}^*) < E_i^*,$$

which is a contradiction. The proof of the other implication is similar and is therefore omitted. $\triangleleft$

It follows from this characterization that every action in the support of any player’s equilibrium mixed strategy yields the same payoff. This characterization extends to games with infinite strategy sets: $\sigma^* \in \Sigma$ is a Nash equilibrium if and only if for each player $i \in \mathcal{I}$, given σ_{-i}^* , no action in S_i yields a payoff that exceeds his equilibrium payoff, and the set of actions that yields a payoff less than his equilibrium payoff has σ_i^* -measure zero.

Example Let us return to the Battle of the Sexes game.

	Ballet Soccer	
Ballet	2, 1	0, 0
Soccer	0, 0	1, 2

Battle of the Sexes

Recall that this game has 2 pure strategy Nash equilibria. Using the characterization result in Proposition 5, we show that it has a *unique* mixed strategy Nash equilibrium (which is not a

pure strategy Nash equilibrium). First, by using Proposition 5 (and inspecting the payoffs), it can be seen that there are no Nash equilibria where only one of the players randomizes over its actions. Now, assume instead that player 1 chooses the action BALLET with probability $p \in (0, 1)$ and SOCCER with probability $1 - p$ and that player 2 chooses BALLET with probability $q \in (0, 1)$ and SOCCER with probability $1 - q$. Using Proposition 5 on player 1's payoffs, we have the following relation:

$$2 \times q + 0 \times (1 - q) = 0 \times q + 1 \times (1 - q).$$

Similarly, we have

$$1 \times p + 0 \times (1 - p) = 0 \times p + 2 \times (1 - p).$$

We conclude that the only possible mixed strategy Nash equilibrium is given by $q = \frac{1}{3}$ and $p = \frac{2}{3}$.

Existence of Nash Equilibrium

The first question that one contemplates in analyzing a strategic form game is whether it has a pure or mixed strategy Nash equilibrium. While it may be possible to explicitly construct a Nash equilibrium (using either computational means or characterization results), this may be a tedious task in the case of both large finite strategy set games or infinite strategy set games with complicated utility functions. One is therefore often interested in establishing existence of an equilibrium, using conditions on the utility functions and constraint sets, before trying to understand its properties. In the sequel, we present results on existence of an equilibrium for games with finite and infinite strategy sets. The proofs of such existence results typically use fixed point arguments on the best-response correspondences of the players. They are omitted here and can be found in graduate-level game theory text books (see Fudenberg and Tirole 1991 and Myerson 1991).

Finite Strategy Set Games

We have seen that while the matching pennies game (and the penalty kick game with the same payoff structure) does not have a pure strategy Nash equilibrium, it has a mixed strategy Nash equilibrium. The next theorem, states that this existence result extends to all finite strategy set games.

Theorem 1 (Nash) *Every strategic form game with finite strategy sets has a mixed strategy Nash equilibrium.*

Infinite Strategy Set Games

A stronger result on existence of a pure strategy Nash equilibrium can be established in infinite strategy set games under some topological conditions on the utility functions and constraint sets (see Debreu 1952, Fan 1952, and Glicksberg 1952).

Theorem 2 (Debreu, Fan, Glicksberg) *Consider a strategic form game $\langle I, (S_i)_{i \in I}, (u_i)_{i \in I} \rangle$ with infinite strategy sets such that for each $i \in I$:*

1. S_i is convex and compact.
2. $u_i(s_i, s_{-i})$ is continuous in s_{-i} .
3. $u_i(s_i, s_{-i})$ is continuous and quasiconcave in s_i . (Let X be a convex set. A function $f : X \rightarrow \mathbb{R}$ is quasiconcave if every upper level set of the function, i.e., $\{x \in X \mid f(x) \geq \alpha\}$ for every scalar α , is a convex set (see Bertsekas et al. 2003).)

The game has a pure strategy Nash equilibrium.

Note that Theorem 1 is a special case of this result. For games with finite strategy sets, mixed strategy sets are simplices and hence are convex and compact, and utilities are linear in (mixed) strategies; hence, they are concave functions of (mixed) strategies (and continuous functions of mixed strategy profiles).

The next example shows that quasiconcavity cannot be dispensed with in the previous existence result.

Example Consider the game where two players pick a location $s_1, s_2 \in \mathbb{R}^2$ on the circle. The payoffs are

$$u_1(s_1, s_2) = -u_2(s_1, s_2) = d(s_1, s_2),$$

where $d(s_1, s_2)$ denotes the Euclidean distance between s_1 and $s_2 \in \mathbb{R}^2$. It can be verified that this game does not have a pure strategy Nash equilibrium. However, the strategy profile where both players mix uniformly on the circle is a mixed strategy Nash equilibrium.

Without quasiconcavity, one can establish the following existence result (see Glicksberg 1952).

Theorem 3 (Glicksberg) *Consider a strategic form game $\langle \mathcal{I}, (S_i)_{i \in \mathcal{I}}, (u_i)_{i \in \mathcal{I}} \rangle$, where the S_i are nonempty compact metric spaces and the $u_i : S \rightarrow \mathbb{R}$ are continuous functions. The game has a mixed strategy Nash equilibrium.*

Uniqueness of Nash Equilibrium

Another important question that arises in the analysis of strategic form games is whether the Nash equilibrium is unique. This is important for the predictive power of Nash equilibrium since with multiple equilibria, the outcome of the game cannot be uniquely pinned down. The following result by Rosen provides sufficient conditions for uniqueness of an equilibrium in games with infinite strategy sets (see Rosen 1965). (Except for games that are strictly dominant solvable, there are no general uniqueness results for finite strategic form games.)

We first introduce some notation to state this result. Given a scalar-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, we use the notation $\nabla f(x)$ to denote the gradient vector of f at point x , i.e.,

$$\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n} \right]^T.$$

Given a scalar-valued function $F : \prod_{i=1}^I \mathbb{R}^{m_i} \rightarrow \mathbb{R}$, we use the notation $\nabla_i F(x)$ to denote the gradient vector of F with respect to x_i at point x , i.e.,

$$\nabla_i F(x) = \left[\frac{\partial F(x)}{\partial x_i^1}, \dots, \frac{\partial F(x)}{\partial x_i^{m_i}} \right]^T.$$

We use the notation $\nabla F(x)$ to denote

$$\nabla F(x) = [\nabla_1 F_1(x), \dots, \nabla_I F_I(x)]^T. \quad (1)$$

We assume that the strategy set S_i of each player i is given by

$$S_i = \{x_i \in \mathbb{R}^{m_i} \mid h_i(x_i) \geq 0\}, \quad (2)$$

where $h_i : \mathbb{R}^{m_i} \mapsto \mathbb{R}$ is a concave function. (Since h_i is concave, it follows that the set S_i is a convex set.) The next definition introduces the key condition used in establishing the uniqueness of a pure strategy Nash equilibrium.

Definition 5 We say that the utility functions $(u_1, \dots, u_I)$ are **diagonally strictly concave** for $x \in S$, if for every $x^*, \bar{x} \in S$, we have

$$(\bar{x} - x^*)^T \nabla u(x^*) + (x^* - \bar{x})^T \nabla u(\bar{x}) > 0.$$

We can now state the result on uniqueness of pure strategy Nash equilibrium in strategic form games.

Theorem 4 (Rosen) *Consider a strategic form game $\langle \mathcal{I}, (S_i), (u_i) \rangle$. For all $i \in \mathcal{I}$, assume that the strategy sets S_i are given by Eq. (2), where h_i is a concave function, and there exists some $\tilde{x}_i \in \mathbb{R}^{m_i}$ such that $h_i(\tilde{x}_i) > 0$. Assume also that the utility functions $(u_1, \dots, u_I)$ are diagonally strictly concave for $x \in S$. Then, the game has a unique pure strategy Nash equilibrium.*

We next provide a tractable sufficient condition for the utility functions to be diagonally strictly concave. Let $U(x)$ denote the Jacobian of $\nabla u(x)$ [see Eq. (1)]. Specifically, if the x_i are all 1-dimensional, then $U(x)$ is given by

$$U(x) = \begin{pmatrix} \frac{\partial^2 u_1(x)}{\partial x_1^2} & \frac{\partial^2 u_1(x)}{\partial x_1 \partial x_2} & \dots \\ \frac{\partial^2 u_2(x)}{\partial x_2 \partial x_1} & \ddots & \\ \vdots & & \end{pmatrix}.$$

Proposition 6 (Rosen) *For all $i \in \mathcal{I}$, assume that the strategy sets S_i are given by Eq. (2), where h_i is a concave function. Assume that the symmetric matrix $(U(x) + U^T(x))$ is negative*

definite for all $x \in S$, i.e., for all $x \in S$, we have

$$y^T (U(x) + U^T(x))y < 0, \quad \forall y \neq 0.$$

Then, the payoff functions $(u_1, \dots, u_I)$ are diagonally strictly concave for $x \in S$.

Rosen's sufficient conditions for uniqueness are quite strong. Recent work has extended such uniqueness results to hold under weaker conditions using differential topology tools. The main idea is to provide sufficient conditions so that the indices of all stationary points can be shown to be positive, which from a generalization of the Poincare-Hopf theorem (Simsek et al. 2007, 2008) implies that there exists a unique equilibrium (see Simsek et al. 2005 for applications of this methodology to several network games).

Efficiency of Nash Equilibria

Because the Nash equilibrium corresponds to the fixed point of the best-response correspondences of the players, there is no presumption that it is efficient or maximizes any well-defined weighted sum of utility functions of the players. This fact is clearly illustrated by the well-known Prisoner's Dilemma game. For some $a > 0, b > 0$, and $c > 0$ with $a > b$, the payoff matrix is given by:

	DON'T CONFESS	CONFESS
DON'T CONFESS	a, a	$b - c, a + c$
CONFESS	$a + c, b - c$	b, b

Prisoner's Dilemma

This game, generally used for capturing the dilemma of cooperation among selfish agents, has a unique (pure strategy) Nash equilibrium. (In fact each player has a dominant strategy, see Fudenberg and Tirole 1991, which is (CONFESS, CONFESS)). This clearly illustrates two aspects of the inefficiencies that arise in Nash equilibria. First, the unique Nash equilibrium is Pareto inferior meaning that if both players cooperated and chose DON'T CONFESS, they would both obtain the higher payoff of a . Second, the extent of inefficiency can be arbitrarily large based on

the values of a and b . We can capture this by the *efficiency loss* (or *Price of Anarchy* as known in the literature) defined as

$$\text{Efficiency Loss} = \inf_{\text{parameters}} \frac{\sum_i u_i(\text{equilibrium})}{\sum_i u_i(\text{social optimum})},$$

where the social optimum is the strategy profile that maximizes the sum of utility functions. In the preceding example, this is clearly

$$\inf_{a,b} \frac{b}{a} = 0,$$

showing that efficiency loss can be arbitrarily large. In problems that have more structure, the efficiency loss can be bounded away from zero. A well-known example is by Pigou, which showed that in a network routing game where the congestion penalty can be described by linear latency functions (see Example 132), the efficiency loss is $3/4$ (Pigou 1920). Roughgarden and Tardos in an important contribution (Roughgarden and Tardos 2000) showed that this is a lower bound for such routing games over all possible network topologies.

Summary and Future Directions

This article has provided an introduction to the basics of strategic form games. After defining the concept of Nash equilibrium, which is the basis of much of recent game theory, we have presented fundamental results on its existence and uniqueness. We also briefly discussed issues of efficiency of Nash equilibria.

Though game theory is a mature field, there are still several important areas for inquiry. The first is a more systematic analysis and categorization of classes of games by their equilibrium and efficiency properties. Recent work by Candogan et al. (2010, 2011, 2013) provides tools for systematically analyzing equivalence classes of games that may be useful for such an investigation. The second area that is very much active concerns computational issues, which we have not considered here. Recent literature showed

that computation of Nash equilibria in finite strategy set games is potentially hard and focused on developing algorithms for computing approximate Nash equilibria (see Daskalakis et al. 2006 and Lipton et al. 2003). Ongoing research in this area focuses on infinite strategy set games and exploits special structure to develop algorithms for computing (exact and approximate) Nash equilibria (Parrilo 2006; Stein et al. 2008). A third area is to develop a better application of tools of strategic form games and understand the resulting efficiency losses in networks and large-scale systems. Work in this area uses game-theoretic models to investigate resource allocation, pricing, and investment problems in networks (Johari and Tsitsiklis 2004; Acemoglu and Ozdaglar 2007; Acemoglu et al. 2009; Njoroge et al. 2013). A fourth area of research is to develop and apply alternative solution concepts for strategic form games. While some of the research in game theory has focused on subsets of Nash Equilibria (see Fudenberg and Tirole 1991), from a computational point of view, the set of correlated equilibria, which is a superset of the set of Nash Equilibria, is also attractive since it can be represented as the optimal solution set of a linear program. Correlated equilibrium can be implemented using a correlation scheme (a trusted party) or cryptographic tools as shown in Izmalkov et al. (2007). Recent work investigates alternative solution concepts for symmetric games intermediate between Nash and correlated equilibria (Stein et al. 2013), which can be implemented using specific correlation schemes.

Cross-References

- [Dynamic Noncooperative Games](#)
- [Game Theory: A General Introduction and a Historical Overview](#)
- [Linear Quadratic Zero-Sum Two-Person Differential Games](#)

Bibliography

Acemoglu D, Ozdaglar A (2007) Competition and efficiency in congested markets. *Math Oper Res* 32(1):1–31

- Acemoglu D, Bimpikis K, Ozdaglar A (2009) Price and capacity competition. *Games Econ Behav* 66(1):1–26
- Basar T, Olsder GJ (1995) *Dynamic noncooperative game theory*. Academic, London/New York
- Bertsekas D, Nedic A, Ozdaglar A (2003) *Convex analysis and optimization*. Athena Scientific, Belmont
- Candogan O, Ozdaglar A, Parrilo PA (2010) A projection framework for near-potential games. In: *Proceedings of the IEEE conference on decision and control, CDC, Atlanta*
- Candogan O, Menache I, Ozdaglar A, Parrilo PA (2011) Flows and decompositions of games: harmonic and potential games. *Math Oper Res* 36(3):474–503
- Candogan O, Ozdaglar A, Parrilo PA (2013, forthcoming) Dynamics in near-potential games. *Games Econ Behav* 82:66–90
- Daskalakis C, Goldberg PW, Papadimitriou CH (2006) The complexity of computing a Nash equilibrium. In: *Proceedings of the 38th ACM symposium on theory of computing, STOC, Seattle*
- Debreu D (1952) A social equilibrium existence theorem. *Proc Natl Acad Sci* 38:886–893
- Fan K (1952) Fixed point and minimax theorems in locally convex topological linear spaces. *Proc Natl Acad Sci* 38:121–126
- Fudenberg D, Tirole J (1991) *Game theory*. MIT, Cambridge
- Glicksberg IL (1952) A further generalization of the Kakutani fixed point theorem with application to Nash equilibrium points. *Proc Natl Acad Sci* 38:170–174
- Izmalkov S, Lepinski M, Micali S, Shalat A (2007) *Transparent computation and correlated equilibrium*. Working paper
- Johari R, Tsitsiklis JN (2004) Efficiency loss in a network resource allocation game. *Math Oper Res* 29(3):407–435
- Lipton RJ, Markakis E, Mehta A (2003) Playing large games using simple strategies. In: *Proceedings of the ACM conference in electronic commerce, EC, San Diego*
- Myerson RB (1991) *Game theory: analysis of conflict*. Harvard University Press, Cambridge
- Njoroge P, Ozdaglar A, Stier-Moses N, Weintraub G (2013) Investment in two-sided markets and the net neutrality debate. *Forthcoming in Review of Network Economics*
- Parrilo PA (2006) Polynomial games and sum of squares optimization. In: *Proceedings of the IEEE conference on decision and control, CDC, San Diego*
- Pigou AC (1920) *The economics of welfare*. Macmillan, London
- Rosen JB (1965) Existence and uniqueness of equilibrium points for concave N-person games. *Econometrica* 33(3):520–534
- Roughgarden T, Tardos E (2000) How bad is selfish routing? In: *Proceedings of the IEEE symposium on foundations of computer science, FOCS, Redondo Beach*
- Simsek A, Ozdaglar A, Acemoglu D (2005) Uniqueness of generalized equilibrium for box-constrained problems and applications. In: *Proceedings of the Allerton*

conference on communication, control, and computing, Monticello

Simsek A, Ozdaglar A, Acemoglu D (2007) Generalized Poincare-Hopf theorem for compact nonsmooth regions. *Math Oper Res* 32(1):193–214

Simsek A, Ozdaglar A, Acemoglu D (2008) Local indices for degenerate variational inequalities. *Math Oper Res* 33(2):291–301

Stein N, Ozdaglar A, Parrilo PA (2008) Separable and low-rank continuous games. *Int J Game Theory* 37(4):475–504

Stein N, Ozdaglar A, Parrilo PA (2013) Exchangeable equilibria, part I: symmetric bimatrix games. Working paper

been successfully implemented in assembly, machining, and semiconductor manufacturing processes. More research and development are needed to extend the SoV theory to manufacturing systems with complex configurations.

Keywords

Data fusion · Engineering-driven statistics · Multistage manufacturing system · Quality improvement · Variation reduction

Stream of Variations Analysis

Jianjun Shi

Georgia Institute of Technology, Atlanta, GA, USA

Abstract

Stream of variation (SoV) theory is a unified, model-based method for modeling, analyzing, and controlling variation in multistage manufacturing systems. A SoV model represents variation and its propagation in a multistage system using the recursive structure of state space models; such models can be derived from physical knowledge and/or estimated empirically using system operational data. Immediately, the SoV model enables integrated design and optimization for product and process tolerancing, allocation of distributed sensors in production lines, and evaluation of multistage system designs. With the help of these functions, the SoV method fulfills the objectives of system monitoring, diagnosis, and control and, ultimately, reduces a system's variation during its operation. The SoV method can be further extended to model the interactions among product quality and tooling reliability, known as the quality and reliability chain effects, which is the crucial element in carrying out quality-ensured maintenance, as well as system reliability evaluation and optimization. The SoV theory has

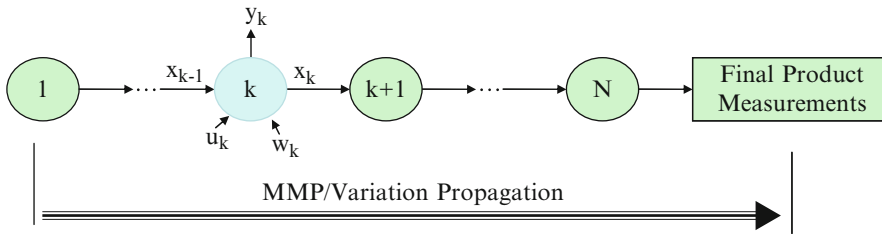
Introduction

A multistage system refers to a system consisting of multiple units, stations, or operations to finish a final product or a service. Multistage systems are ubiquitous in modern manufacturing processes and service systems. In most cases, the final product or service quality of a multistage system is determined by complex interactions among multiple stages – the quality characteristics of one stage are not only influenced by the local variations at that stage but also by the variations propagated from upstream stages. Multistage systems present significant challenges for quality engineering research and system improvement.

The stream of variation (SoV) theory has been developed to understand and represent the complex production stream and data stream involved in the modeling and analysis of variation and its propagation in a multistage manufacturing system (Fig. 1).

Stream of Variation Model

The foundation of the SoV theory is a mathematical model that links the key product quality characteristics with key process control characteristics (e.g., fixture error, machine error, etc.) in a multistage system. This model has a state space representation that describes the deviation and its propagation in an N -stage process (as shown in Fig. 1) and takes the form of



Stream of Variations Analysis, Fig. 1 Variation propagation in a multistage manufacturing process (MMP) and notations in SoV modeling (Reproduced from Shi 2006)

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k, \quad k = 1, 2, \dots, N, \quad (1)$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k, \quad \{k\} \subset \{1, 2, \dots, N\}, \quad (2)$$

where k is the stage index, $\mathbf{x}_k$ is the state vector representing the key quality characteristics of the product (or intermediate work piece) after stage k , $\mathbf{u}_k$ is the control vector representing the tooling deviations (e.g., no fault occurs if all tooling deviations are within their tolerances; fault occurs when excessive tooling deviations are beyond their tolerances; active adjustments of tooling deviations can be done to achieve error compensation objectives) at stage k , and $\mathbf{y}_k$ is the measurement vector representing product quality measurements at stage k . Vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ represent modeling error and sensing error, respectively. The coefficient matrices $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$ are determined by product and process design information: $\mathbf{A}_k$ represents the impact of the deviation transition from stage $k-1$ to stage k , $\mathbf{B}_k$ represents the impact of the local tooling deviation on the product quality at stage k , and $\mathbf{C}_k$ is the measurement matrix, which can be obtained from the defined quality features of the product at stage k .

If we repeat the modeling efforts for each stage from $k = 1$ to N , we will get the deviation and its propagation throughout the multistage manufacturing systems. By taking variances on both sides of (1) and (2) and by assuming independence among certain variables, we will obtain the variation and its propagation model for the multistage manufacturing system.

The SoV models (1) and (2) can be obtained from product and process design information and/or from the system operational data. In Shi (2006), two basic modeling methods, a physics-driven method and a data-driven method, were investigated. In the physics-driven modeling, the kinematic relationships between key control characteristics (KCC) and key product characteristics (KPC) are identified through a detailed physical analysis of the product and manufacturing process. A set of carefully defined coordinate systems are defined to represent the whole system, including the quality features in the part coordinates, part orientation to fixture/machine coordinates, and tooling to fixture/machine coordinates. Based on these coordinate systems, SoV models (1) and (2) are obtained using the state space model framework. In the data-driven modeling approach, system operational data are measured for those selected KPC and KCC variables. System identification and estimation methods are adopted to construct the SoV model. In some cases, data mining and clustering techniques are used to identify inherent relationships of the system in pre-processing. The SoV model may have different formulations, such as the state space model, input-output model, and piecewise linear regression tree model. In most cases, engineering-driven statistical analysis is commonly used in the data analysis and modeling efforts.

With models (1) and (2), variation reduction can be achieved in both design and manufacturing phases by using mathematical optimization to make optimal decisions. However, significant challenges exist in both the model development

for specific processes and model utilization to realize the benefits of the analytical capability of this model. These challenges are addressed in the SoV methodological research (Shi 2006). In more detail, the SoV methodology addresses the following important questions for variation reduction in a multistage manufacturing process.

SoV-Enabled Monitoring and Diagnosis

In multistage manufacturing systems, it is challenging to systematically find the root causes of a severe variability in terms of isolating both the manufacturing station and the underlying cause in that station. During continuous production, excessive product variation may occur at any stage of a multistage manufacturing system due to worn tooling, tooling breakage, and/or abnormal incoming part variation. The SoV theory presents systematic approaches for root cause identification. In this approach, a new concept of “statistical methods driven by engineering models” is proposed to integrate the product and process design knowledge with the on-line statistics. By solving the difference equation of models (1) and (2) and with some mathematical simplifications, the SoV model can be transformed into an input-output format as

$$\mathbf{y} = \mathbf{\Gamma} \cdot \mathbf{u} + \mathbf{\varepsilon}, \quad (3)$$

where $\mathbf{y}$ is an $n \times 1$ vector of product quality measurements, $\mathbf{\Gamma}$ is an $n \times p$ constant system matrix determined by product/process designs, $\mathbf{u}$ is a $p \times 1$ random vector representing the process faults, and $\mathbf{\varepsilon}$ is an $n \times 1$ random vector representing measurement noises, un-modeled faults, and high-order nonlinear terms. During production, the product quality features ($\mathbf{y}$) are measured, and the data are used to conduct statistical analysis based on the model (1) to identify root causes. Two basic methods are developed for root cause diagnosis: (i) variation pattern matching: In this method, all potential variation patterns can be obtained from the matrix $\mathbf{\Gamma}$ resulting from the off-line system design.

During the system operation, observed variation patterns can be obtained from the covariance matrix of $\mathbf{y}$. A pattern matching can be performed to identify the root causes. (ii) estimation-based diagnosis: With the SoV model and availability of on-line measurement of quality feature ($\mathbf{y}$), the deviation value of $\mathbf{u}$ can be estimated on-line. A hypothesis testing of $\mathbf{u}$ and its variance reveals the significant changes that occurred to $\mathbf{u}$, corresponding to the root causes of the system. Various estimators and their performances are evaluated in the diagnosis study (Chapter 11 of Shi 2006).

SoV-Enabled Sensor Allocation and Diagnosability

The issue of diagnosability refers to the problem of whether the product measurements contain sufficient information for the diagnosis of critical process faults, i.e., if root causes of process faults can be diagnosed. The diagnosability analysis is investigated based on model (3) that links potential process faults ($\mathbf{u}$) and product quality measurements ($\mathbf{y}$). In the SoV theory, a set of criteria is developed to evaluate the *mean* diagnosability and *variance* diagnosability for a system. Similar to observability in control theory, diagnosability is determined by the $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$ matrices ($k = 1, \dots, N$) in the SoV models (1) and (2) (or the $\mathbf{\Gamma}$ matrix in model (3)). In some cases, only a subset of variables (vs. specific root cause variables) can be identified as potential root causes of the process faults, which are referred to as minimum diagnosable classes.

One emphasis in the SoV-enabled diagnosability study is to promote the concept of the “process-oriented measurement” strategy. In current industrial practice, most of the existing measurement strategies focus on the product coherence inspection (i.e., product-oriented measurements), which is effective for detecting product imperfection, but may not be effective to identify the root causes of product quality failures. The SoV theory proposes a “process-oriented measurement” concept with a distributed sensing strategy. In this strategy, selected key control characteristics, as well as selected key

product characteristics, will be measured in the selected stages for both detecting product defects and identifying their root causes.

SoV-Enabled Design and Optimization

Variation analysis and design evaluations are conducted in the product and process design stage to identify critical components, features, and manufacturing operations. With the SoV model defined in (3) and certain assumptions, we can represent the KPC-to-KCC relationship as

$$\tilde{\Sigma}_y = \sum_{k=1}^N \Gamma_k \Sigma_{u_k} \Gamma_k^T, \quad (4)$$

where $\tilde{\Sigma}_y$ is the variance-covariance matrix of product quality features resulting from the variance-covariance matrix (Σ_{u_k}) of tooling errors. Based on (3) and (4), the following four tasks can be performed: (i) tolerance analysis by allocating the tooling tolerance ($\mathbf{u}_k$) and then predicting the final product tolerance ($\mathbf{y}_N$); (ii) tolerance synthesis by fixing the final product tolerance ($\mathbf{y}_N$) and then assigning the tolerance for individual tooling components ($\mathbf{u}_k$) with certain cost objectives minimized; (iii) sensitivity study by identifying the critical tooling components ($\mathbf{u}_k$) that have significant impacts on the final production variation through evaluation of the defined sensitivity indices; and (iv) process planning by optimizing parameters in $\mathbf{A}_k$ and $\mathbf{B}_k$ matrices to minimize the final product variation.

One unique feature of SoV-enabled design and optimization is to provide a unified method for simultaneous optimization of product and process tooling tolerance, as well as process planning. This is because the SoV models (1) and (2) represent the product quality features ($\mathbf{x}_k$ and $\mathbf{y}_k$), tooling features ($\mathbf{u}_k$), and the process planning formation ($\mathbf{A}_k$ and $\mathbf{B}_k$) within one mathematical model. As a result, a math-based optimization is feasible to achieve the best quality through process-oriented tolerance synthesis for product and process, as well as optimized process planning.

SoV-Enabled Process Control and Quality Compensation

The SoV model provides the opportunity to apply active control for dimensional variation reduction in a multistage manufacturing system. The basic idea is to implement a system-level control strategy during production to minimize the end-of-line product variance, which is propagated from upstream manufacturing stages. An optimal control scheme was devised to use the state space structure of the SoV model by treating the control as a stochastic discrete-time predictive control problem. The optimization index for determining the optimal control action is formulated as

$$\begin{aligned} J_k^* &= \min_{\mathbf{u}_k} J_k = \min_{\mathbf{u}_k} E \left[\hat{\mathbf{y}}_{N|k}^T \mathbf{Q}_N \hat{\mathbf{y}}_{N|k} + \mathbf{u}_k^T \mathbf{R}_k \mathbf{u}_k \right], \\ \text{s.t. } C_{k,c}^L &\leq u_{k,c} \leq C_{k,c}^U, \quad k = 1, \dots, N, \quad c = 1, \dots, n_{u,k}. \end{aligned} \quad (5)$$

where $\hat{\mathbf{y}}_{N|k}$ denotes the product quality at the final stage N that is predicted at stage k and $n_{u,k}$ is the dimension of the control action $\mathbf{u}_k$. The constraints $[C_{k,c}^L, C_{k,c}^U]$ define the upper and lower actuator limits that can be applied on each part/substage. $\mathbf{Q}_N \in \mathbf{R}^{m \times m}$ is a positive semi-definite matrix, and $\mathbf{R}_k \in \mathbf{R}^{n \times n}$ is a positive definite matrix.

This optimization index takes the form of the widely accepted cost function of a linear-quadratic regulator under the predictive control framework and thus satisfies the common requirements in control theory. Various research topics have been investigated under this framework, including the feed-forward control for multistage process, cautious control considering model uncertainties, and actuator layout optimization in control system designs.

SoV-Enabled Product Quality and Reliability Chain Modeling and Analysis

There is a complex, intricate relationship between product quality and tooling reliability in a multistage manufacturing system. A degraded (or

failed) production tool leads to a large variability in product quality and/or an excessive number of defects; on the other hand, excessive variability of product quality features accelerates the degradation and failure rates of production tooling at the station thereafter. For a multistage manufacturing system, these interactions are more complex as variations propagate from one stage to the next stage. Thus, a “chain effect” between the product quality (Q) and tooling reliability (R) can be observed and thus noted as the “QR chain” effect. Modeling of the QR chain is an integrated effort of the SoV model and the semi-Markov process model. The QR chain model plays an essential role in system reliability modeling and maintenance decisions and has led to new concepts of quality-ensured maintenance strategy, and tolerance synthesis considering tool degradation and system down time.

Summary and Future Directions

The concept of stream of variation for multistage systems can be applied to a very broad range of systems, although the existing work mostly focuses on the quality control of multistage discrete manufacturing processes. A comprehensive discussion on the stream of variation theory for a multistage manufacturing system is summarized in a monograph (Shi 2006). In addition, Shi and Zhou (2009) provides a survey of emerging methodologies for tackling various issues in multistage systems including modeling, analysis, monitoring, diagnosis, control, inspection, and design optimization.

The success of the multistage system framework in manufacturing processes will certainly stimulate the application of this framework to other systems. For example, monitoring and diagnosis of the abnormalities in throughput, cycle time, and lead time of a multistage production system are very promising application areas under the multistage system framework. The supply chain and logistics management,

which involve multiple suppliers/vendors in an interconnected fashion, can be treated as another multistage system with network structures. Most service systems such as health-care clinics, hospitals, and transportation systems are inherently multistage as well. It will be interesting to expand the stream of variation theory to these broadly defined multistage systems for their quality control, variation reduction, and other system-level performance improvement.

Cross-References

- [Fault Detection and Diagnosis](#)
- [Multiscale Multivariate Statistical Process Control](#)
- [Statistical Process Control in Manufacturing](#)

Recommended Reading

The monograph (Shi 2006) provides detailed results of the stream of variation theory discussed in this entry. In addition, the first five chapters of Shi (2006) provide views of basic statistical and system analysis tools needed for the SoV research and development. Some recent developments related to the SoV theory and applications are summarized in a review paper (Shi and Zhou 2009).

Bibliography

- Shi J (2006) Stream of variation modeling and analysis for multistage manufacturing processes. CRC, Boca Raton, 469pp. ISBN:0-8493-2151-4
- Shi J, Zhou S (2009) Quality control and improvement for multistage systems: a survey. IIE Trans Qual Reliab Eng 41:744–753

Structural Properties of Biological and Ecological Systems

Franco Blanchini¹, Elisa Franco², and Giulia Giordano³

¹University of Udine, Udine, Italy

²Mechanical and Aerospace Engineering, University of California, Los Angeles, CA, USA

³Department of Industrial Engineering, University of Trento, Trento, TN, Italy

Abstract

It is astounding how systems in nature can survive under completely different environmental conditions and in the presence of huge parameter variations. Structural analysis aims at explaining why this is possible by studying properties of biological models that hold regardless of parameter values. Here, we discuss selected system properties that have been successfully investigated and explained just looking at the structure, without the need of quantitative information.

Keywords

Dynamical systems · Structural properties · Systems biology

Introduction

Mathematical models of biological and ecological systems are important to help predict their dynamics, as well as to formulate and support hypotheses explaining their behavior (Alon 2006; Edelstein-Keshet 2005). Deterministic models are often employed when the species being modeled are abundant and do not experience stochastic fluctuations (cf. “► [Deterministic Description of Biochemical Networks](#)” by J. Stelling and H.-M. Kalthenbach and “► [Stochastic Description of Biochemical Networks](#)” by J. P. Hespanha

and M. Khammash). Yet, deterministic models still include uncertain parameters that reflect the complexity of biological environments. Tools from control and dynamical systems theory allow us to draw conclusions on their properties and dynamic behaviors (Cosentino and Bates 2011; Del Vecchio and Murray 2014; Sontag 2005), also in the presence of uncertainty.

We call *structural* a property that is *independent* of the particular parameter values (Blanchini and Franco 2011; Shinar and Feinberg 2010). We consider properties that can be mathematically defined, such as the number of equilibria and their stability, and we survey methods to structurally assess these properties for biological and ecological systems. The qualitative (parameter-free) methods discussed in this entry complement quantitative (numerical) approaches to establish parametric robustness, described in “► [Robustness Analysis of Biological Models](#)” by S. Waldherr and F. Allgöwer.

Systems and Structures

We consider the ordinary differential equation model:

$$\dot{x}(t) = f(x(t), \mu), \quad (1)$$

where $x \in \mathbb{R}_+^n$ is a vector of species concentrations or population densities, $f : \mathbb{R}_+^n \times \mathbb{R}_+^q \rightarrow \mathbb{R}^n$, and $\mu \in \mathbb{R}_+^q$ is a vector of parameters. Both f and μ may be uncertain or even unknown and time-varying. For instance, protein degradation may be temperature dependent and may be modeled with a linear or saturating function (Sontag 2005); population birth rates depend on variable environmental factors and may be modeled as exponential or logistic growth (May 1974). It is often reasonable to assume that the components of f ($f_i, i = 1, \dots, n$) are *monotonic* functions of the x_j s, i.e., each $\partial f_i(x, \mu) / \partial x_j$ is sign-definite (either always nonnegative or non-positive) in the domain of interest; under this assumption, the Jacobian matrix J of the system is also sign-definite and can be mapped to a sign matrix Σ

whose element (i, j) is the sign of $\partial f_i(x, \mu)/\partial x_j$. Then, this *sign* matrix Σ captures the *structure* of the system, which remains fixed despite uncertainty or variability in the model.

For (bio)chemical reaction networks (Angeli 2009; Del Vecchio and Murray 2014), the model in (1) can be rewritten as:

$$\dot{x}(t) = Sg(x(t), \mu) + g_0(\mu), \quad (2)$$

where $S \in \mathbb{Z}^{n \times m}$ is a stoichiometry matrix of signed integer entries, $g : \mathbb{R}_+^n \times \mathbb{R}_+^q \rightarrow \mathbb{R}_+^m$ is a vector of reaction rate functions, and $g_0 : \mathbb{R}_+^q \rightarrow \mathbb{R}_+^n$ models external inputs. When the *law of mass action* applies, the reaction rates can be modeled as polynomial functions of the species concentrations. More in general, the components of g ($g_i, i = 1, \dots, m$) are nonnegative functions, *monotonic* in the x_j s; they depend on the system state and usually include uncertain or fluctuating parameters. Conversely, matrix S is constant, independent of both states and parameters μ , and represents the *interconnection structure* of the various reactions. System (1) is a particular case of (2) with $S = I$ (the identity), $g = f$, and $g_0 = 0$. The Jacobian of a system of the form (2) is not sign-definite in general, but it always admits the *BDC* decomposition (Blanchini and Giordano 2014; Giordano et al. 2016):

$$J(x) = BD(x)C, \quad (3)$$

where $D(x)$ is a diagonal matrix carrying on the diagonal the absolute value of the partial derivatives of the vector function g , while B and C are constant matrices that capture the system *structure*.

Matrix Σ , matrix S , and the matrix pair B, C are examples of *structures* of a biological model, which are not affected by parameter uncertainty or variability. If a property holds due to the particular structure of the model, regardless of the value of the parameters μ , then it is a *structural property*.

Positivity and Boundedness

Any biological or ecological model should be a positive system, because negative concentrations or population densities are not physically acceptable. If a biological model does not *structurally* satisfy the positivity property, it is not well-posed and should be revised.

System (1) is positive if assuming that $x(0) \geq 0$ (componentwise), then $x(t) \geq 0$ for all $t \geq 0$.

A system of the form

$$\dot{x}_i = f_i(x_1, x_2, \dots, x_n, \mu), \quad i = 1, \dots, n$$

is positive if and only if, for all k , when $x_k = 0$, its derivative $\dot{x}_k$ is nonnegative:

$$f_k(x_1, x_2, \dots, x_{k-1}, \underbrace{0}_{x_k}, x_{k+1}, \dots, x_n, \mu) \geq 0.$$

If this condition holds for arbitrary values of μ , then the system is structurally positive. For a *linear* autonomous system $\dot{x} = Ax$, positivity is equivalent to the fact that A is a Metzler matrix ($A_{ij} \geq 0$ for $i \neq j$).

Boundedness is another important property often (although not always) structurally verified by biological and ecological models. The concentrations of biomolecular species such as proteins and mRNA may fluctuate, but remain bounded in a living cell due to the presence of degradation or secretion mechanisms; similarly, biological species such as predators and preys may experience wide fluctuations, but they rarely exceed thresholds related to the carrying capacity of their ecosystem. In a model of the form (1), we say that the solutions are bounded if, for any initial state $x(0) \geq 0$, there exists a positive constant ξ such that $0 \leq x_i(t) \leq \xi$ for all $t > 0$ and for all i .

The concept of boundedness is directly related to the existence of positively invariant sets; a (bounded) subset $\mathcal{P}$ of the state space is said to be invariant if $x(0) \in \mathcal{P}$ implies that $x(t) \in \mathcal{P}$ for all $t > 0$. Nagumo's theorem provides general necessary and sufficient conditions for a set to be invariant, conditions that may be verified structurally (Blanchini and Miani 2015). The applicability of Nagumo's theorem is very

general. Yet, for significant classes of systems, structural boundedness can be assessed by direct inspection. For example, consider the model:

$$\dot{x} = -\Lambda x + Sg(x) + g_0,$$

where Λ is a positive diagonal matrix, modeling the presence of self-degradation for all species, and all the mutual interactions expressed by the entries of g are assumed to be bounded. This system has the form (2), with extended matrix $\tilde{S} = [-\Lambda \ S]$ and extended vector $\tilde{g}(x) = [x \ g(x)]^\top$. If g is a vector of bounded functions, $|[Sg(x) + g_0]_i| \leq \nu$, then

$$\dot{x}_i = [Sg(x) + g_0]_i - \lambda_i x_i \leq \nu - \lambda_i x_i.$$

Hence for all i , if $x_i \geq \xi = \max\{\nu/\lambda_i\}$, we have $\dot{x}_i < 0$, and the variable cannot grow above ξ . The existence of such a bound is a structural property, even though its value depends on the parameters.

Boundedness of the solutions implies the existence of at least an equilibrium, a topic discussed next.

Analysis of the Equilibria

The vector $\bar{x}(\mu) \geq 0$ is an equilibrium for (1) if

$$0 = f(\bar{x}(\mu), \mu)$$

or an equilibrium for (2) if $0 = Sg(\bar{x}(\mu), \mu) + g_0(\mu)$. The equilibria of nonlinear systems like (1) and (2) are generally found via numerical methods. Their stability can be analyzed via standard methods such as Routh-Hurwitz criterion or eigenvalue computation for the linearized system. Unfortunately, in general this requires the knowledge of the parameter values.

Tools from *topological degree theory* (Lloyd 1978) can help structurally establish existence, uniqueness, and, in some cases, stability of equilibria. If the solution of (1) is bounded in a convex set $\mathcal{P}$ that is positively invariant and has a non-empty interior, then at least one equilibrium $\bar{x}$ exists. To analyze how many, assume

that no equilibrium is on the boundary of $\mathcal{P}$ and that all equilibria $\bar{x}_k$ are regular, namely, $\det[J(\bar{x}_k)] \neq 0$ for all k . Then it must hold that $\sum_k \text{sign}\{\det[-J(\bar{x}_k)]\} = 1$; see Lloyd (1978). Hence, there cannot be an even number of regular equilibria in the interior of $\mathcal{P}$. If there is an odd number of equilibria, they must obey a rule: for instance, if three regular equilibria exist, exactly one must be such that $\det[-J(\bar{x}_k)] < 0$, hence it must be unstable (because $\det[-J(\bar{x}_k)]$ is the constant term of the corresponding characteristic polynomial and a characteristic polynomial with all positive coefficients is necessary for stability). If $\det[-J(\bar{x})] > 0$ structurally, then the equilibrium must be unique.

Stability of the Equilibria

Analyzing stability of the equilibria is challenging if the parameter values are unknown. However, for important classes of systems, powerful structural results are available. For instance, compartmental systems (Jacquez and Simon 1993), which can be seen as networks of unimolecular chemical reactions $X_i \rightarrow X_j$, have a remarkable intrinsic stability property (Maeda et al. 1978).

Interesting results come from the theory of chemical reactions (Clarke 1980; Feinberg 1987). For example, the structural local stability of an equilibrium can be cast as a D -stability problem (Clarke 1980); a matrix A is D -stable if A is Hurwitz and AD is Hurwitz for any arbitrary diagonal matrix D with positive elements. Also, the famous *deficiency zero theorem* (Feinberg 1987) exploits the algebraic properties of matrix S to provide structural results. The stoichiometric matrix can be decomposed as $S = NM$, where N captures the molarity of each species x_i in each complex (group of species appearing on either side of the reactions whose rates are in vector g), while M is the complex-reaction incidence matrix; then, the deficiency of the network is the structural quantity $\delta = \dim\{\ker(N) \cap \text{range}(M)\}$, which depends only on S and not on the parameters μ . If the network is weakly reversible (if there is a directed path from complex i to complex j , then a path from j to i is also present) and all components of g are of the mass action type (namely, for a chemical reaction

$p_1X_1 + p_2X_2 + \dots p_rX_r \rightarrow q_1Y_1 + q_2Y_2 + \dots q_sY_s$, the rate has the form $g_j = \mu_j x_1^{p_1} x_2^{p_2} \dots x_r^{p_r}$, the deficiency zero theorem states that, if $\delta = 0$, then there exists within each positive stoichiometric compatibility class (associated with the initial conditions) a single equilibrium, which is locally asymptotically stable. Remarkably, the theorem is proven by showing that the system entropy is a Lyapunov function.

Structural stability of biochemical networks can be assessed via quadratic Lyapunov functions (even parameter dependent) (Clarke 1980) and, also for *general* reaction rates that are not mass action, via non-quadratic Lyapunov functions (Al-Radhawi and Angeli 2016; Blanchini and Franco 2011; Blanchini and Giordano 2014). *Structural* piecewise-linear Lyapunov functions can be systematically built for a general system (2) based on its *BDC* decomposition (Blanchini and Giordano 2014).

Perturbation of the Equilibrium

A biological system can be subject to unknown or uncertain inputs/perturbations, whose structural effect can be analyzed as follows. Suppose that the system is at some (unknown) stable steady-state $\bar{x}$, corresponding to some nominal (unknown) value $\bar{\mu}$ of the parameters, and consider a relevant system output $y = Hx$, where H is a row vector. Assume that one of the parameters suddenly becomes $\bar{\mu}_j + u$, with u positive (without loss of generality) and not too large (not to compromise stability). After a transient, the system settles at a new equilibrium $\bar{x} + z(\infty)$ and a new output $\bar{y} + v(\infty)$. We have a *structural influence* if the sign of the steady-state output variation, $v(\infty)$, induced by the perturbation does not depend on the parameters $\bar{\mu}$ (Dambacher et al. 2002). Then,

$$\text{sign}[v(\infty)] \in \{+, -, 0, ?\},$$

corresponding to the cases when the steady-state output variation is always positive, always negative, always zero, or indeterminate (i.e., it depends on the parameters). For systems of the form (2), structural influences can be assessed

based on the *BDC* decomposition (3). We can investigate the *structural steady-state influence* of a constant input variation $u > 0$ (which is the parameter we are perturbing) on the output variation v by analyzing the steady-state output of the system:

$$\dot{z} = BDCz + Eu, \quad v = Hz, \quad D \text{ positive diagonal,}$$

where $z = x - \bar{x}$. The structural input-output influence of u on v can be assessed based on the sign of

$$\phi(D) = \det \begin{bmatrix} -BDC & -E \\ H & 0 \end{bmatrix} = \frac{v(\infty)}{u}.$$

As shown in Giordano et al. (2016), function $\phi(D_1, D_2, \dots, D_m)$ is positive (negative) for all $D_k > 0$ if and only if $\phi(1, 1, \dots, 1) > 0$ (< 0) and $\phi(\hat{D}_1, \hat{D}_2, \dots, \hat{D}_m) \geq 0$ (≤ 0) for all possible choices of $\hat{D}_k \in \{0, 1\}$, while it is zero if and only if it is zero for all possible choices of $\hat{D}_k \in \{0, 1\}$. It is undetermined otherwise.

When both E and H have a single nonzero element, say $E_j = 1$, $H_i = 1$, we can investigate the steady-state influence on variable i due to an input affecting the equation of variable j . The structural influence matrix collects all possible (i, j) pairs (Giordano et al. 2016). A zero steady-state influence is associated with perfect adaptation, a remarkable feature of some biological systems that transiently respond to a persistent input change and eventually return to the original equilibrium (Alon 2006; Briat et al. 2016; Steuer et al. 2011).

Structural Feedback Loops

We use simple examples to illustrate the effects of structural feedback loops, and of their sign, in systems of the form (1). Feedback, present in most biological dynamical systems, enables all homeostatic processes, as well as complex dynamic behaviors such as bistability (or multi-stability) and oscillations. As conjectured in Thomas (1981), positive feedback is a necessary condition for bistability, while negative feedback is a necessary condition for oscillations.

These conjectures were proved in Gouze (1998) and Snoussi (1998). We consider a two-node gene network:

$$\tau_a \dot{a} = -a + f_b(b), \quad \tau_b \dot{b} = -b + f_a(a) + u,$$

where a and b are concentrations of molecular species and u is an external signal. Functions $f_b(b)$ and $f_a(a)$ model the production rate of species a and b and are monotonically increasing Hill-type functions:

$$f(x) = \alpha \frac{x^p}{\beta + x^p} + \gamma,$$

where α is the maximal production rate, β is a threshold, p measures the sharpness of the sigmoid, and γ represents a basal production rate; for $x \gg \beta$, $f(x)$ saturates reaching the value

$\gamma + \alpha$. The factors τ_a and τ_b are introduced to rescale time so that the degradation rate of a and b is normalized (unitary). Species a and b mutually activate, generating a positive-feedback loop.

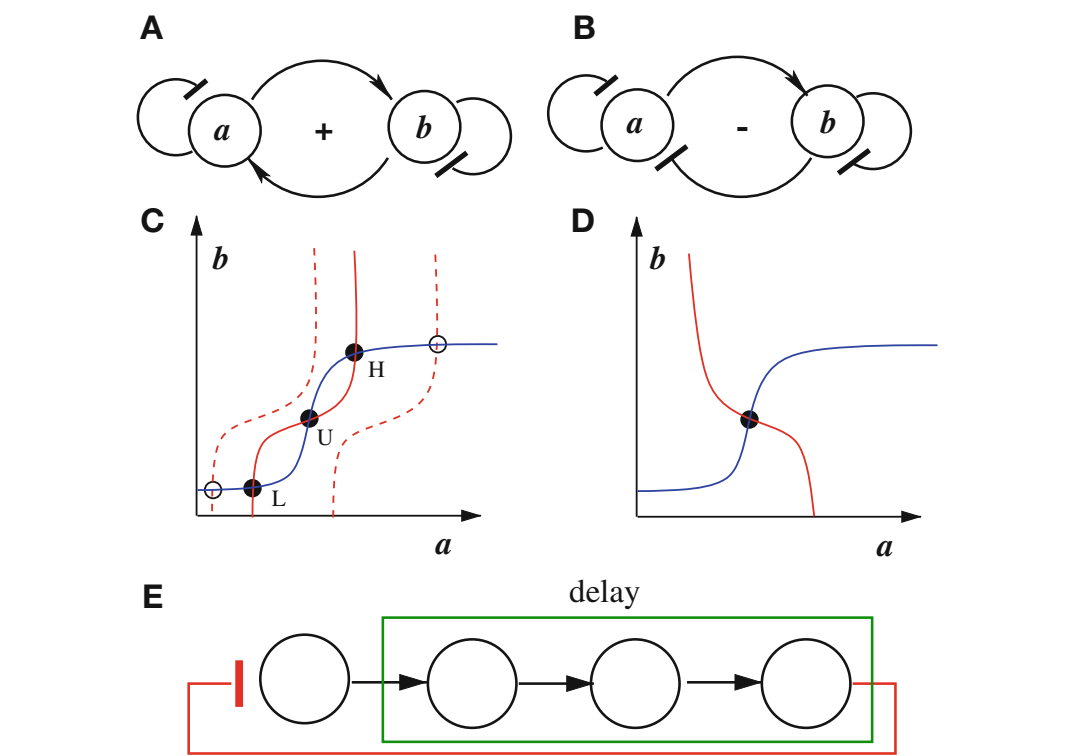
A negative-feedback loop could be generated by modifying the effect of b on a (without loss of generality):

$$\tau_a \dot{a} = -a + g_b(b), \quad \tau_b \dot{b} = -b + f_a(a) + u,$$

where $f_a(a)$ is defined as above (activator), while $g_b(b)$ is a decreasing Hill-type function (inhibition):

$$g(x) = \delta \frac{1}{\phi + x^p} + \epsilon,$$

with similar definitions of the parameters. Both systems are positive and evolve in bounded sets: this may be verified as suggested in section “Positivity and Boundedness”. The feedback loops



Structural Properties of Biological and Ecological Systems, Fig. 1 Graph representation of a two-node positive-feedback loop (A) and negative-feedback loop (B) system; pointed arrows represent positive influences, and hammerhead arrows represent negative influences.

Nullclines and equilibria for the positive-feedback loop (C) and the negative-feedback loop (D) system. A negative-feedback loop including at least three nodes, or a time delay (E), is necessary to have oscillations

generated by these systems are clearly identifiable when inspecting the Jacobian matrices, which are, respectively:

$$J_P = \begin{bmatrix} -1/\tau_a & f'_b(b)/\tau_a \\ f'_a(a)/\tau_b & -1/\tau_b \end{bmatrix},$$

$$J_N = \begin{bmatrix} -1/\tau_a & g'_b(b)/\tau_a \\ f'_a(a)/\tau_b & -1/\tau_b \end{bmatrix}.$$

Due to the monotonicity of Hill functions, since f_a and f_b are increasing while g_b is decreasing, the Jacobians can be mapped to the sign matrices:

$$\Sigma_P = \begin{bmatrix} - & + \\ + & - \end{bmatrix}, \quad \Sigma_N = \begin{bmatrix} - & - \\ + & - \end{bmatrix}.$$

These matrices can be associated with the graphs shown in Fig. 1A, B, whose nodes represent the species and whose arcs represent the signed influence of each species on the other (pointed arrows indicate positive influence; hammerhead arrows indicate negative influence) corresponding to the nonzero Jacobian entries. The feedback loops in these systems are structural, i.e., they are present for arbitrary choices of the parameters.

The sign matrix Σ_P has only positive loops; therefore, the corresponding system is a *strong candidate multistationary system* (Blanchini et al. 2014; Mincheva and Craciun 2008): if, by perturbing a parameter, we destabilize an equilibrium, then instability is generated by a dominant real eigenvalue that becomes positive (any transition to instability is exponential). In fact, instability of an equilibrium for this system implies $\det(-J_P) < 0$ in that equilibrium; however, according to the degree theory arguments in section “[Analysis of the Equilibria](#)”, this implies the existence of (at least) other two equilibria with $\det(-J_P) > 0$.

Conversely, matrix J_N only has negative loops, so the corresponding system is a *strong candidate oscillator* (Blanchini et al. 2014; Mincheva and Craciun 2008): if, by perturbing a parameter, we destabilize an equilibrium, then instability is generated by a pair of dominant complex eigenvalues whose real part

becomes positive (any transition to instability is oscillatory). Also, the system admits a single equilibrium because $\det(-J_N) > 0$ structurally. In this special (2×2) case, the unique equilibrium is always asymptotically stable, as we will find out soon.

Indeed, a strong candidate oscillator (multistationary system) does not necessarily oscillate (exhibit multiple equilibria); however, the onset of *instability* (if any) is necessarily associated with sustained oscillations (appearance of more equilibria).

In addition to the above information about the admissible dynamics, obtained through degree theory arguments, the low order of the two considered systems enables a more direct and specific qualitative analysis of the equilibria and their stability.

The equilibrium conditions for the positive-feedback system are:

$$-a + f_b(b) = 0, \quad -b + f_a(a) + u = 0.$$

Their solutions correspond to the intersections of the two curves illustrated in Fig. 1C. Depending on the value of u , we may have either one or three intersections (we consider tangentiality as two coincident solutions). For small u , there is a single equilibrium (L) corresponding to low levels of both variables. For large u , there is a single equilibrium (H) corresponding to high levels of both variables. For intermediate values of u , we may have three equilibria, and the intermediate one (U) is unstable. This configuration yields a bistable device that can act as a “switch”: from the equilibrium U, if we increase u , we move the system to the upper equilibrium H, while, if we decrease u , we move the system to the lower equilibrium L. The situation remains unchanged if we restore u to the nominal bistable level. In the negative-feedback system, similar reasonings show the presence of a single equilibrium, as shown in Fig. 1D, regardless of the parameters.

To investigate the stability of the equilibria, we first find the characteristic polynomial associated with J_P :

$$\Phi_P(\lambda) = \lambda^2 + (1/\tau_a + 1/\tau_b)\lambda$$

$$+ (1 - f'_a f'_b)/(\tau_a \tau_b).$$

It can be checked that:

- (a) at the equilibria L and H, $(1 - f'_a f'_b)/(\tau_a \tau_b) > 0$; hence, the roots of $\Phi_P(\lambda)$ (the eigenvalues of J_P) have negative real part, since $(1/\tau_a + 1/\tau_b) > 0$;
- (b) at the equilibrium U, $(1 - f'_a f'_b)/(\tau_a \tau_b) < 0$; hence, there exists a positive real root.

Similarly, the characteristic polynomial of J_N is:

$$\begin{aligned}\Phi_N(\lambda) = & \lambda^2 + (1/\tau_a + 1/\tau_b)\lambda \\ & + (1 - g'_b f'_a)/(\tau_a \tau_b).\end{aligned}$$

Since $g'_b f'_a < 0$, all the coefficients of the polynomial are positive; therefore, all the eigenvalues have negative real part, and the unique equilibrium is asymptotically stable for any value of the parameters.

We may be tempted to conclude that negative feedback stabilizes the system; however, this is not a generalizable conclusion. Let us modify this model to include the additional species c :

$$\begin{aligned}\tau_a \dot{a} &= -a + g_c(c), \\ \tau_b \dot{b} &= -b + f_a(a), \\ \tau_c \dot{c} &= -c + f_b(b) + u.\end{aligned}$$

The eigenvalues of the Jacobian associated with this system are the roots of the equation

$$(\tau_a \lambda + 1)(\tau_b \lambda + 1)(\tau_c \lambda + 1) + \kappa = 0,$$

where $\kappa = -f'_a f'_b g'_c > 0$. These roots cannot be real positive, consistent with the fact that we have a negative loop only. Standard root locus or Routh-Hurwitz analysis shows that, if κ is sufficiently large, two roots move to the right half of the complex plane, so a necessary requirement to have oscillations is that the loop includes at least three first-order sub-systems in series or a single “delay” element whose effect is comparable to that of a longer chain (see Fig. 1E). Smaller

values of κ are sufficient to destabilize the system when the time constants τ_i are similar, while the most favorable situation to ensure stability is when one of these constants, say τ_1 , is much larger than the others (Blanchini et al. 2018b).

Aggregation Can Simplify the Structural Analysis of Complex Networks

The methods and examples we discussed so far are easy to apply to small systems. Yet, comprehensive models of biological processes involve many species, making it difficult to use analytical methods. A route toward the simplification of large systems is “aggregation.” Consider a system of the form

$$\dot{x} = f(x) + Fu, \quad y = Gx \quad (4)$$

where, for simplicity, F and G are column and row vectors, while u and y are scalars. A fundamental property that can be exploited is *monotonicity* (see also “► Monotone Systems in Biology” by D. Angeli). If system (4) is input-output monotone (Angeli and Sontag 2003; Hirsch and Smith 2005), it is order-preserving, i.e., given two initial states, $x^a(0)$ and $x^b(0)$, such that $x^a(0) \geq x^b(0)$ componentwise, and two inputs $u^a(t) \geq u^b(t)$, the two corresponding solutions $x^a(t)$ and $x^b(t)$ satisfy $x^a(t) \geq x^b(t)$ (componentwise) and the outputs satisfy $y^a(t) \geq y^b(t)$. Input-output monotonicity is guaranteed if F and G are positive vectors and the Jacobian J of $f(x)$ is a Metzler matrix (namely, $J_{ij} \geq 0$ for $i \neq j$). If we assume stability, the order-preserving property of a monotone system makes its behavior qualitatively comparable to that of a first-order system; therefore, a high-order model can be collapsed to a single node (see Blanchini et al. (2018a) and the references therein). Aggregation of several nodes associated with monotone sub-systems allows us to drastically reduce the complexity of a large network and then apply the structural analysis tools described earlier to the reduced system.

Discussion and Future Directions

Structural analysis reveals parameter-free properties of natural systems and explains how they can keep performing their life-preserving function in the most different environmental conditions. Many challenges related to structural analysis are still open, such as the full characterization of structurally stable chemical reaction networks with arbitrary kinetics. Structural perturbation analysis could go beyond constant inputs to include, e.g., periodic input signals and consider not only steady-state effects but also the short-term system response.

Aggregating sub-systems that enjoy particular properties and can be collapsed into a single node enables the structural analysis of large-scale complex systems through model simplification; to this aim, properties of interest besides monotonicity could be exploited, such as the positive-impulse-response property (Blanchini et al. 2018a).

Structural analysis not only can establish that a property holds for a given system even under huge parametric uncertainties, but it can also reveal that a property is necessarily ensured by a certain *structure*. This allows us not only to explain observed phenomena but also to falsify models by comparing structural predictions to experimental results. Finally, structural analysis can be precious to design de novo artificial biomolecular circuits (see ► “[Synthetic Biology](#)” by D. Del Vecchio and R. M. Murray) that are guaranteed to exhibit the desired qualitative function, despite perturbations, in view of their structure.

Cross-References

- [Deterministic Description of Biochemical Networks](#)
- [Monotone Systems in Biology](#)
- [Robustness Analysis of Biological Models](#)
- [Stochastic Description of Biochemical Networks](#)
- [Synthetic Biology](#)

Bibliography

- Alon U (2006) An introduction to systems biology: design principles of biological circuits. Chapman & Hall/CRC, Boca Raton
- Al-Radhawi MA, Angeli D (2016) New approach to the stability of chemical reaction networks: piecewise linear in rates Lyapunov functions. *IEEE Trans Autom Control* 61(1):76–89
- Angeli D (2009) A tutorial on chemical reaction network dynamics. *Eur J Control* 15(3–4):398–406
- Angeli D, Sontag ED (2003) Monotone control systems. *IEEE Trans Autom Control* 48(10):1684–1698
- Blanchini F, Franco E (2011) Structurally robust biological networks. *BMC Syst Biol* 5(1):74
- Blanchini F, Giordano G (2014) Piecewise-linear Lyapunov functions for structural stability of biochemical networks. *Automatica* 50(10):2482–2493
- Blanchini F, Miani S (2015) Set-theoretic methods in control, 2nd edn. Systems & control: foundations & applications. Birkhäuser, Basel
- Blanchini F, Franco E, Giordano G (2014) A structural classification of candidate oscillatory and multistationary biochemical systems. *Bull Math Biol* 76(10):2542–2569
- Blanchini F, Cuba Samaniego C, Franco E, Giordano G (2018a) Aggregates of monotonic step response systems: a structural classification. *IEEE Control Netw Syst* 5(2):782–792
- Blanchini F, Cuba Samaniego C, Franco E, Giordano G (2018b) Homogeneous time constants promote oscillations in negative feedback loops. *ACS Synth Biol* 7(6):1481–1487
- Briat C, Gupta A, Khammash M (2016) Antithetic integral feedback ensures robust perfect adaptation in noisy biomolecular networks. *Cell Syst* 2(1):15–26
- Clarke BL (1980) Stability of complex reaction networks. In: Prigogine I, Rice SA (eds) *Advances in chemical physics*. Wiley, New York
- Cosentino C, Bates D (2011) *Feedback control in systems biology*. Taylor & Francis, Boca Raton/London
- Dambacher J, Li H, Rossignol P (2002) Relevance of community structure in assessing indeterminacy of ecological predictions. *Ecology* 83(5):1372–1385
- Del Vecchio D, Murray RM (2014) *Biomolecular feedback systems*. Princeton University Press, Princeton
- Edelstein-Keshet L (2005) *Mathematical models in biology*. Volume 46 of classics in applied mathematics. SIAM, Philadelphia
- Feinberg M (1987) Chemical reaction network structure and the stability of complex isothermal reactors I. The deficiency zero and deficiency one theorems. *Chem Eng Sci* 42:2229–2268
- Giordano G, Cuba Samaniego C, Franco E, Blanchini F (2016) Computing the structural influence matrix for biological systems. *J Math Biol* 72(7):1927–1958
- Gouze JL (1998) Positive and negative circuits in dynamical systems. *J Biol Syst* 6:11–15

- Hirsch MW, Smith H (2005) Monotone dynamical systems. In: Canada A, Drabek P, Fonda A (eds) Handbook of differential equations: ordinary differential equations, vol 2, Elsevier, pp 239–357
- Jacquez JA, Simon CP (1993) Qualitative theory of compartmental systems. SIAM Rev 35(1):43–79
- Lloyd NG (1978) Degree theory. Cambridge University Press, London
- Maeda H, Kodama S, Ohta Y (1978) Asymptotic behavior of nonlinear compartmental systems: nonoscillation and stability. IEEE Trans Circuits Syst 25(6):372–378
- May RM (1974) Stability and complexity in model ecosystems. Princeton University Press, Princeton
- Mincheva M, Craciun G (2008) Multigraph conditions for multistability, oscillations and pattern formation in biochemical reaction networks. Proc IEEE 96(8):1281–1291
- Shinar G, Feinberg M (2010) Structural sources of robustness in biochemical reaction networks. Science 327(5971):1389–1391
- Snoussi E (1998) Necessary conditions for multistationarity and stable periodicity. J Biol Syst 6:3–9
- Sontag ED (2005) Molecular systems biology and control. Eur J Control 11(4–5):396–435
- Steuer R, Waldherr S, Sourjik V, Kollmann M (2011) Robust signal processing in living cells. PLoS Comput Biol 7(11):e1002218
- Thomas R (1981) On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations. In: Della Dora J, Demongeot J, Lacolle B (eds) Numerical methods in the study of critical phenomena, vol 9. Springer series in synergetics. Springer, Berlin/Heidelberg

Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems

Andrew Packard¹, Peter Seiler², and Gary Balas²

¹Mechanical Engineering Department, University of California, Berkeley, CA, USA

²Aerospace Engineering and Mechanics Department, University of Minnesota, Minneapolis, MN, USA

Abstract

This entry presents the most commonly used formulations of robust stability and robust $\mathcal{H}_\infty$ performance for linear systems

with highly structured, linear, time-invariant uncertainty. The structured singular value function (μ) is specifically defined for this purpose, involving a problem-specific set, called the *uncertainty* set. With the uncertainty set chosen, μ is a real-valued function defined on complex matrices of a fixed dimension. A few key properties are easily derived from the definition and then applied to solve the robustness analysis problem. Computation of μ , which is required to implement the analysis tests, is difficult, so computable and refinable upper and lower bounds are derived.

Keywords

Robustness analysis · Robust control · Structured uncertainty

Notation, Definition, and Properties

$\mathbf{R}$ and $\mathbf{C}$ are the real and complex numbers; $\mathbf{C}_+ = \{\gamma \in \mathbf{C} : \text{Re}(\gamma) \geq 0\}$; $\mathbf{C}^n$ is the set of $n \times 1$ vectors and $\mathbf{C}^{n \times m}$ the set of $n \times m$ matrices with elements in $\mathbf{C}$. $\bar{\sigma}(\cdot)$ refers to the maximum singular value of a matrix; for $A \in \mathbf{C}^{n \times n}$, $\rho(A)$ is the spectral radius (largest, in magnitude, eigenvalue of A), and $\rho_{\mathbf{R}}(A)$ is the real spectral radius (largest, in magnitude, real, eigenvalue of A); $\mathcal{R}$ is the ring of proper rational functions, $S = \{g \in \mathcal{R} : g \text{ has no poles in } \mathbf{C}_+\}$; $S^{\bullet \times \bullet}$ denotes matrices with elements in S , where the exact dimensions are unspecified, but clear from context; finally, no notational distinction is made between a linear system, its transfer function, and/or its frequency response function.

Let R, S , and F be nonnegative integers and $r_1, \dots, r_R$, $s_1, \dots, s_S$, and $f_1, \dots, f_F$ be positive integers. Define sets $\Delta_{\mathbf{R}} := \{\text{diag}[\delta_1 I_{r_1}, \dots, \delta_R I_{r_R}] : \delta_i \in \mathbf{R}\}$,

$$\Delta_{\mathbf{C}} := \{\text{diag}[\delta_1 I_{s_1}, \dots, \delta_S I_{s_S}, \Delta_1, \dots, \Delta_F] : \delta_i \in \mathbf{C}, \Delta_k \in \mathbf{C}^{f_k \times f_k}\}$$

and their diagonal augmentation, $\Delta := \{\text{diag}[\Delta_R, \Delta_C] : \Delta_R \in \Delta_{\mathbf{R}}, \Delta_C \in \Delta_{\mathbf{C}}\} \subseteq$

$\mathbf{C}^{n \times n}$. The set Δ is called the *block structure*. The block structure can be generalized to handle nonsquare blocks in Δ_C at the expense of additional notation. If $R = 0$, then Δ is called a *complex* block structure. If $S = F = 0$, then Δ is called a *real* block structure. For $M \in \mathbf{C}^{n \times n}$, $\mu_\Delta(M)$ is defined as

$$\mu_\Delta(M) := \frac{1}{\min\{\bar{\sigma}(\Delta) : \Delta \in \Delta, \det(I - M\Delta) = 0\}}$$

unless no $\Delta \in \Delta$ makes $I - M\Delta$ singular, in which case $\mu_\Delta(M) := 0$, (Doyle 1982; Safonov 1982). The function $\mu_\Delta(\cdot) : \mathbf{C}^{n \times n} \rightarrow \mathbf{R}$ is upper semicontinuous. Following Fan et al. (1991), the constraint set in the definition can be written as $\{\bar{\sigma}(\Delta) : \exists w, z \in \mathbf{C}^n, w = Mz, z = \Delta w, w \neq 0_n\}$, so that without loss of generality, at the minimum, the elements $\Delta_1, \dots, \Delta_F$ each have rank equal to 1. For specific block structures, simplifications occur: if $R = S = 0$ and $F = 1$, then $\mu_\Delta(M) = \bar{\sigma}(M)$; if $R = F = 0$ and $S = 1$, then $\mu_\Delta(M) = \rho(M)$; and if $S = F = 0$ and $R = 1$, then $\mu_\Delta(M) = \rho_R(M)$. In general $\rho_R(M) \leq \mu_\Delta(M) \leq \bar{\sigma}(M)$. Associated with Δ define $\mathbf{B}_\Delta := \{\Delta \in \Delta : \bar{\sigma}(\Delta) \leq 1\}$. Since $I - M\Delta$ is singular if and only if $M\Delta$ has an eigenvalue exactly equal to 1, it follows that $\mu_\Delta(M) = \max_{\Delta \in \mathbf{B}_\Delta} \rho_R(M\Delta)$. If Δ is a complex block structure, then $\rho_R(\cdot)$ can be replaced with $\rho(\cdot)$, and in that case $\mu_\Delta(\cdot) : \mathbf{C}^{n \times n} \rightarrow \mathbf{R}$ is continuous.

A common application is to quantify the effect (in structured singular value terms) that an uncertain matrix Δ has on the expression $F_L(M, \Delta) := M_{11} + M_{12}\Delta(I - M_{22}\Delta)^{-1}M_{21}$, a *linear fractional transformation* (LFT) of Δ by M . This is conceptually straightforward (informally called the *main loop theorem*) using the Schur formula for determinants. Specifically, let $\Delta_1 \subseteq \mathbf{C}^{n_1 \times n_1}$, $\Delta_2 \subseteq \mathbf{C}^{n_2 \times n_2}$ be block structures Δ and $\subseteq \mathbf{C}^{(n_1+n_2) \times (n_1+n_2)}$ be their block-diagonal augmentation. For $M \in \mathbf{C}^{(n_1+n_2) \times (n_1+n_2)}$, $\mu_\Delta(M) < 1$ if and only if $\mu_{\Delta_2}(M_{22}) < 1$ and

$$\max_{\Delta_2 \in \mathbf{B}_{\Delta_2}} \mu_{\Delta_1}(F_L(M, \Delta_2)) < 1.$$

Finally (Packard and Pandey 1993) if Δ_1 is a block structure, and Δ_2 is a complex block structure, and M satisfies $\mu_{\Delta_1}(M_{11}) < \mu_\Delta(M)$, then $\mu_\Delta(\cdot)$ is continuous on an open ball around M . Loosely speaking, “if there are any complex blocks, and M is such that they matter, then μ is continuous at M .” This means that at points of discontinuity, only $\Delta_R \in \Delta_R$ need to be nonzero. For any polynomial $p : \mathbf{C}^n \rightarrow \mathbf{C}$, there is a minimum-norm root (using $\|\cdot\|_\infty$ on $\mathbf{C}^n$) whose components all have equal modulus (Doyle 1982). Defining

$$\mathbf{Q}_\Delta := \{\text{diag}[\Delta_R, \Delta_C] : \bar{\sigma}(\Delta_R) \leq 1, \Delta_C^* \Delta_C = I\}$$

and employing this result (Young and Doyle 1997) derives that $\mu_\Delta(M) = \max_{Q \in \mathbf{Q}_\Delta} \rho_R(MQ)$. This gives a generalized maximum-modulus-like theorem for LFTs (Packard and Pandey 1993). Revisiting the setup for the main loop theorem, assume further that Δ_2 is a complex block structure. If $\mu_{\Delta_2}(M_{22}) < 1$, then

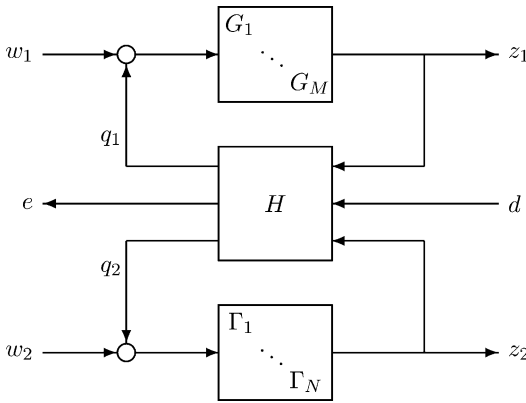
$$\max_{\Delta_2 \in \mathbf{B}_{\Delta_2}} \mu_{\Delta_1}(F_L(M, \Delta_2)) = \max_{Q_2 \in \mathbf{Q}_{\Delta_2}} \mu_{\Delta_1}(F_L(M, Q_2)).$$

This leads to specialized results per Boyd and Desoer (1985), Packard and Pandey (1993), and Tits and Fan (1995) for stable transfer function matrices. For any block structure $\Delta \subseteq \mathbf{C}^{n \times n}$ and $M \in \mathbf{S}^{n \times n}$, then

$$\begin{aligned} & \max \left\{ \sup_{\omega \in \mathbf{R}} \mu_\Delta(M(j\omega)), \mu_\Delta(M(\infty)) \right\} \\ &= \max \left\{ \sup_{s \in \mathbf{C}_+} \mu_\Delta(M(s)), \mu_\Delta(M(\infty)) \right\}. \end{aligned}$$

Robustness of Stability and Performance

There are several uncertain system formulations that all result in the same μ -analysis test to assess the robustness of stability and/or performance (Wall et al. 1982; Foo and Postlethwaite 1988). In this article, we present the simplest and most common interpretation. Consider an interconnec-



Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems, Fig. 1 Interconnection of $G_1, \dots, G_M, \Gamma_1, \dots, \Gamma_N$

tion of known systems, $\{G_i\}_{i=1}^M$, and unknown systems $\{\Gamma_k\}_{k=1}^N$, as described by

$$\begin{bmatrix} q_1 \\ e \\ q_2 \end{bmatrix} = H \begin{bmatrix} z_1 \\ d \\ z_2 \end{bmatrix}$$

where $z_1 = \text{diag}[G_1, \dots, G_M](q_1 + w_1)$, $z_2 = \text{diag}[\Gamma_1, \dots, \Gamma_N](q_2 + w_2)$, and $H \in \mathbf{R}^{(n_1+n_e+n_2) \times (p_1+n_d+p_2)}$ (naturally partitioned as a block 3-by-3 array). This is depicted in Fig. 1. Each G_i and Γ_k is assumed to be a finite-dimensional, time-invariant linear system, with proper transfer function, and a stabilizable and detectable internal state-space description.

The interconnection is *well posed* if for any initial conditions and any (say) piecewise continuous inputs w_1, w_2 , and d , there exist unique solutions to the interconnection equations. By manipulating the state-space or transfer function descriptions of a well-posed interconnection, a state-space model or proper transfer function description for the map from (d, w) to (e, z) can be derived. A well-posed interconnection is *stable* if the resultant state-space model is internally stable – the eigenvalues of its “A” matrix are in the open, left-half plane. Given some restrictions on the values of the elements of Γ , *robustness analysis* poses the question: is the interconnection well posed and stable for all possible values

of Γ ? And if so, then is the $\|\cdot\|_\infty$ gain from d -to- $e \leq 1$ for all possible values of Γ ? The goal of the analysis is to confirm “yes” or supply a particular Γ which proves that the answer is “no” (by rendering the interconnection ill-posed, unstable, or with d -to- e gain > 1). Standard linear systems theory gives that the interconnection is well posed if and only if

$$\det \left(I - \begin{bmatrix} H_{11} & H_{13} \\ H_{31} & H_{33} \end{bmatrix} \begin{bmatrix} G(\infty) & 0 \\ 0 & \Gamma(\infty) \end{bmatrix} \right) \neq 0,$$

and that the interconnection is stable if and only if the transfer function matrix $T^{w,z}$, mapping $[w_1; w_2]$ to $[z_1; z_2]$, is an element of $S^{\bullet \times \bullet}$.

The assumptions on each Γ_k are of three kinds: (i) Γ_k is a stable linear system, known only to satisfy $\|\Gamma_k\|_\infty < 1$; (ii) Γ_k is a stable linear system of the form $\gamma_k I$, where the scalar linear system γ_k is known to satisfy $\|\gamma_k\|_\infty < 1$; (iii) Γ_k is a constant gain, of the form $\gamma_k I$, where the scalar $\gamma_k \in \mathbf{R}$ is known to satisfy $-1 < \gamma_k < 1$. Note the similarity between this and the block structure Δ (via $\Delta_{\mathbf{R}}$ and $\Delta_{\mathbf{C}}$) introduced earlier. After rearrangement, this block-diagonal augmentation of uncertain systems is a norm-bounded (by 1) element of the set

$$\Gamma := \{ \text{diag}[\Gamma_R, \Gamma_U] : \Gamma_R \in \Delta_{\mathbf{R}}, \Gamma_U \in S^{\bullet \times \bullet}, \Gamma_U(s_0) \in \Delta_{\mathbf{C}} \forall s_0 \in \mathbf{C}_+ \}.$$

Since 0 is a possible value of Γ , two necessary conditions (denoted c.1 and c.2, respectively) for robust well-posedness and stability are at $\Gamma = 0$, specifically $\det(I - G(\infty)H_{11}) \neq 0$ and $V := G(s)(I - H_{11}G(s))^{-1} \in S^{\bullet \times \bullet}$. Assuming $\det(I - G(\infty)H_{11}) \neq 0$ (i.e., c.1), the Schur formula for block determinants reduces the well-posedness condition to

$$\det \left(I - \Gamma(\infty) \left[H_{33} + H_{31}(I - G(\infty)H_{11})^{-1} G(\infty)H_{13} \right] \right) \neq 0.$$

Define $M := H_{33} + H_{31}G(I - H_{11}G)^{-1}H_{13} \in S^{\bullet \times \bullet}$, and $X := I - \Gamma M$. Then

$$T^{w,z} = \begin{bmatrix} V + VH_{13}X^{-1}\Gamma H_{31}V & VH_{13}X^{-1}\Gamma \\ X^{-1}\Gamma H_{31}V & X^{-1}\Gamma \end{bmatrix}$$

Assuming c.2, namely, $V \in \mathbf{S}^{\bullet \times \bullet}$, then $X^{-1} \in \mathbf{S}^{\bullet \times \bullet}$ implies that $T^{w,z} \in \mathbf{S}^{\bullet \times \bullet}$ – moreover $T^{w,z} \in \mathbf{S}^{\bullet \times \bullet}$ implies that $X^{-1} = I + T_{22}^{w,z}M \in \mathbf{S}^{\bullet \times \bullet}$. Finally, since both M and Γ are stable, it follows that $X^{-1} \in \mathbf{S}^{\bullet \times \bullet}$ if and only if $\det(I - M(s_0)\Gamma(s_0)) \neq 0 \quad \forall s_0 \in \mathbf{C}_+$. The maximum-modulus property gives the robustness theorem. With the definition of M and conditions c.1 and c.2, the uncertain system is robustly stable and well posed if and only if

$$\max \left\{ \sup_{\omega \in \mathbf{R}} \mu_{\Delta}(M(j\omega)), \mu_{\Delta}(M(\infty)) \right\} \leq 1.$$

Indeed, if the condition holds, then by maximum-modulus theorem and the definition of μ , it follows that $\det(I - M(s)\Gamma(s)) \neq 0$ for all $s \in \mathbf{C}_+$ as well as $s = \infty$, since $\Gamma(s) \in \Delta$ and $\bar{\sigma}(\Gamma(s)) < 1$. This gives well-posedness and stability for all such Γ , as desired (an alternate proof, using the Nyquist criterion is also common). Conversely, if the condition is violated, then at some frequency (0, nonzero, or ∞), μ is larger than 1, as evidenced by a (constant matrix) $\Delta \in \Delta \subseteq \mathbf{C}^{n \times n}$, $\bar{\sigma}(\Delta) < 1$, which causes singularity. If the frequency is nonzero (and finite), the interpolation lemmas in the appendix enable replacing the complex blocks with stable, real-rational entries. Otherwise (0 or ∞), the matrix is such that μ is continuous, and hence a finite, nonzero frequency also has $\mu > 1$, or only the real blocks are necessary to cause singularity. In all cases, $\Gamma \in \mathbf{\Gamma}$ with $\|\Gamma\|_{\infty} < 1$ exists to cause ill-posedness or instability (Tits and Fan 1995).

Robustness of performance, measured as $\|T^{e,d}\|_{\infty}$, can be addressed, using the main loop theorem, and an additional complex full block (recall $\bar{\sigma}(\cdot) = \mu_{\Delta}(\cdot)$ when $F = 1, S = R = 0$). Define

$$M_P := \begin{bmatrix} H_{22} & H_{23} \\ H_{32} & H_{33} \end{bmatrix} + \begin{bmatrix} H_{21} \\ H_{31} \end{bmatrix} G(I - H_{11}G)^{-1} \begin{bmatrix} H_{12} & H_{13} \end{bmatrix}$$

and $\Delta_P := \{\text{diag}[\Delta_P, \Delta] : \Delta_P \in \mathbf{C}^{n_d \times n_e}, \Delta \in \Delta\}$. With conditions c.1 and c.2, the uncertain system is robustly stable and well posed and satisfies $\|T^{e,d}\|_{\infty} \leq 1$ if and only if

$$\max \left\{ \sup_{\omega \in \mathbf{R}} \mu_{\Delta_P}(M_P(j\omega)), \mu_{\Delta_P}(M_P(\infty)) \right\} \leq 1.$$

Computations

The robust stability and robust performance theorems require computing μ on the frequency response function $M(j\omega)$. Computing μ is known to be a computationally difficult problem (Toker and Ozbay 1998), so exact computational methods are generally not pursued. Reliable algorithms have been developed which yield upper and lower bounds, which are often sufficiently close for many engineering problems.

Lower Bounds

Recall that $\mu_{\Delta}(M) = \max_{\Delta \in \mathbf{B}_{\Delta}} \rho_{\mathbf{R}}(M\Delta) = \max_{Q \in \mathbf{Q}_{\Delta}} \rho_{\mathbf{R}}(MQ)$. Practically speaking, these maximizations yield lower bounds for $\mu_{\Delta}(M)$, since the global maximum may not be attained. In addition to gradient-based ascent methods, the optimality conditions for $Q \in \mathbf{Q}_{\Delta}$ to be a local maximum of the function $\rho_{\mathbf{R}}(M\Delta)$ on the set $\mathbf{B}_{\Delta}$ can be derived (Young and Doyle 1997). A solution approach, similar to a Jacobi iteration, leads to an iteration that resembles combinations of the familiar power methods for spectral radius and maximum singular value. If the iteration converges (which is not guaranteed), a lower bound for $\mu_{\Delta}(M)$ (along with a corresponding $\Delta \in \Delta$) is produced. Studies with matrices constructed to have $\mu_{\Delta}(M) = 1$ suggest that the iteration is very reliable for complex block structures, though usually quite poor for purely real block structures. There are several, more computationally demanding algorithms available for purely real block structures (de Gaston and Safonov 1988; Sideris and Sanchez Pena 1989). For the common situation, with both real and complex blocks, where continuity is assured, the power algorithm generally has adequate performance.

Upper Bounds

Define $\mathbf{G}_\Delta := \{G = -G^* : G\Delta = -\Delta^*G^* \forall \Delta \in \Delta\}$, $\mathbf{D}_\Delta := \{D = D^* > 0 : D\Delta = \Delta D \forall \Delta \in \Delta\}$, subsets of $\mathbf{C}^{n \times n}$. Elements of $\mathbf{D}_\Delta$ are of the form $\text{diag}[D_{r_1}, \dots, D_{r_R}, D_{s_1}, \dots, D_{s_S}, d_1 I_{f_1}, \dots, d_F I_{f_F}]$, and therefore $D \in \mathbf{D}_\Delta$ implies that $D^{\frac{1}{2}} \in \mathbf{D}_\Delta$ too. Likewise,

$$\mathbf{G}_\Delta := \{\text{diag}[G_R, 0] : G_R = -G_R^* \in \mathbf{C}^{\bullet \times \bullet}, \\ G_R \Delta_R = \Delta_R G_R \forall \Delta_R \in \Delta_R\}.$$

A concise derivation (Helmersson 1995) verifies the upper bound formula (Fan et al. 1991). If $\beta > 0$, $G \in \mathbf{G}_\Delta$, and $D \in \mathbf{D}_\Delta$ satisfy $M^*DM - \beta^2 D + GM + M^*G^* \leq 0$, then $\mu_\Delta(M) \leq \beta$. Indeed, if $\Delta \in \Delta$ has $\det(I - M\Delta) = 0$, there exist nonzero $w, z \in \mathbf{C}^n$ with $w = Mz$, $z = \Delta w$. Certainly $z^*(M^*DM - \beta^2 D + GM + M^*G^*)z \leq 0$. Making substitutions gives

$$\begin{aligned} 0 &\geq w^*Dw - \beta^2 w^*\Delta^*D\Delta w \\ &\quad + w^*\Delta^*Gw + w^*G^*\Delta w \\ &= w^*Dw - \beta^2 w^*D^{\frac{1}{2}}\Delta^*D^{\frac{1}{2}}w \\ &\quad + w + w^*\Delta^*Gw - w^*\Delta^*Gw \\ &= w^*D^{\frac{1}{2}}\left(I - \beta^2\Delta^*\Delta\right)D^{\frac{1}{2}}w. \end{aligned}$$

Since D is invertible and $w \neq 0_n$, it must be that $\bar{\sigma}(\Delta) \geq \beta^{-1}$, as desired. The constraint $M^*DM - \beta^2 D + GM + M^*G^* \leq 0$ is a linear matrix inequality (LMI) in the variables D and G . Minimizing β over $G \in \mathbf{G}_\Delta$ and $D \in \mathbf{D}_\Delta$ subject to the LMI constraint (using Boyd and El Ghaoui 1993, for instance) yields the best upper bound that this inequality can produce.

Further Perspectives

The robustness tests involve bounding $\mu_\Delta(M(j\omega))$ over the entire real axis. A common approach is to use a dense frequency gridding and upper/lower bound calculations at each gridded point. The advantages, simplicity and trivial parallelization, are offset with disadvantages, in that the peak value (over $\mathbf{R}$) may not be reflected accurately by the peak across the finite grid. In fact, such a grid-based test determines the

smallest $\Delta \in \Delta$ which can cause a pole to migrate from the left-half plane into the right-half plane *at exactly one of the frequency grid points* (as opposed to any location). Nevertheless, with some continuity assurances in place and a dense grid, this is often adequate knowledge for most engineering decisions. However, the brute-force grid approach can be avoided by treating the frequency-variable (ω) as an additional real parameter (since $M(j\omega)$ is an LFT of $\frac{1}{\omega}$) (Ferrerres et al. 2003). This is a generalization of the Hamiltonian methods to compute the $\mathcal{H}_\infty$ norm of a linear system without a frequency grid, coupled with an alternative form of the upper bound (Young et al. 1995). Moreover, if only the peak value (upper bound, say) across frequency is desired, this approach can be fast, as some calculations rule out large frequency ranges to not contain the peak.

Improved upper bounds can be derived using higher-order arguments, changing the LMI constraint into a sum-of-squares constraint (which ultimately is just a larger LMI). Alternatively, branch-and-bound techniques are especially useful at reducing the conservativeness of the (D, G) upper bound when there are several real parameters ($R > 0$) (Newlin and Young 1997).

Appendix: Interpolation Lemmas

Two interpolation lemmas make the connection between robustness to constant-gain, complex-valued uncertainties (Δ) and stable, finite-dimensional, time-invariant linear systems described by ODEs with real coefficients (Γ). Lemma 1 is used (block by block and element by element on the relevant vector directions within each block) to interpolate complex blocks causing singularity into real-rational blocks which cause singularity at a particular frequency.

Lemma 1 *Given a positive $\bar{\omega} > 0$ and a complex number δ , with $\text{Imag}(\delta) \neq 0$, there is a $\beta > 0$ such that by proper choice of sign $\pm |\delta| \frac{s-\beta}{s+\beta} \Big|_{s=j\bar{\omega}} = \delta$.*

Lemma 2 *Suppose $M \in \mathbf{C}^{n \times n}$ and $\bar{\omega} > 0$. If $\Delta \in \Delta$ satisfies $\det(I_n - M\Delta) = 0$, then there*

is a $\Gamma \in \mathbf{\Gamma}$ with $\|\Gamma\|_\infty \leq \bar{\sigma}(\Delta)$ and $\det(I_n - M\Gamma(j\bar{\omega})) = 0$.

Summary and Future Directions

The structured singular value, μ , is a linear algebra construct, defined to exactly deal with linear, time-invariant uncertainty in linear systems. The main issues are computational, focused on efficient manners to compute reasonably tight upper and lower bounds at each frequency and, more specifically, ascertain the peak value across frequency. Alternatives to the worst-case approach to robustness analysis are gaining favor and may be applicable in analysis and design situations where the abstraction of a worst-case view is too conservative (Calafiore et al. 2000).

Cross-References

- [Fundamental Limitation of Feedback Control](#)
- [KYP Lemma and Generalizations/Applications](#)
- [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)
- [LMI Approach to Robust Control](#)
- [Optimization-Based Robust Control](#)
- [Robust Control in Gap Metric](#)
- [Robust Fault Diagnosis and Control](#)
- [Robust \$\mathcal{H}_2\$ Performance in Feedback Control](#)

Recommended Reading

A comprehensive list of references, including theory, computations, and diverse applications would require many pages. The list below is minimal and does not do justice to the many researchers who have made significant contributions to this subject. In addition to the cited work, connections to Kharitonov's theorem can be found in Chen et al. (1994). Textbooks, such as Dullerud and Paganini (2000) and Zhou et al. (1996), include derivations and additional citations.

Bibliography

- Boyd S, Desoer CA (1985) Subharmonic functions and performance bounds on linear time-invariant feedback systems. *IMA J Math Control Inf* 2:153–170
- Boyd S, El Ghaoui L (1993) Method of centers for minimizing generalized eigenvalues. *Linear Algebra Appl* 188:63–111
- Calafiore GC, Dabbene F, Tempo R (2000) Randomized algorithms for probabilistic robustness with real and complex structured uncertainty. *IEEE Trans Autom Control* 45(12):2218–2235
- Chen J, Fan MKH, and Nett CN (1994) Structured singular values and stability analysis of uncertain polynomials, part 1 and 2. *Syst Control Lett* 23:53–65 and 97–109
- de Gaston RRE, Safonov MG (1988) Exact calculation of the multiloop stability margin. *IEEE Trans Autom Control* 33(2):156–171
- Doyle J (1982) Analysis of feedback systems with structured uncertainties. *IEE Proc Part D* 129(6):242–250
- Dullerud G, Paganini F (2000) A course in robust control theory, vol 6. Springer, New York
- Fan MKH, Tits AL, Doyle JC (1991) Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. *IEEE Trans Autom Control* 36(1):25–38
- Ferreres G, Magni JF, Biannic JM (2003) Robustness analysis of flexible structures: practical algorithms. *Int J Robust Nonlinear Control* 13(8):715–733
- Foo YK, Postlethwaite I (1988) Extensions of the small- μ test for robust stability. *IEEE Trans Autom Control* 33(2):172–176
- Helmersson A (1995) Methods for robust gain scheduling. PhD thesis, Linköping
- Newlin MP, Young PM (1997) Mixed μ problems and branch and bound techniques. *Int J Robust Nonlinear Control* 7:145–164
- Packard A, Pandey P (1993) Continuity properties of the real/complex structured singular value. *IEEE Trans Autom Control* 38(3):415–428
- Safonov MG (1982) Stability margins of diagonally perturbed multivariable feedback systems. *Control Theory Appl IEE Proc D* 129:251–256. IET
- Sideris A, Sanchez Pena RS (1989) Fast computation of the multivariable stability margin for the real interrelated uncertain parameters. *IEEE Trans Autom Control* 34(12):1272–1276
- Tits A, Fan MKH (1995) On the small- μ theorem. *Automatica* 31(8):1199–1201
- Toker O, Ozbay H (1998) On the complexity of purely complex μ ; computation and related problems in multidimensional systems. *IEEE Trans Autom Control* 43(3):409–414
- Wall J, Doyle JC, Stein G (1982) Performance and robustness analysis for structured uncertainty. In: IEEE conference on decision and control, Orlando, pp 629–636

- Young PM, Doyle JC (1997) A lower bound for the mixed μ problem. *IEEE Trans Autom Control* 42(1):123–128
- Young P, Newlin M, Doyle J (1995) Computing bounds for the mixed μ problem. *Int J Robust Nonlinear Control* 5(6):573–590
- Zhou K, Doyle JC, Glover K (1996) *Robust and optimal control*, vol 40. Prentice Hall, Upper Saddle River

Sub-Riemannian Optimization

Roger Brockett

Harvard University, Cambridge, MA, USA

Abstract

Optimization problems arising in the control of some important types of physical systems lead naturally to problems in sub-Riemannian optimization. Here we provide context and background material on the relevant mathematics and discuss some specific problem areas where these ideas play a role.

Keywords

Carnot-carathéodory metric · Lie algebras · Periodic processes · Subelliptic operators · Sub-riemannian geodesics · Symmetric spaces

Introduction

After a start in the early 1970s, over the last two decades, sub-Riemannian geometry and the related theory of subelliptic operators have become popular topics in the control literature. Their study is sometimes linked to questions involving the dynamics and control of mechanical systems with nonholonomic (nonintegrable) constraints and the use of what has classically been called quasi-coordinates because both subjects depend on Lie algebraic techniques. However, here we limit ourselves to

problems in sub-Riemannian optimization per se, describing how they arise in various areas of physics and engineering. Most famously, the second law of thermodynamics, as recast by Carathéodory in differential geometric form, provides an example of the reach of sub-Riemannian geometry into the engineering world.

The statement of control theoretic problems often begins with a description of the system of interest in differential equation form:

$$\dot{x} = f(x) + \sum u_i g_i(x) ; x \in X, u \in \mathbb{R}^m$$

with X an n -dimensional manifold. In well-motivated control problems, n is almost always larger than m ; the dimension of the space of controls is less than the dimension of the state space. In the case of mechanical systems, the phrase *under actuated* is sometimes used to characterize this, but the situation is ubiquitous. The analysis is complicated by presence of the immutable *drift term* f . When it is desired to use an optimization principle to find a good choice for u , one introduces a performance measure, often of the form

$$\eta = \int_0^{t_1} L(x, u) dt$$

and attempts to minimize η subject to whatever constraints there may be on u and x . If there is no drift term and if the Lie algebra generated by $\{g_1, g_2, \dots, g_m\}$ defines a distribution that spans the tangent space of X at every point, the problem falls under the purview of *sub-Riemannian geometry*. In this case, one can describe the situation as $\dot{x} = G(x)u$ with G being an x -dependent rectangular matrix of rank m everywhere.

This entry is written from a control theory point of view. The problems discussed here provided the impetus for some later mathematical work, often not discussing the motivation. The purely mathematical work is de-emphasized here, much as the mathematical work often gives little or no attention to the control theoretic work that preceded it.

The Distance Function

A prototype control problem leading to sub Riemannian geometry is that of steering the system $\dot{x}_1 = u_1$ $\dot{x}_2 = u_2$ $\dot{x}_3 = x_1 u_2 - x_2 u_1$ from one state to another while minimizing

$$\eta = \int_0^1 \sqrt{u_1^2 + u_2^2} dt$$

It might seem that this is just a minor change from a standard shortest path problem in Riemannian geometry, e.g., it might be thought as a limiting case of a standard Riemannian geodesic problem in which the infinitesimal length is given by

$$(ds)^2 = \begin{bmatrix} dx_1 & dx_2 & dx_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & -y \\ 0 & 1 & x \\ -y & x & \epsilon + x^2 + y^2 \end{bmatrix}^{-1} \begin{bmatrix} dx_1 \\ dx_2 \\ dx_3 \end{bmatrix}$$

and ϵ is allowed to go to zero. However, because when ϵ equals zero this matrix is singular, it cannot be used to define the equations for geodesics. The most direct attack seems to be to use a Lagrange multiplier to enforce the condition on x_3 , which leads to the minimization of

$$\eta = \int_0^1 \dot{x}_1^2 + \dot{x}_2^2 + \lambda(x_1 \dot{x}_2 - x_2 \dot{x}_1) dt$$

This yields a set of λ -dependent linear equations for x_1 and x_2 . Solving these shows that the projections of the minimum length trajectories onto the (x_1, x_2) -plane are circular arcs.

In Riemannian geometry, the set of points which are of distance r from a given point will, for r sufficiently small, form a co-dimension one manifold diffeomorphic to a sphere. In this qualitative sense, Riemannian spaces are locally isotropic. In sub-Riemannian geometry, the set of points of distance $r > 0$ from a distinguished point x_0 does not have such a simple structure. For example, for the problem just discussed, we have the approximations

$$d = \sqrt{x_1^2 + x_2^2} + |x_3|/(x_1^2 + x_2^2)$$

$$\text{for } |x_3| \ll (x_1^2 + x_2^2)$$

and

$$d = 2\pi|x_3| - \sqrt{8\pi(x_1^2 + x_2^2)}|x_3|$$

$$\text{for } \sqrt{x_1^2 + x_2^2} \ll |x_3|$$

That is, for points bounded by paraboloids, defining a region near the (x_1, x_2) -plane, the distance is close to the Riemannian distance, whereas in a cone containing the x_3 axis, the distance is close to the square root of the Riemannian distance. These approximations make it clear that $d(x_1, x_2, x_3)$ is not differentiable at points on the x_3 axis. There is much more that can be said here. One interesting topic concerns the number of trajectories that satisfy the first-order necessary conditions and join a point to the origin.

More Examples

Consider the kinematic equations of the unicycle. If (x, y) are the coordinates of the center of the wheel and θ is the heading angle, then these are

$$\dot{x} = \cos \theta u_2 ; \dot{y} = \sin \theta u_2 ; \dot{\phi} = u_1$$

It is of interest to generate a “shortest path” between two points in (x, y, θ) -space where shortest is defined as the integral of some function of x, y, θ, u_1, u_2 . This is typical of the kind of path planning problems in which nonholonomic constraints lead to sub-Riemannian problems. A variety of such problems arise in robotics with optimal steering programs for cars being one example.

As an example involving a compact manifold, let X be the space of 3-by-3 orthogonal matrices and consider the system described by

$$\dot{x} = \begin{bmatrix} 0 & u_1 & u_2 \\ -u_1 & 0 & 0 \\ -u_2 & 0 & 0 \end{bmatrix} x$$

In this case, the manifold X is three dimensional and the control space is two dimensional. If we wish to minimize the integral of $u^2 + v^2$ subject to $x(0) = x_0$ and $x(1) = x_1$, we have a typical sub-Riemannian geodesic problem.

If the controls contain random effects, efforts to analyze the situation lead to related problems in stochastic process. The most widely studied of these are described by an Itô equation of the form

$$dx = f(x)dt + \sum g_i(x)dw_i$$

The corresponding equation for the evolution of the probability density $\rho(t, x)$ can be put in the form

$$\begin{aligned} \frac{\partial \rho}{\partial t} = & \sum a_i(x) \frac{\partial}{\partial x_i} \rho(t, x) \\ & + \sum b_{ij}(x) \frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} \rho(t, x) \end{aligned}$$

However, rather than the right-hand side being a fully elliptic operator, as it would be in a typical heat equation (e.g., the Laplace-Beltrami operator), the symmetric matrix $B(x) = b_{ij}(x)$ is singular. If the g_i satisfy the bracket-generating condition, the density equation is said to be *subelliptic*. The system described by the Itô equation

$$\begin{bmatrix} dx_1 \\ dx_2 \\ dx_3 \end{bmatrix} = \begin{bmatrix} -dt & dw_1 & dw_2 \\ -dw_1 & -dt/2 & 0 \\ -dw_2 & 0 & -dt/2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

evolves on the two-sphere and the spectrum of the subelliptic operator is discrete. The diffusion time constants, i.e., the eigenvalues of the subelliptic operator, can be computed explicitly and compared with those of the fully elliptic operator, i.e., the standard Laplacian on the spherical shell.

Much has been written on the ways in which subelliptic diffusion does, and does not, share the properties of the ordinary diffusion equation.

A Special Structure

A rich, and especially tractable, class of sub-Riemannian problems come from the following situation. Suppose that $\mathcal{G}$ is a Lie group with Lie

algebra G and that $\mathcal{H}$ is a closed subgroup with Lie algebra H . According to one definition, the pair $\mathcal{H} \subset \mathcal{G}$ is said to define a *symmetric space* if the Lie algebra G , viewed as a vector space, is the direct sum of H and K with $[H, K] \subset K$ and $[K, K] \subset H$. Let x evolve in $\mathcal{G}$ as

$$\dot{x} = ux ; \quad x \in \mathcal{G} ; \quad u \in K$$

For the sake of exposition, suppose that $\mathcal{G}$ is a matrix Lie group. We look for paths joining x_0 and x_1 that are shortest in the sense that

$$\eta = \int_0^1 \|u\| dt ;$$

is minimized, where $\|u\|^2 = \text{tr}(u^T u)$. (This leads to the same trajectories as those which minimize the integral of $\|u\|^2$.) To find the first-order necessary conditions using the maximum principle, define a Hamiltonian as $h(x, p, u) = \text{tr}(p^T ux + u^T u)$. Thus, $\dot{p} = -u^T p$ and minimizing over u implies $2u = -\pi_1(xp^T)$ where π_1 is the projection onto K . The product $m = xp^T$ satisfies $\dot{m} = [m, \pi_1(m)]$. Using the structural properties of the Lie algebra, we see that $(d/dt)\pi_0(m) = 0$ and that $(d/dt)\pi_1(m) = [\pi_1(m), m_0]$. Working out the implications, we see that trajectories of the form $x(t) = e^{at} e^{(b-a)t}$ with $a \in H$ and $b \in K$ satisfy the first-order optimality conditions.

To illustrate, we consider the generalization of an earlier example. Let X be the space of n -by- n orthogonal matrices and consider the system described by

$$\dot{x} = \begin{bmatrix} 0 & u_1 & u_2 & \cdots & u_{n-1} \\ -u_1 & 0 & 0 & \cdots & 0 \\ -u_2 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ -u_{n-1} & 0 & 0 & \cdots & 0 \end{bmatrix} x$$

Here the role of H is played by the sub-algebra of the set of real n -by- n skew-symmetric matrices consisting those whose first row and column vanish and K consists of the subset whose lower-right $(n-1)$ -by- $(n-1)$ sub-matrix vanishes. In this case, the paths satisfying the

first-order necessary conditions take the form $x(t) = e^{ht} e^{(k-h)t} x(0)$.

Nonintegrability and Cyclic Processes

Of course *nonintegrable* stands in opposition to the word *integrable*, as it is used in the consideration of integration performed along paths, e.g.,

$$I = \int_{\gamma} g_1(x) dx_1 + g_2(x) dx_2 + \cdots + g_n(x) dx_n$$

If the path γ starts at $\bar{x}$ and ends at $\hat{x}$, then the equality of mixed partials $\partial g_i / \partial x_j = \partial g_j / \partial x_i$ implies that along any two paths with these end points, the integral has the same value, provided that one of the paths can be continuously deformed into the other with the g_i being well defined along the deformation. In particular, if γ is a closed curve so $\bar{x} = \hat{x}$, then under these assumptions, the integral is zero.

On the other hand, there is a large list of important processes in biology and engineering, such as those involving the thermodynamic cycles of internal combustions engines or air conditioners, that depend critically on nonintegrable effects. These include cyclic phenomena such as walking and breathing and widely used mechanisms for efficient voltage conversion in electrical engineering. Thus, both nature and technology provide examples of processes in which the pistons, valves, etc. move along a smooth path and at the end of a cycle return to their initial configuration, while a related integral is not zero. Perhaps, the best-known path problem of this type is the Carnot cycle.

Questions about sub-Riemannian optimization enter here both as the optimization of the path defining the cycle and in the optimal regulation of the output of such cyclic processes. In general, the output can adjust both the amplitude and frequency of the cycle (volume of air per cycle and respiration rate), although in some cases one or the other of these might be fixed. For example, cruise control for automobiles regulates the frequency (rpm) of the engine but

cannot adjust the stroke length of the pistons, whereas speed control of a running animal ordinarily involves adjusting both the length of the stride and the “steps” per minute. The primary considerations for these control processes are stability and response time, with the shape of the cycles being determined by some measure of efficiency. It seems that the optimization of such regulatory processes deserves more attention.

Cross-References

- ▶ [Differential Geometric Methods in Nonlinear Control](#)
- ▶ [Feedback Stabilization of Nonlinear Systems](#)
- ▶ [Nonlinear Adaptive Control](#)
- ▶ [Optimal Control and Pontryagin's Maximum Principle](#)
- ▶ [Robustness Issues in Quantum Control](#)
- ▶ [Redundant Robots](#)
- ▶ [Singular Trajectories in Optimal Control](#)

Recommended Reading

Material on sub-Riemannian geometry can be found in the very readable survey (Strichartz 1986) and in more depth in Gromov (1996). The examples discussed here have mostly come from the literature Brockett (1973a, b), Baillieul (1975), and Brockett (1999) and these papers contain motivational material as well. Symmetric spaces are discussed in the sub-Riemannian context in Strichartz (1986), but for the optimization aspect, see Brockett (1999). Reference Brockett (2003) studies the regulation of sub-Riemannian cycles.

Bibliography

- Baillieul J (1975) Some optimization problems in geometric control theory. PhD thesis, Harvard University
 Brockett R (1973a) Lie theory and control systems defined on spheres. SIAM J Appl Math 25:213–225
 Brockett R (1973b) Lie algebras and lie groups in control theory. In: Mayne DQ, Brockett RW (eds) Geometric methods in system theory. Reidel, Dordrecht, pp 43–82

- Brockett R (1999) Explicitly solvable control problems with nonholonomic constraints. In: Proceedings of the 1999 CDC conference, Phoenix, pp 13–16
- Brockett R (2003) Pattern generation and the control of nonlinear systems. *IEEE Trans Autom Control* 48:1699–1712
- Gromov M (1996) Carnot–Carathéodory spaces seen from within. *Sub-Riemannian geometry*. Progress in mathematics, vol 144. Birkhäuser, Basel, pp 79–323
- Strichartz RS (1986) Sub-Riemannian geometry. *J Diff Geom* 24:221–263

Subspace Techniques in System Identification

Michel Verhaegen

Delft Center for Systems and Control, Delft University, Delft, The Netherlands

Abstract

An overview is given of the class of subspace techniques (STs) for identifying linear, time-invariant state-space models from input-output data. STs do not require a parametrization of the system matrices and as a consequence do not suffer from problems related to local minima that often hamper successful application of parametric optimization-based identification methods.

The overview follows the historic line of development. It starts from Kronecker's result on the representation of an infinite power series by a rational function and then addresses, respectively, the deterministic realization problem, its stochastic variant, and finally the identification of a state-space model given in innovation form.

The overview summarizes the fundamental principles of the algorithms to solve the problems and summarizes the results about the statistical properties of the estimates as well as the practical issues like choice of weighting matrices and the selection of dimension parameters in using these STs in practice. The overview concludes with probing some future challenges and makes suggestions for further reading.

Keywords

Extended observability matrix · Hankel matrix · Innovation model · State-space model · Singular value decomposition (SVD)

Introduction

Subspace techniques (STs) for system identification address the problem of identifying state-space models of MIMO dynamical systems. The roots of ST were laid by the German mathematician Leopold Kronecker (^o1823–[†]1891). In Kronecker (1890) Kronecker established that a power series could be represented by a rational function when the *rank of the Hankel* operator with that power series as its symbol was *finite*. In the early 1990s of the twentieth century, new generalizations of the idea of Kronecker were presented for identifying linear, time-invariant (LTI) state-space models from input-output data or output data only. These new generalizations were formulated from different perspectives, namely, within the context of canonical variate analysis (Larimore 1990), within a linear algebra context (Van Overschee and De Moor 1994; Verhaegen 1994), and subspace splitting (Jansson and Wahlberg 1996). Despite their different origins, the close relationship between these methods was quickly established by a unifying theorem that interpreted these methods as a singular value decomposition (SVD) of a weighted matrix from which an estimate of the column space of the observability matrix or the row space of the state sequence of the given system or Kalman filter for observing the state of that system is derived (Van Overschee and De Moor 1995). This subspace calculation is the key feature that leads to the indication by ST for system identification or subspace identification methods (SIM).

The STs are attractive *complementary* techniques to the maximum likelihood or prediction error framework. They do not require the user to specify a parametrization of the system matrices of the state-space model, and the user is not confronted with the problems due to possible local minima of a nonlinear parameter

optimization method that is often necessary in estimating the parameters of a state-space model via, e.g., prediction error methods. Though the statistical properties such as consistency and efficiency have been investigated, such as in Bauer and Ljung (2002), the estimates obtained via ST are in general not optimal in the statistical minimum variance sense. However, practical evidence with the use of ST in a wide variety of problems has indicated that ST provides accurate estimates. As such they are often used as an initialization to the maximum likelihood or prediction error parametric identification methods.

In this entry we make a distinction between *output only* or stochastic identification problems and *input-output* or combined deterministic-stochastic identification problems. The first occurs when identifying, e.g., the eigenmodes of a bridge from ambient acceleration responses of the bridge. The second occurs when, in addition to ambient excitations that cannot be directly measured, controlled excitations through actuators integrated in the system are used during the collection of the input-output data.

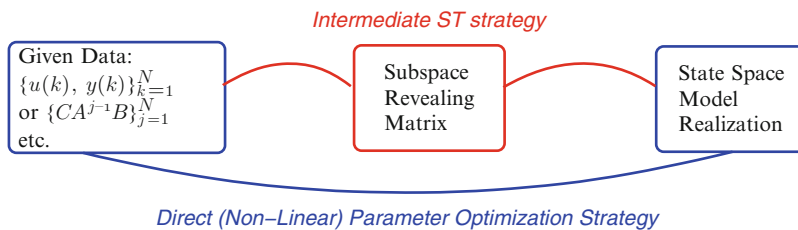
The outline of this entry is as follows. In the next section, we formulate the LTI state-space model identification problems and outline the general strategy of ST. The presentation of ST is given according to the historical development of ST. It starts with a summary of the solution to the deterministic realization problem, which considers the noise-free “impulse” response of the system. Subsequently we present the

stochastic realization problem which considers the output-only identification problem where the output is assumed to be a filtered zero-mean, white-noise sequence. The ST solution is discussed assuming samples of the covariance function of the output to be given. The deterministic-stochastic identification problem is considered in Section “[Combined Deterministic-Stochastic ST](#).” In this section we first consider open-loop identification experiments. For this case, the basic linear regression problem is formulated that is at the heart of many ST. Second reference is made to a framework for analyzing and understanding the statistical properties of ST, the selection of the order, as well as to a number of open problems in the understanding of important choices the user has to make. Closed-loop identification experiments are considered in the third part of Section “[Combined Deterministic-Stochastic ST](#),” while the fourth part makes a brief reference to ST papers that go beyond the LTI case.

Finally we provide a brief overview on future research directions and conclude with some recommended literature for further exploration.

ST in Identification: Problems and Strategy

The LTI system to be analyzed in this entry is given by the following state-space model:



Subspace Techniques in System Identification, Fig. 1 Schematic representation of the intermediate step of ST to derive from the given data (input-output data $\{u(k), y(k)\}$, Markov parameters $\{CA^{j-1}B\}$, etc.) a subspace revealing matrix, from which the subspace of interest is computed via, e.g., singular value decomposition and that enables the computation of the

state-space model realization by solving a (convex) linear least-squares problem. The commonly used approach to directly go from the given data to a state-space realization via in general nonlinear parameter optimization methods is indicated by the arrow directly connecting the given data box to the state-space realization box

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) + Ke(k) \\y(k) &= Cx(k) + Du(k) + e(k)\end{aligned}\quad (1)$$

with $u(k) \in \mathbb{R}^m$ the (measurable) input, $e(k)$ a zero-mean, white-noise sequence with $E[e(k)e(k)^T] = R$, $y(k) \in \mathbb{R}^\ell$ the (measurable) output, and $x(k) \in \mathbb{R}^n$ the state vector. This model is in the so-called innovation form since the sequence $e(k)$ is the innovation signal in a Kalman filtering context.

The historical sequence of ST developments considers the following open-loop problem formulations. In the deterministic realization problem, the innovation sequence $e(k)$ is zero, and the input $u(k)$ is an impulse. The stochastic realization problem considers the case where the input $u(k)$ is zero and the given data is assumed to be samples of the covariance function of the output. The combined deterministic-stochastic identification problem considers the model (1) for generic input $u(k)$.

The general strategy of ST is to formulate an *intermediate step* in deriving the parameters of the system matrices of interest from the given data; see Fig. 1. This intermediate step makes the ST different from the parametric model identification framework that aims for a direct estimation of the parameters of the system matrices by (in general) nonlinear parameter optimization techniques. The intermediate step in ST aims to determine a matrix from the given data that *reveals* an (approximation of an) essential subspace of the unknown system. This essential subspace can be the extended observability matrix of (1) as given by the matrix $\mathcal{O}_s$:

$$\mathcal{O}_s = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix} \quad \text{for } s > n,$$

or the state sequence of a Kalman filter designed for (1). Essential for ST is that both the intermediate step to reveal the subspace of interest and the subsequent derivation of the system matrices from that subspace and the given data are done

via convex optimization methods and/or linear algebra methods.

Realization Theory: The Progenitor of ST

The Deterministic Realization Problem

In the 1960s, the cited result of Kronecker inspired independently Ho and Kalman, Silverman and Youla, and Tissi to present an algorithm to construct a state-space model from a Hankel matrix of impulse response coefficients (Schutter 2000). This breakthrough gave rise to the field of *realization theory*. One key problem in realization theory that paved the way for subspace identification is the determination of a minimal realization from a finite number of samples of the impulse response of a deterministic system, assumed to have a minimal representation as in (1) for $e(k) \equiv 0$. The samples of the impulse response are called the *Markov parameters*. The minimal realization sought for is the LTI model with quadruple of system matrices $[A_T, B_T, C_T, D]$, with $A_T \in \mathbb{R}^{n \times n}$ and n minimal such that the pair (A_T, C_T) is observable, the pair (A_T, B_T) is controllable, and the transfer function $D + C_T(zI - A_T)^{-1}B_T$ equals $D + C(zI - A)^{-1}B$ with z the complex variable of the z -transform. When A is stable, the latter transfer function can be written into the matrix power series:

$$D + C(zI - A)^{-1}B = D + \sum_{j=1}^{\infty} CA^{j-1}Bz^{-j} \quad (2)$$

Following the cited result of Kronecker, the solution to the minimum realization problem is based on the construction of the (block-)Hankel matrix $H_{s,N}$ constructed from the Markov parameters $\{CA^{j-1}B\}_{j=1}^N$ as

$$H_{s,N} = \begin{bmatrix} CB & CAB & \cdots & CA^{N-s}B \\ \vdots & & \ddots & \vdots \\ CA^{s-1}B & CA^sB & \cdots & CA^{N-1}B \end{bmatrix} \quad (3)$$

For the deterministic realization problem, the *intermediate ST step* simply is the storage of the impulse response data into a Hankel matrix. The subsequent step is to derive from this matrix a subspace from which the system matrices can be either read-off or computed via linear least squares. How this is done is outlined next.

When the order n of the minimal realization is known and the Hankel matrix dimension parameters s, N are chosen such that

$$s > n \quad N \geq 2n - 1 \quad (4)$$

the Hankel matrix $H_{s,N}$ has **rank** n . A numerically reliable way to compute that rank is via the SVD of $H_{s,N}$. Under the assumption that the rank of $H_{s,N}$ is n , we can denote that SVD as $U_n \Sigma_n V_n^T$, with $\Sigma_n \in \mathbb{R}^{n \times n}$ positive definite and with the columns of the matrices U_n and V_n orthonormal. By the minimality of (1) (for $e(k) \equiv 0$), $H_{s,N}$ can be factored as $\mathcal{O}_s [B \ AB \ \cdots \ A^{N-s} B] = \mathcal{O}_s \mathcal{C}_{N-s+1}$ or as $\left(U_n \Sigma_n^{\frac{1}{2}} \right) \left(\Sigma_n^{\frac{1}{2}} V_n^T \right)$, and these factors are related as

$$U_n \Sigma_n^{\frac{1}{2}} = \mathcal{O}_s T^{-1} = \mathcal{O}_{s,T} \quad \Sigma_n^{\frac{1}{2}} V_n^T = T \mathcal{C}_{N-s+1} = \mathcal{C}_{N-s+1,T}$$

for $T \in \mathbb{R}^{n \times n}$ a nonsingular transformation. Therefore $\mathcal{O}_{s,T}$ resp. $\mathcal{C}_{N-s+1,T}$ act as the extended observability resp. controllability matrix of a similarly equivalent triplet of system matrices (A_T, B_T, C_T) . This correspondence allows to read-off the system matrices C_T and B_T as the first ℓ rows of the matrix $\mathcal{O}_{s,T}$ and the first m columns of $\mathcal{C}_{N-s+1,T}$ resp. Further the *shift-invariance* property of the extended observability resp. controllability matrices allows to find the system matrix A_T of the minimal realization. For example, consider the extended observability matrix $\mathcal{O}_s$, then the shift-invariance property states that:

$$\mathcal{O}_{s,T}(1 : (s-1)\ell, :) A_T = \mathcal{O}_{s,T}(\ell+1 : s\ell, :) \quad (5)$$

where the notation $M(u : v, :)$ indicates the submatrix of M from rows u to rows v . The shift-invariance property delivers a set of linear equations from which the system matrix A_T can be computed via the solution of a **linear** least-squares problem when $s > n$.

Finding the dimension parameters s (and N) of the Hankel matrix $H_{s,N}$ is a nontrivial problem in general. When only the Markov parameters are given and the knowledge that they stem from a finite-order state-space model, a possible sequential strategy is to select s and N equal to the upper bounds in (4) for presumed orders n and $n+1$, respectively. When the rank of the Hankel matrices for these two selections of s (and N) is identical, the right dimensioning of the Hankel matrix $H_{s,N}$ is found. Otherwise the presumed order is increased by one.

The Stochastic Realization Problem

The output-only identification problem aims at determining a mathematical model from a measured multivariate time series $\{y(k)\}_{k=1}^N$ with $y(k) \in \mathbb{R}^\ell$. Such a model can be then used for predicting future values of the (output) data from past values.

In the vein of the revival of the work of Kronecker on realizing dynamical systems from its impulse response, Faure and a number of contemporaries like Akaike and Aoki made pioneering contributions to extend this methodology to stochastic processes (Van Overschee and De Moor 1993). These extensions are known as solutions to the stochastic realization problem.

This problem is formulated for $y(k)$ to be a Markovian stochastic process. Reusing the notation in (1) $y(k)$ is assumed to be generated by (1) with the input $u(k) \equiv 0$. The A matrix in (1) is again assumed to be stable. The given data in the early formulations of the stochastic realization problem was the samples of the covariance function

$$R_y(j) = E[y(k)y(k-j)^T]$$

These samples define the strictly positive real spectral density function of $y(k)$:

$$\Phi_y(z) = \sum_{j=-\infty}^{\infty} R_y(j)z^{-j} > 0 \quad (6)$$

Given the samples of the covariance function $R_y(j)$, the stochastic realization problem was to find an innovation model representation of the form

$$\begin{aligned} \hat{x}(k+1) &= A_T \hat{x}(k) + K_T e'(k) \\ \tilde{y}(k) &= C_T \hat{x}(k) + e'(k) \end{aligned} \quad (7)$$

with $e'(k)$ a zero-mean, white-noise input with covariance matrix R_e , the pair (A_T, C_T) observable, and A_T stable, such that the spectral density functions $\Phi_y(z)$ and $\Phi_{\tilde{y}}(z)$ are equal.

The partial similarity between this problem and the minimal realization problem becomes clear when expressing the covariance function samples $R_y(j)$ in terms of the system matrices in (1)—for $u(k) \equiv 0$ as

$$R_y(j) = C A^{j-1} G \quad \text{for } j \neq 0 \quad (8)$$

with the matrices G and $R_y(0)$ derived from the following covariance expressions:

$$\begin{aligned} E[x(k)x(k)^T] &= \Sigma_x : \Sigma_x \\ &= A \Sigma_x A^T + K R K^T \end{aligned} \quad (9)$$

$$\begin{aligned} E[x(k+1)y(k)^T] &= G : G \\ &= A \Sigma_x C^T + K R \end{aligned} \quad (10)$$

$$\begin{aligned} E[y(k)y(k)^T] &= R_y(0) : R_y(0) \\ &= C \Sigma_x C^T + R \end{aligned} \quad (11)$$

Since the spectral density has a two-sided series expansion, there is a so-called forward stochastic realization problem (considering $R_y(j)$ for $j \geq 0$ only) and a backward version. Here we only treat the forward one. Drawing the parallel between the samples of the covariance function $R_y(j)$, as given in (6)–(8) and the Markov parameters in (2), we can use the deterministic tools from realization theory to find a minimal realization (A_T, C_T, G_T) .

The *intermediate ST step* in the stochastic realization problem is the construction of a Hankel matrix similar to the matrix $H_{s,N}$ as in the deterministic realization problem but now from the samples of the covariance function $R_y(j)$ in (8).

With the triplet (A_T, C_T, G_T) determined, the innovation model (7) is classically completed via the solution of a Riccati equation in the unknown Σ_x . This Riccati equation results by noting that $R > 0$, and therefore, $K R K^T$ can be written as $K R(R)^{-1} R^T K^T$. This reduces the expression for Σ_x in (9) with the help of (10) and (11) as

$$\begin{aligned} \Sigma_x &= A \Sigma_x A^T + (G - A \Sigma_x C^T)(R_y(0) \\ &\quad - C \Sigma_x C^T)^{-1}(G - A \Sigma_x C^T)^T \end{aligned} \quad (12)$$

By replacing the triplet (A, C, G) with the found minimal realization (A_T, C_T, G_T) in this Riccati equation, its solution $\Sigma_{x,T}$ enables in the end to define the missing quantities as

$$\begin{aligned} R_e &= R_y(0) - C_T \Sigma_{x,T} C_T^T \\ K_T &= (G_T - A_T \Sigma_{x,T} C_T^T) R_e^{-1} \end{aligned} \quad (13)$$

By the positive realness of $\Phi_y(z)$ and the similar equivalence between the triplets (A_T, C_T, G_T) and (A, C, G) , the solution $\Sigma_{x,T}$ is positive definite.

A persistent problem in solving the stochastic realization problem has existed for a long time when using approximate values of the samples $R_y(j)$. This problem is that the estimated power spectrum based on estimates of the triplet (A_T, C_T, G_T) is *no longer positive real*.

An approximate solution overcoming the problem of the loss of positive realness of the estimated power spectrum was provided in the vein of the ST developed in the early 1990s as discussed in the next section.

Combined Deterministic-Stochastic ST

Identification of LTI MIMO Systems in Open Loop

Since the golden 1960s and 1970s of the twentieth century, many attempts have been made to make the insights from deterministic and stochastic realization theory useful for system identification. To mention a few, there are attempts to use the solutions to the deterministic realization problem with measured or estimated impulse response data. One such method is known under the name of the eigensystem realization algorithm (ERA) (Juang and Pappa 1985) and has been used for modal analysis of flexible structures, like bridges, space structures, etc. Although these methods tend to work well in practice for these resonant structures that vibrate (strongly), they did not work well for other types of systems and an input different from an impulse. Extensions to the stochastic realization problem considered the use of finite sample average estimates of the covariance function as an attempt to make the method work with finite data length sequences. As indicated in Section “[The Stochastic Realization Problem](#),” these approximations of the covariance function tended to violate the positive realness property of the underlying power spectrum.

In the early 1990s of the twentieth century, new breakthroughs were made working directly with the input–output data of an assumed LTI system without the need to first compute the Markov parameters or estimating the samples of covariance functions. Pioneers that contributed to these breakthroughs were Van Overschee and De Moor, introducing the N4SID approach (Van Overschee and De Moor 1994); Verhaegen, introducing the MOESP approach (Verhaegen 1994); and Larimore, presenting ST in the framework of canonical variate analysis (CVA) (Larimore 1990).

These three pioneering contributions considered the identification of the state-space model (1) from the input–output data $\{u(k), y(k)\}_{k=1}^N$ recorded in *open loop*. The pair (A, C) was assumed to be observable and the pair (A, KR)

controllable. The innovation noise covariance matrix R was assumed to be positive definite.

The formulation of the *intermediate ST step* from which these three pioneering contributions can be derived (by weighting the result of Theorem 1) and that is at the heart of many more variants is summarized in Theorem 1. This theorem requires two preparations: first the storage of the input and output sequences into (block-) Hankel matrices and relating these Hankel matrices via the model parameters and second to make three observations about the model (1) when presented in the prediction form. This form is obtained by replacing $x(k)$ by $\hat{x}(k)$ and $e(k)$ by $y(k) - C\hat{x}(k) - Du(k)$ and is given by

$$\begin{aligned}\hat{x}(k+1) &= (A - KC)\hat{x}(k) \\ &\quad + (B - KD)u(k) + Ky(k) \\ y(k) &= C\hat{x}(k) + Du(k) + e(k)\end{aligned}\quad (14)$$

To compact the notation, we make the following substitutions: $\mathcal{A} = (A - KC)$ and $\mathcal{B} = [(B - KD) \ K]$.

Let the Hankel matrix with the “future” part $\{y(k)\}_{k=p+1}^N$ be defined as

$$Y_f = \begin{bmatrix} y(p+1) & y(p+2) & \cdots & y(N-f+1) \\ y(p+2) & & & \\ \vdots & & \ddots & \\ y(p+f) & & \cdots & y(N) \end{bmatrix} \quad (15)$$

for the dimensioning parameters p and f selected such that

$$p \geq f > n$$

In a similar way we define the Hankel matrices U_f and E_f from the input $u(k)$ and the innovation $e(k)$, respectively. Then with the definition of the (block-)Toeplitz matrix T_u from the quadruple of system matrices (A, B, C, D) as

$$T_u = \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & & 0 \\ CAB & CB & & 0 \\ & & \ddots & \\ CA^{f-1}B & CA^{f-2}B & \cdots & D \end{bmatrix}$$

and similarly the definition of the Toeplitz matrix T_e from the quadruple of system matrices (A, K, C, I) , we can relate the data Hankel matrices Y_f and U_f as

$$\begin{aligned} Y_f &= O_f [\hat{x}(p+1) \cdots \hat{x}(N-f+1)] \\ &\quad + T_u U_f + T_e E_f \\ &= O_f \hat{X}_f + T_u U_f + T_e E_f \end{aligned} \quad (16)$$

Based on the prediction form (14), 3, key observations are made to support the rational of the intermediate step summarized in Theorem 1:

O1: The standard assumption that the transfer function from $e(k)$ to $y(k)$ is minimum phase leads to the fact that matrix $\mathcal{A}$ is stable. Therefore, there exists a finite integer p such that

$$\mathcal{A}^p \approx 0$$

O2: The state-space model of (14) has inputs $u(k)$ and $y(k)$. Grouping both together into the new vector $z(k) = \begin{bmatrix} u(k) \\ y(k) \end{bmatrix}$ enables to express the state $\hat{x}(k+p)$ as

$$\hat{x}(k+p) = \mathcal{A}^p \hat{x}(k) + \sum_{j=1}^p \mathcal{A}^{j-1} \mathcal{B} z(k+p-j)$$

for $k \geq 1$. With the assumption that $\mathcal{A}^p \approx 0$ and the definition of the input-output data vector sequence $Z(k) = [z(k)^T \cdots z(k+p-1)^T]^T$, we have the following approximation of the state:

$$\hat{x}(k+p) \approx [\mathcal{A}^{p-1} \mathcal{B} \cdots \mathcal{B}] Z(k) = \mathcal{L}^z Z(k)$$

As such the state sequence $\hat{X}_f$ in (16) can be approximated by

$$\mathcal{L}^z Z_p = \mathcal{L}^z [Z(1)Z(2) \cdots Z(N-f-p+1)].$$

O3: The (approximate) knowledge of the row space of the state sequence in $\hat{X}_f$ makes that the unknown system matrices $(\mathcal{A}, \mathcal{B}, C, D, K)$ appear (approximately) linearly in the model (14).

The intermediate ST step to retrieve a matrix with relevant subspaces is summarized in the following theorem taken from Peternell et al. (1996).

Theorem 1 (Peternell et al. 1996) *Consider the model (1) with all stochastic processes assumed to be ergodic and with the input $u(k)$ to be statistically uncorrelated from the innovation $e(\ell)$ for all k, ℓ . Consider the following least-squares problem:*

$$[\hat{L}_N^u \hat{L}_N^z] = \arg \min_{L^u, L^z} \|Y_f - [L^u \ L^z] \begin{bmatrix} U_f \\ Z_p \end{bmatrix}\|_F^2 \quad (17)$$

with $\|\cdot\|_F^2$ denoting the Frobenius norm of a matrix, then

$$\lim_{N \rightarrow \infty} \hat{L}_N^z = \mathcal{O}_f \mathcal{L}^z + \mathcal{O}_f \mathcal{A}^p \Delta_z$$

with Δ_z a bounded matrix.

The theorem delivers the matrix $\hat{L}_N^z$ via the solution of a convex linear least-squares problem that has asymptotically (in the number of measurements N) the extended observability matrix $\mathcal{O}_f$ as its column space and that has asymptotically (in the number of measurements as well as in the dimension parameter p) the matrix $\mathcal{L}^z$ as its row space. Based on the expression of the state sequence $\hat{X}_f$ given in the observation O2 above, the estimate of the row space of $\mathcal{L}^z$ delivers an estimate of the row space of the state sequence $\mathcal{X}_f$. The observation O3 then shows that this intermediate step allows to derive an estimate of the system matrices $[A, B, C, D, K]$ (up to a similarity transformation) via a linear least-squares problem.

Towards Understanding the Statistical Properties

Many ST variants for system identification using data recorded in open loop have been developed since the early 1990s of the twentieth century. These variants mainly differ in the use of weighting matrices $\mathcal{W}_\ell$ and $\mathcal{W}_r$ in the product $\mathcal{W}_\ell \hat{L}_N^z \mathcal{W}_r$ prior to computing the subspaces of interest. The effect on the accuracy and the statistical properties of the estimated model by these weighting matrices is yet not fully understood as is that of the dimensioning parameters p and f in the definition of the data Hankel matrices Y_f, U_f, Z_p . Only for very specific restrictions, results have been achieved. For example, in Bauer and Ljung (2002), it has been shown that when the input $u(k)$ in (1) is either non-present or zero-mean white noise, as well as when the system order n of the underlying system to be known and letting in addition to the dimension parameter p and the number of data points N the dimension parameter f go to infinity, that the weighting matrices selected to represent the CVA approach (Larimore 1990) yield an optimal *minimum variance* estimate. A framework for analyzing the statistical properties like consistency and asymptotic distribution of the estimates determined by the class of STs that were discovered in the 1990s is given in Bauer (2005).

The minimum variance property of the estimates by the CVA approach (Larimore 1990) is theoretically not yet proven for more generic and practically relevant experimental conditions. For these cases, the choices of the different weighting matrices, the dimensioning parameters f, p , as well as selecting the system order are often diverted to user. Despite this fact, practical evidence has shown that STs are able to accurately identify state-space models for LTI MIMO systems under industrially realistic circumstances. As such they are by now accepted and widely used as a common engineering tool in various areas, such as model-based control, fault diagnostics, etc. Further they generally provide excellent initial estimates to the nonlinear parametric optimization methods in prediction error or maximum likelihood estimation methods.

Identification of LTI MIMO Systems in Closed Loop

The least-squares problem (17) in Theorem 1 leads to biased estimates when using input-output data that is recorded in a closed-loop identification experiment. This is because of the correlation between the measurable input and the innovation sequence. A number of solutions have been developed to overcome this problem. We refer to the paper van der Veen et al. (2013) for an overview of a number of these rescues. A simple and performant rescue is described here based on the work in Chiuso (2010). The *intermediate ST step* in order to avoid biased estimates is to estimate a high-order vector autoregressive models with exogenous inputs, a so-called VARX model:

$$\min_{\Theta} \sum_{k=1}^{N-p} \|y(k+p) - \Theta Z(k) - Du(k+p)\|_2^2 \quad (18)$$

Using the result on the approximation of the state vector $\hat{x}(k+p)$ in observation O2, it can be shown that the solution $\hat{\Theta}$ of (18) is an approximation of the parameter vector:

$$\hat{\Theta} = [\widehat{CA^{p-1}B} \dots \widehat{CB}]$$

Then using this solution $\hat{\Theta}$ and O1 above leads to the following “subspace revealing matrix” (cf. Fig. 1):

$$\begin{bmatrix} \widehat{CA^{p-1}B} & \widehat{CA^{p-2}B} & \dots & \widehat{CA^{p-f}B} & \dots & \widehat{CB} \\ 0 & \widehat{CA^{p-1}B} & \dots & \widehat{CA^{p-f+1}B} & \dots & \widehat{CAB} \\ \vdots & & \ddots & & & \\ 0 & 0 & \dots & \widehat{CA^{p-1}B} & \dots & \widehat{CA^{f-1}B} \end{bmatrix} \quad (19)$$

As in the open-loop case of Section “Identification of LTI MIMO Systems in Open Loop,” column and row weighting matrices as well as changing the size of the subspace revealing matrix (19) can be used to influence the accuracy of the estimates (Chiuso 2010). The subspace of interest of this weighted subspace revealing matrix is its row space that is an

approximation of that of the state sequence $\hat{X}_f$ as in (16), now extended to make the size compatible to the weighted version of (19). Similarly as in the open-loop case, knowledge of this subspace turns the estimation of the system matrices $[A, B, C, D, K]$ (up to a similarity transformation) into a linear least-squares problem. The statistical asymptotic properties of this closed-loop ST and the treatment of the dimensioning parameters have also been studied in Chiuso (2010). Here, the result is proven that the asymptotic variance of any system invariant of the model estimated via the above closed-loop ST is a nonincreasing function of the dimensioning parameter f when the input $u(k)$ to the plant is generated by an LQG controller with a white-noise reference input.

Subspace Identification with Prediction Error Criteria

The subspace revealing step of the SI methods outlined in the previous subsection, generally are preceded by (a) linear in the parameters prediction error problem(s). However both problems may be integrated resulting into regularization of the subspace revealing step.

One such regularization that seeks to make a trade-off between a prediction error criterium and the subspace revealing step is described in Verhaegen (1994).

Beyond LTI Systems

The summarized discrete-time ST methodology has been extended in various ways. A number of important extensions including representative papers are towards continuous-time systems (van der Veen et al. 2013), using frequency-domain data (Cauberghe 2006) or for different classes of nonlinear systems, like block-oriented Wiener and/or Hammerstein and linear parameter-varying systems (van Wingerden and Verhaegen 2009). ST for linear time-varying systems with changing dimension of the state vector is treated in Verhaegen and Yu (1995), and finally we mention the developments to make ST recursive (van der Veen et al. 2013).

Summary and Future Directions

Subspace techniques aim at simplifying the system identification cycle and make it more user-friendly. Still a number of challenges persist in improving on this general goal. A critical one is the “optimal” selection of the weighting matrices and the dimensioning parameters p and f of the subspace revealing matrix. Optimality here can be expressed, e.g., by the minimality of the variance of the estimates but could also be viewed more generally in relationship with the use of the model, e.g., in terms of the performance of a model-based closed-loop design. A profound theoretical framework is necessary to fully automate the selection of the weighting matrices and dimensioning and order indices. This would substantially contribute to fully automated identification procedures for doing system identification (for linear systems).

A second challenge is to better integrate ST with robust controller design. This requires the assessment of the model quality and the selection of an optimal input. Particular to the integration of ST to control design is the striking similarity of data equations used in ST and model predictive control. The challenge is to further exploit this similarity to develop data-driven model predictive control methodologies that are robust w.r.t. the identified model uncertainty.

One interesting development in ST is the use of regularization via the nuclear norm in order to improve the model order selection with respect to, e.g., SVD-based ST in Liu and Vandenberghe (2010).

A final challenge is to extend ST for LTI systems to other classes of dynamic systems, such as nonlinear, hybrid, and large-scale systems.

Cross-References

- [Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions](#)
- [Realizations in Linear Systems Theory](#)
- [Sampled-Data Systems](#)
- [System Identification: An Overview](#)

Recommended Reading

The recommended readings for further study are the books Verhaegen and Verdult (2007) and Katayama (2005), the topic of subspace identification is treated in a wider context for classroom teaching at the MSc level since more elaborate topics relevant in the understanding of ST are treated, such as key results from linear algebra, linear least squares, and Kalman filtering. The book Van Overschree and De Moor (1996) is focused on subspace identification only and also emphasizes the success of ST on various applications. All these books provide access to numerical implementations for getting hands-on experience with the methods. The integration of subspace methods with other identification approaches is done in the toolbox (Ljung 2007).

There also exist a number of overview articles. An overview of the early developments of ST since the 1990s of the twentieth century is given in Viberg (1995). Here also the link between ST for identifying dynamical systems and the signal processing application of direction-of-arrival problems was clearly made. A more recent overview article is van der Veen et al. (2013). In this article also reference is made to the statistical analysis and closed-loop application of ST.

Many papers have appeared reporting successful application of subspace methods in practical applications. We refer to the book Van Overschree and De Moor (1996) and the overview paper van der Veen et al. (2013).

Bibliography

- Bauer D (2005) Asymptotic properties of subspace estimators. *Automatica* 41(3):359–376
- Bauer D, Ljung L (2002) Some facts about the choice of the weighting matrices in larimore type of subspace algorithms. *Automatica* 38(5):763–773
- Cauberghe B, Guillaume P, Pintelon R, Verboven P (2006) Frequency-domain subspace identification using {FRF} data from arbitrary signals. *J Sound Vib* 290(3–5):555–571
- Chiuso A (2010) Asymptotic properties of closed-loop cca-type subspace identification. *IEEE-TAC* 55(3):634–649
- Jansson M, Wahlberg B (1996) A linear regression approach to state-space subspace systems. *Signal Process* 52:103–129
- Juang J-N, Pappa RS (1985) Approximate linear realizations of given dimension via Ho's algorithm. *J Guid Control Dyn* 8(5):620–627
- Katayama T (2005) *Subspace methods for system identification*. Springer, London
- Kronecker L (1890) *Algebraische reduktion der schaaren bilinearer formen*. S.B. Akad. Berlin, pp 663–776
- Larimore W (1990) Canonical variate analysis in identification, filtering, and adaptive control. In: *Proceedings of the 29th IEEE conference on decision and control*, 1990, Honolulu, vol 2, pp 596–604
- Liu Z, Vandenberghe L (2010) Interior-point method for nuclear norm approximation with application to system identification. *SIAM J Matrix Anal Appl* 31(3):1235–1256
- Ljung L (2007) *The system identification toolbox: the manual*. The MathWorks Inc., Natick. 1st edition 1986, 7th edition 2007
- Peternell K, Scherrer W, Deistler M (1996) Statistical analysis of novel subspace identification methods. *Signal Process* 52(2):161–177
- Schutter BD (2000) Minimal state space realization in linear system theory: an overview. *J Comput Appl Math* 121(1–2):331–354
- van der Veen GJ, van Wingerden JW, Bergamasco M, Lovera M, Verhaegen M (2013) Closed-loop subspace identification methods: an overview. *IET Control Theory Appl* 7(10):1339–1358
- Van Overschree P, De Moor B (1993) Subspace algorithms for the stochastic identification problem. *Automatica* 29(3):649–660
- Van Overschree P, De Moor B (1994) N4sid: subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica* 30(1):75–93
- Van Overschree P, De Moor B (1995) A unifying theorem for three subspace system identification algorithms. *Automatica* 31(12):1853–1864
- Van Overschree P, De Moor B (1996) *Identification for linear systems: theory – implementation – applications*. Kluwer Academic Publisher Group, Dordrecht
- van Wingerden J, Verhaegen M (2009) Subspace identification of bilinear and LPV systems for open and closed loop data. *Automatica* 45(2):372–381
- Verhaegen M (1994) Identification of the deterministic part of mimo state space models given in innovations form from input–output data. *Automatica* 30(1):61–74
- Verhaegen M, Verdult V (2007) *Filtering and identification: a least squares approach*. Cambridge University Press, Cambridge/New York
- Verhaegen M, Yu X (1995) A class of subspace model identification algorithms to identify periodically and arbitrarily time-varying systems. *Automatica* 31(2):201–216
- Viberg M (1995) Subspace-based methods for the identification of linear time-invariant systems. *Automatica* 31(12):1835–1851

Supervisory Control of Discrete-Event Systems

Kai Cai¹ and W.M. Wonham²

¹Department of Electrical and Information Engineering, Osaka City University, Osaka, Japan

²Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada

Abstract

We introduce background and base model for supervisory control of discrete-event systems, followed by discussion of optimal controller existence, a small example, distributed control through localization, and summary of control under partial observations. Control architecture and symbolic computation are noted as approaches to manage state space explosion.

Keywords

Asynchronous · Control architectures · Controllability · Discrete · Dynamics · Finite automata · Observability · Optimality · Regular languages · Symbolic computation

Introduction

Discrete-event (dynamic) systems (DES or DEDS) constitute a relatively new area of control science and engineering, which has taken its place in the mainstream of control research. Recently, DES have been combined with continuous systems in areas called hybrid or cyber-physical systems.

Problems and methods for DES have been investigated for some time, although not necessarily with a “control” flavor. The parent domains can be identified as operations research and software engineering.

Operations research deals with systems of interconnected stores and servers which operate

on processed items. For instance, manufacturing systems employ queues, buffers, and bins (which store workpieces). These are served by machines, robots, and automatic guided vehicles (AGVs), which process workpieces. The main problems are to measure quantitative performance and establish trade-offs, for instance, workflow vs. cost, and to optimize design parameters such as buffer size and maintenance frequency.

The relevant areas of software engineering include operating systems control, concurrent computing, and real-time (embedded or reactive) systems, with focus on synchronization algorithms that enforce mutual exclusion and resource sharing in the presence of concurrency, as in the classical problems of “readers and writers” and “dining philosophers.” The main objectives are (i) to guarantee safety (“nothing bad will ever happen”), as in mutual exclusion and deadlock prevention, and (ii) to guarantee liveness (“something good will happen eventually”), for instance, successful computational termination and eventual access to a desired resource.

DES from a Control Viewpoint

With these domains in mind, we consider DES from a control viewpoint. In general, control deals with dynamical systems, defined as entities consisting of an internal state space, together with a state evolution or transition structure, and equipped (for control purposes) with both an input mechanism for actuation and an output channel for observation and feedback. The objective of control is to bring together information and dynamics in some purposeful combination: the interplay between observation and control or decision-making is fundamental.

In this framework, a DES is a dynamical system that is discrete, in time and usually in state space; is asynchronous or event driven, that is, driven by events or instantaneous happenings in time (which may or may not include the tick of a clock); and is nondeterministic, namely, embodies internal chance or other unmodeled mechanisms of choice which govern its state transitions. With a manufacturing system, for

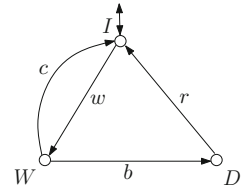
example, the dynamic state might include the status of machines (idle, working, down, under maintenance or repair), the contents of queues and buffers, and the locations and loads of robots and AGVs, while transitions (discrete events) occur when queues and buffers are incremented or decremented, robots load or unload, and machines start work, finish work, or break down (the “choice” between finishing work successfully and breaking down being thus nondeterministic). In this example and many others, the objectives of design and analysis include logical correctness in the presence of concurrency and timing constraints and quantitative performance such as rates of production, all of which depend crucially on feedback control synthesis and optimization. To this end the models will tend to be DES or hybrid systems. Nevertheless one finds the continuing relevance of standard control-theoretic concepts like feedback, stability, controllability, and observability, along with their roles in large-system architectures embodying hierarchical, decentralized, and distributed functional organization.

Here we focus on models and problems from which explicit constraints of timing are absent and which can be considered in a framework of finite-state machines and the corresponding regular languages. While the theory has been generalized to more flexible and technically advanced settings, our restricted framework is already rich enough to support numerous applications and remains challenging for large systems of industrial size.

Base Model for Control of DES

The formal structure of a DES to be controlled will resemble the simple “machine” called **MACH** shown in Fig. 1. The state set of **MACH** is $Q = \{I, W, D\}$, interpreted as Idle, Working, or Broken Down. **MACH** is initialized at state $q_o = I$, denoted by an entering arrow without source. The transition structure is displayed in Fig. 1 as a transition diagram, whose nodes are the states $q \in Q$ and edges are the transitions,

Supervisory Control of Discrete-Event Systems,
Fig. 1 MACH



each labeled with a symbol σ in the alphabet Σ , here $\{w, c, b, r\}$. If a transition (labeled) σ is an edge from q to q' , then “the event σ can occur at state q , and when σ occurs, q transits to state q' .” Transitions (or events) are interpreted as instantaneous in time, while states are thought of as locations where **MACH** is able to reside for some indeterminate time interval. The occurrence of w means “**MACH** enters the Working state from Idle” and similarly for c, b, r . These transitions determine the state-transition function of **MACH**, denoted by $\delta : Q \times \Sigma \rightarrow Q$. Thus $\delta(I, w) = W$, $\delta(W, b) = D$, and so on. Notice that δ is a partial function, defined at each state $q \in Q$ for only a subset of event (labels) in Σ . To denote that $\delta(q, \sigma)$ is defined at state $q \in Q$ for the event $\sigma \in \Sigma$, we write $\delta(q, \sigma)!$. The function δ can be extended by iteration to $\delta : Q \times \Sigma^* \rightarrow Q$, where Σ^* is the set of all finite strings of elements of Σ , including the empty string ϵ . Thus $\delta(q, \epsilon) := q$ and inductively if $q' := \delta(q, s)!$, then

$$\delta(q, s.\sigma) := \delta(\delta(q, s), \sigma) := \delta(q', \sigma)$$

whenever $\delta(q', \sigma)!$. Graphically the strings $s = \sigma_1 \dots \sigma_k \in \Sigma^*$ for which $\delta(q, s)!$ are precisely those for which there exists a path in the transition diagram starting from q and having successive edges labeled $\sigma_1, \dots, \sigma_k$.

We call any subset of Σ^* (i.e., any set of strings of elements from Σ) a language over Σ and accordingly speak of sublanguages of a language over Σ .

For **MACH**, the execution of a production cycle, namely, the event sequence (or string) $w.c$, or a work-breakdown-repair cycle, the string $w.b.r$, can be considered successful, and the corresponding string is said to be *marked*. States which are entered by marked strings are marked

states and identified in a transition diagram by an outgoing arrow with no target. In Fig. 1, the only marked state happens to be the initial state, which is thus shown with a double arrow; in general there could be several marked states, which may or may not include the initial state. The marked states comprise a subset $Q_m \subseteq Q$, which may be empty (at one extreme) or equal to Q (at the other). The case $Q_m = Q$ (all states marked) would imply that every string of events is considered as significant or successful as any other, while the case $Q_m = \emptyset$ (no state marked, so there are no successful strings) plays a technical role in computation.

In general a DES is a tuple $\mathbf{G} = (Q, \Sigma, \delta, q_o, Q_m)$ usually interpreted physically as for **MACH** above, but mathematically consisting merely of the finite state set Q ; finite alphabet Σ ; marked subset $Q_m \subseteq Q$, with initial state $q_o \in Q$; and (partial) transition function $\delta : Q \times \Sigma \rightarrow Q$. Additionally we bring in the closed behavior $L(\mathbf{G})$ of $\mathbf{G}$, defined as all the strings of Σ^* which $\mathbf{G}$ can generate starting from the initial state, in the sense:

$$L(\mathbf{G}) := \{s \in \Sigma^* \mid \delta(q_o, s)!\}.$$

Of central importance also is the marked behavior of $\mathbf{G}$, namely, the sublanguage of $L(\mathbf{G})$ given by

$$L_m(\mathbf{G}) := \{s \in L(\mathbf{G}) \mid \delta(q_o, s) \in Q_m\}.$$

We need several definitions. A string s' is a *prefix* of a string $s \in \Sigma^*$, written $s' \leq s$, if s' can be extended to s , namely, there exists a string w in Σ^* such that $s'.w = s$. The *closure* of a language $M \subseteq \Sigma^*$ is the language $\overline{M}$ consisting of all prefixes of strings in M

$$\overline{M} := \{s' \in \Sigma^* \mid s' \leq s \text{ for some } s \text{ in } M\}$$

A language N over Σ is (prefix-)closed if it contains all its prefixes, namely, $N = \overline{N}$. In this notation $\mathbf{G}$ is said to be *nonblocking* if $L(\mathbf{G}) = \overline{L_m(\mathbf{G})}$, namely, any (generated) string in $L(\mathbf{G})$ is a prefix of, and so can be extended to, a marked string of $\mathbf{G}$.

The semantics of $\mathbf{G}$ (its mathematical meaning) is simply the pair of languages $L_m(\mathbf{G})$, $L(\mathbf{G})$. In general the latter may be infinite subsets of Σ^* , while $\mathbf{G}$ itself is a finite object, considered to represent an algorithm for the generation of its behaviors. Unless $\mathbf{G}$ is trivial (has empty state set), it is always true that (the empty string) $\epsilon \in L(\mathbf{G})$.

Transition labeling of $\mathbf{G}$ is deterministic: at every q , at most one transition is defined for each given event σ , namely,

$$\delta(q, \sigma) = q' \text{ \& } \delta(q, \sigma) = q'' \text{ implies } q' = q''.$$

It is quite acceptable, however, that at distinct states q and r , both $\delta(q, \sigma)!$ and $\delta(r, \sigma)!$ (where these evaluations may or may not be equal).

To formulate a control problem for $\mathbf{G}$, we first adjoin a control technology or mechanism by which $\mathbf{G}$ may be actuated to affect its temporal behavior, namely, determine the strings it is permitted to generate. To this end we assume that a subset of events $\Sigma_c \subseteq \Sigma$, called the *controllable* events, are capable of being enabled or disabled by an external controller. Think of a traffic light being turned green or red to allow or prohibit passage (vehicle transition) through an intersection. The complementary event subset $\Sigma_u := \Sigma - \Sigma_c$ is *uncontrollable*; events $\sigma \in \Sigma_u$ cannot be externally disabled but may be considered permanently enabled. For $\mathbf{G} = \mathbf{MACH}$ one might reasonably assume $\Sigma_c = \{w, r\}$, $\Sigma_u = \{c, b\}$. At a given state q of $\mathbf{G}$, it will be true in general that $\delta(q, \sigma)!$ both for some (controllable) events $\sigma \in \Sigma_c$ and for some (uncontrollable) events $\sigma \in \Sigma_u$. Among the $\sigma \in \Sigma_c$, at a given time, some may be externally enabled and others disabled. So, $\mathbf{G}$ will nondeterministically choose its next generated event from the subset

$$\{\sigma \in \Sigma_u \mid \delta(q, \sigma)!\} \cup \{\sigma \in \Sigma_c \mid \delta(q, \sigma)! \text{ \& } \sigma \text{ is externally enabled}\} \quad (1)$$

We formalize external enablement by a *supervisory control function* $V : L(\mathbf{G}) \rightarrow \text{Pwr}(\Sigma)$, where $\text{Pwr}(\cdot)$ stands for power set. For $s \in L(\mathbf{G})$, the evaluation $V(s)$ is defined to be the

event subset

$$V(s) := \Sigma_u \cup \{\sigma \in \Sigma_c \mid \sigma \text{ is externally enabled following } s\} \quad (2)$$

In other words, the set (1) is expressible as

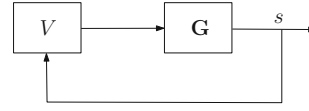
$$V(s) \cap \{\sigma \in \Sigma \mid s.\sigma \in L(\mathbf{G})\} \quad (3)$$

namely, the subset of events that, immediately following the generation of s by $\mathbf{G}$, are either enabled by default (executable events in Σ_u) or else by the external controller's decision (a subset of executable events in Σ_c).

It is now easy to visualize how the generating action of $\mathbf{G}$ is restricted by the action of $V(\cdot)$. Initially (having generated the empty string) $\mathbf{G}$ chooses $\sigma_1 \in V(\epsilon) \cap L(\mathbf{G})$. Proceeding inductively, after $\mathbf{G}$ has generated $s = \sigma_1.\sigma_2 \dots \sigma_k \in L(\mathbf{G})$, s is fed back to the controller, which evaluates $V(s)$ according to (2), announcing the result to $\mathbf{G}$, which then chooses σ_{k+1} in (3), and the process repeats. Of course the process would terminate any time the set (3) happened to become empty (although it need not). In any case, we denote the subset of $L(\mathbf{G})$ so determined as $L(V/\mathbf{G})$, called the *closed behavior* of $V/\mathbf{G}$, where the latter symbol (formally undefined) stands for $\mathbf{G}$ under the supervision of V . It is clear that supervision is a feedback process (Fig. 2), inasmuch as the choice of σ_{k+1} in (3) is not, in general, known in advance, hence must be executed before the succeeding evaluation $V(s.\sigma_{k+1})$ can allow the generating process to continue. With the closed behavior of $V/\mathbf{G}$ now determined, we define the *marked behavior*

$$L_m(V/\mathbf{G}) := L(V/\mathbf{G}) \cap L_m(\mathbf{G}) \quad (4)$$

namely, those marked strings of $\mathbf{G}$ that survive under supervision by V . Thus supervisory control is nonblocking if $L(V/\mathbf{G}) = \overline{L_m(V/\mathbf{G})}$.



Supervisory Control of Discrete-Event Systems,
Fig. 2 Feedback loop $V/\mathbf{G}$

Existence of Controls for DES: Controllability

Of fundamental interest is the question: what sublanguages of $L(\mathbf{G})$ qualify as a language $L(V/\mathbf{G})$ for some choice of supervisory control function V ? In other words, what is the scope of controlled behavior(s) for a given $\mathbf{G}$? So far we know that $L(V/\mathbf{G})$ is a sublanguage of $L(\mathbf{G})$, but it is not usually the case that an arbitrary sublanguage would qualify. For instance, the empty string language $\{\epsilon\} \neq L(V/\mathbf{G})$ for any V as in (2) above, in case $\delta(q_0, \sigma)!$ for some σ in Σ_u , for such σ cannot be disabled.

Assume $\mathbf{G}$ is equipped with the technology of controllable events, hence uncontrollable events $\Sigma_u \subseteq \Sigma$. We make the basic definition: the language $K \subseteq \Sigma^*$ is *controllable* (with respect to $\mathbf{G}$) provided

$$\begin{aligned} &\text{For all } s \in \overline{K} \text{ and for all } \sigma \in \Sigma_u, \\ &\text{whenever } s.\sigma \in L(\mathbf{G}) \text{ then } s.\sigma \in \overline{K}. \end{aligned} \quad (5)$$

Informally, a string s can never exit from $\overline{K}$ as the result of the execution by $\mathbf{G}$ of an uncontrollable event: $\overline{K}$ is invariant under the uncontrollable flow. In terms of $\mathbf{G} = \mathbf{MACH}$, above, the languages $\{\epsilon\}$, $\{w.b, w.c\}$ are controllable, but $\{w\}$, $\{w, w.c.w\}$ are not. For instance, $H := \{w, w.c.w\}$ has closure $\overline{H} = \{\epsilon, w, w.c, w.c.w\}$, which contains the string $s := w$, but $s.b = w.b$ can be executed in $\mathbf{MACH}$, b is uncontrollable, and $s.b$ has exited from $\overline{H}$. It is logically trivial from (5) that the empty language $\emptyset$ (with no strings whatever) is controllable.

We can now answer the fundamental question posed above.

Given a nonempty sublanguage $K \subseteq L(\mathbf{G})$,
there exists a supervisory control function V
such that $\overline{K} = L(V/\mathbf{G})$, if and only if K
is controllable. (6)

This result exhibits the $L(V/\mathbf{G})$ property in a structured way; furthermore, both the containment $K \subseteq L(\mathbf{G})$ and the controllability property (5) (or its absence) can be effectively (algorithmically) decided in case K itself is the closed or marked behavior of some given DES over Σ .

A key fact easily provable from (5) is that the family of all controllable languages (with respect to a fixed $\mathbf{G}$) is algebraically closed under union, namely,

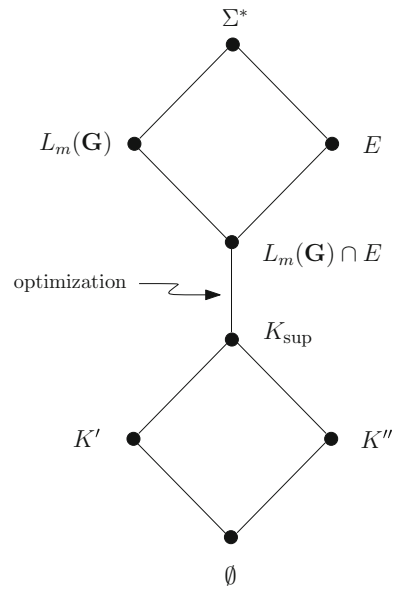
If K_1 and K_2 are controllable languages,
then so is $K_1 \cup K_2$. (7)

In fact (7) can be extended to an arbitrary finite or infinite union of controllable languages.

Given $\mathbf{G}$ as above, considered as the plant to be controlled, suppose a new (regular) language E is specified, as the maximal set of strings that we are prepared to tolerate for generation by $\mathbf{G}$; for instance, E could be considered the legal language for $\mathbf{G}$ (irrespective of what $\mathbf{G}$ is potentially capable of generating, namely, $L(\mathbf{G})$). Let us confine attention to the sublanguage of E that contains only marked strings of $\mathbf{G}$, namely, $E \cap L_m(\mathbf{G})$. We now bring in the family $\mathcal{C}(E \cap L_m(\mathbf{G}))$ of all controllable sublanguages of $E \cap L_m(\mathbf{G})$ (including the empty language). From (7) and its infinite extension, there follows the existence of the controllable language

$$K_{\text{sup}} := \cup \{K \mid K \in \mathcal{C}(E \cap L_m(\mathbf{G}))\} \quad (8)$$

We have $K_{\text{sup}} \subseteq E \cap L_m(\mathbf{G})$, and clearly if K' is controllable and $K' \subseteq E \cap L_m(\mathbf{G})$, then $K' \subseteq K_{\text{sup}}$. K_{sup} is therefore the supremal (largest) controllable sublanguage of $E \cap L_m(\mathbf{G})$. Furthermore, if K_{sup} is nonempty, then by (6) there exists a supervisory control V such that $K_{\text{sup}} = L(V/\mathbf{G})$; in this sense V is optimal (maximally

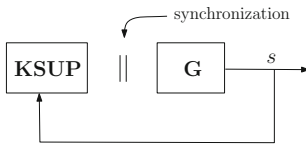


Supervisory Control of Discrete-Event Systems, Fig. 3 Hasse diagram

permissive), allowing the generation by $\mathbf{G}$ of the largest possible set of marked strings that the designer considers legal. We have thus established abstractly the existence and uniqueness of an optimal control for given $\mathbf{G}$ and E . This simple conceptual picture is displayed (Fig. 3) as a Hasse diagram, in which nodes represent sublanguages of Σ^* and rising lines (edges) the relation of sublanguage containment.

In a Hasse diagram it could be that K_{sup} collapses to the empty language $\emptyset$. This means that there is no supervisory control for the problem considered, either because the specifications are too severe and the problem is over-constrained or because the control technology is inadequate (more events need to be controllable).

Under the finite-state assumption, K_{sup} is effectively representable by a DES **KSUP**, which may serve as the optimal feedback controller, as displayed in Fig. 4. Here a string s generated by $\mathbf{G}$ drives **KSUP**; at each state of **KSUP**, the events defined in its transition structure are exactly those available to $\mathbf{G}$ for nondeterministic execution (in its corresponding state) at the next step of the process. In this way the feedback control process



Supervisory Control of Discrete-Event Systems,
Fig. 4 Implementation of V/G

is inductively well defined. The computational complexity of this design (cf. (8)) is $O(|E|^2 \cdot |G|^2)$ where E is a DES with $L_m(E) = E$ and $|\cdot|$ denotes state size. The controller state size is $|KSUP| \leq |E| \cdot |G|$, the product bound being of typical order.

Supervisory Control Design: Small Factory

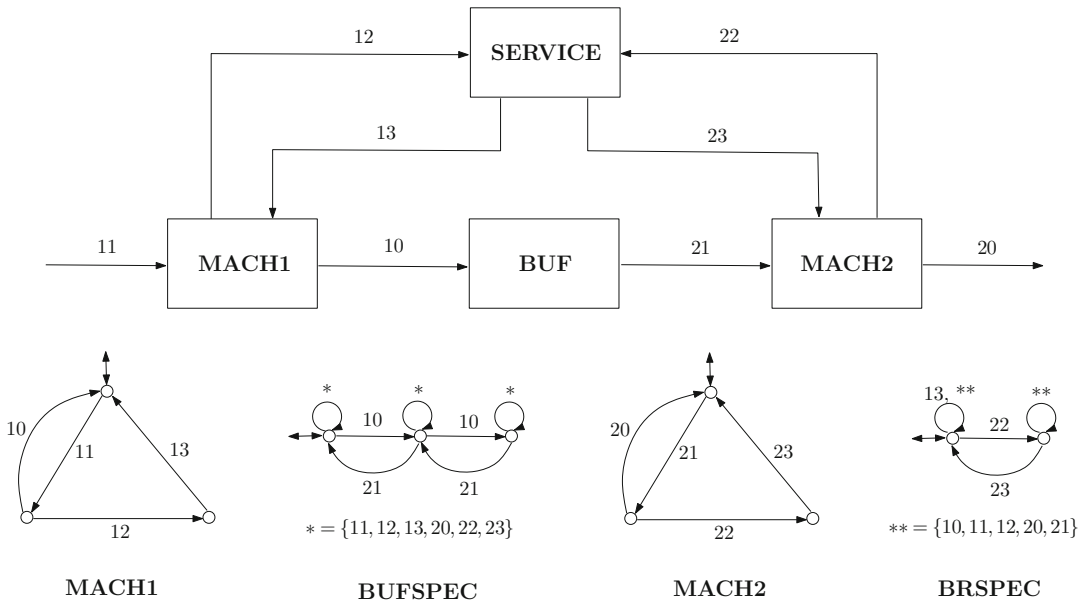
The following example, Small Factory (SF), is an illustration of supervisor design. As in Fig. 5, SF consists of two machines **MACH1** and **MACH2** each similar to **MACH** above, connected by a buffer **BUF** of capacity 2. In case of breakdown the machines can be repaired by a **SERVICE** facility as shown. Transition structures of the machines and design specifications are also displayed in Fig. 5. Σ_c (Σ_u) are odd (even)-numbered events. When self-looped with all irrelevant events to form specification **BUFSPEC**, the latter declares that the machines must be controlled in such a way that **BUF** is not overflowed (an attempt by **MACH1** to deposit a workpiece in **BUF** when it is full) or subject to underflow (an attempt by **MACH2** to take a workpiece from **BUF** when it is empty). In addition, **SERVICE** must enforce priority of repair for **MACH2**: when the latter is down, repair of **MACH1** (if in progress) must be interrupted and only resumed after **MACH2** has been repaired; this logic is expressed by **BRSPEC**. To form the plant model **G** for the DES to be controlled, we compute the synchronous product of **MACH1** and **MACH2**. The result, say $G = \text{FACT}$, is a DES of which the components **MACHi** are free to execute their events independently except for synchronization on events that are shared (here,

none). Similarly we form the synchronous product of **BUFSPEC** and **BRSPEC** to obtain the full specification **DES SPEC**. We now execute the optimization step in the Hasse diagram (Fig. 3); this yields the SF controller **KSUP**(21,47) with 21 states and 47 transitions. Online synchronization of **KSUP** with **FACT** will result in generation of the optimal controlled behavior K_{sup} by the feedback loop. Since $K_{\text{sup}} \subseteq L_m(G)$ by (8), our marking conventions ensure that **KSUP** is nonblocking.

In general the language K_{sup} will include in its structure not only the constraints required by control but also the physical constraints enforced by the plant structure itself (here, **FACT**). The latter are thus redundant in the online synchronization of the plant with the controller **KSUP**. A more economical controller is obtained if the plant constraints are projected out of **KSUP** to obtain a reduced controller, say **KSIM**. Mathematically, projection amounts to constructing a control congruence or dynamically (and control) consistent partition on the state set of **KSUP** and taking the cells of this partition, abstractly, as the new states for **KSIM**. In SF **KSUP**(21,47) is reduced to **KSIM**(5,18), which when synchronized with **FACT** yields exactly **KSUP** but is less than one-quarter the state size. In practice a state size reduction factor of ten or more is not uncommon.

Distributed Control by Supervisor Localization

The projection technique above that yields a reduced controller can be extended further to create *local controllers* for individual component DES. In Small Factory, to create a local controller for machine **MACH1**, not only the constraints imposed by the plant structure but also those by control of machine **MACH2** are redundant or irrelevant. Projecting these constraints out of the monolithic controller **KSUP** generates a controller dedicated solely to control of machine **MACH1** and in this sense is "local." A symmetric projection generates another local controller for **MACH2**. This procedure of creating local controllers by decomposing the monolithic



Supervisory Control of Discrete-Event Systems, Fig. 5 Small Factory

controller is called *supervisor localization*. It can be generalized as well to the case of multicomponent DES and guarantees that the local controllers when synchronized with the plant yield exactly the monolithic controller. Hence, equipped with its own private controller, each component DES may act autonomously without centralized supervision; meanwhile, the collective local controlled behavior is ensured to be the same as that achieved by the monolithic controller. This arrangement with only local controllers is called (purely) *distributed* control architecture; it is useful for systems comprised of multiple autonomous components, such as networked robots, vehicles, or sensors.

Typically, local controllers need to communicate certain events to one another to achieve critical synchronization. The identity between local controlled behavior and global controlled behavior is based on the assumption that event communication is instantaneous. In practice, however, communication is through physical channels, which are subject to (greater or less) time delay. Consequently, when a local controller sends an event σ to another local controller, the latter can receive σ only after some delay. This communication delay may be reflected by

a channel model that treats σ as sending the event, but a distinct σ' as receiving the event, such that σ' occurs after σ with (variable but unbounded) delay. With this channel model, one may test if the local controllers are *robust* to communication delay, in the sense that the channeled behavior (including σ') is *complete* and *correct* with respect to the original, idealized, zero-delay controlled behavior. Completeness means that every zero-delay behavior is some projected channeled behavior (where the delayed event σ' is projected to ϵ), while correctness means that every projected channeled behavior is some zero-delay behavior.

Supervisor Architecture and Computation

As noted earlier, the state size $|\mathbf{KSUP}|$ of controller **KSUP** is on the order of the product of state sizes of the plant, $|\mathbf{G}|$, and specification, $|\mathbf{E}|$. As these in turn are the synchronous products of individual plant components or partial specifications, $|\mathbf{KSUP}|$ tends to increase exponentially with the numbers of plant components and

specifications, the phenomenon of exponential state space explosion. The result is that centralized or monolithic controllers such as **KSUP** can easily reach astronomical state sizes in realistic industrial models, thereby becoming infeasible in terms of computer storage for practical design. This issue can be addressed in two basic ways: by decentralized and hierarchical architectures, possibly in heterarchical combination, and by symbolic DES representation and computation, where what is stored are not DES and their controller transition structures in extensional (explicit) form, but instead intensional or algorithmic recipes from which the required state and control variable evaluations are computed online only when actually needed.

Supervisory Control Under Partial Observations

For a DES G over alphabet Σ , let $\Sigma_o \subseteq \Sigma$ be a subalphabet interpreted as the events that can be viewed by some external observer. A mapping $P : \Sigma^* \rightarrow \Sigma_o^*$ is called a *natural projection* if its action is simply to erase from a string s in Σ^* all the events in s (if any) that do not belong to Σ_o while preserving the order of events in Σ_o . By use of P it is possible to carry over to DES the control-theoretic concept of observability. Two strings $s, s' \in \Sigma^*$ are *look-alikes* with respect to P if $Ps = Ps'$, namely, are indistinguishable to an observer (or channel) modeled by P . Thus, given G and P as above, a sublanguage $K \subseteq L(G)$ is *observable* if, roughly, look-alike strings in $\bar{K}$ have the same one-step extensions in $\bar{K}$ that are compatible with membership in $L(G)$ and also satisfy a consistency condition with respect to membership in $L_m(G)$. For control under observations through P , one defines a supervisory control function $V : L(G) \rightarrow Pwr(\Sigma)$ to be *feasible* if it assumes the same value on look-alike strings, in other words respects the observation constraint enforced by P . It then turns out that a language $K \subseteq L_m(G)$ can be feasibly synthesized in a feedback loop including G and the feedback

channel P if and only if K is both controllable and observable.

Although this result is conceptually satisfying, it is computationally inconvenient because, by contrast with controllability, the property of sublanguage observability is not in general closed under union. A substitute for observability is sublanguage *normality*, a property stronger than observability but one that is indeed closed under union. Since the family of controllable and normal sublanguages of a given specification language is nonempty (the empty language belongs) and is closed under union, a (unique) supremal (or optimal) element exists and can be computed; it therefore solves the problem of supervisory control under partial observations, albeit under the normality restriction. The latter has the feature that the resulting supervisor can disable a controllable event only if the latter is observable, i.e., belongs to Σ_o . In some applications this restriction might preclude the existence of a solution altogether; in others it could be harmless, or even desirable as a safety property, in that if the intended disablement of a controllable event happened to fail, and the event occurred after all, the fault would necessarily be observable and thus optimistically remediable in good time.

An intermediate property is known that is weaker than normality but stronger than observability, called *relative observability*. The family of relatively observable sublanguages of a given specification language is closed under union and thus does possess a supremal element, which in the regular case can be effectively computed. When combined with controllability, relative observability yields a solution to the problem of supervisory control under partial observations which places no limitation on the disablement of unobservable controllable events. Examples show that a nontrivial solution of this type may exist in cases where the normality solution is empty.

Summary and Future Directions

Supervisory control of discrete-event systems, while relatively new, has reached a first level

of maturity in that it is soundly based in a standard framework of (especially) finite-state machines and regular languages. It has effectively incorporated its own versions of control-theoretic concepts like stability (in the sense of nonblocking), controllability, observability, and optimality (in the sense of maximal permissiveness). Modular architectures and, on the computational side, symbolic approaches enable design of both monolithic and heterarchical/distributed controllers for DES models of industrial size. Major challenges remain, especially to develop criteria by which competing architectures can be meaningfully compared and to organize control functionality in ways that are not only tractable but also transparent to the human user and designer.

Cross-References

- [Applications of Discrete Event Systems](#)
- [Models for Discrete Event Systems: An Overview](#)

Bibliography

- Cai K, Wonham WM (2016) Supervisor localization: a top-down approach to distributed control of discrete-event systems. Lecture notes in control and information sciences, vol 459. Springer International Publishing
- Cassandras CG, LaFortune S (2008) Introduction to discrete event systems. Springer, New York
- Cieslak R, Desclaux C, Fawaz A, Varaiya P (1988) Supervisory control of discrete-event processes with partial observations. *IEEE Trans Autom Control* 33:249–260
- Lin F, Wonham WM (1988) On observability of discrete-event systems. *Inf Sci* 44:173–198
- Ma C, Wonham WM (2005) Nonblocking supervisory control of state tree structures. Lecture notes in control and information sciences, vol 317. Springer, Berlin
- Ramadge PJ, Wonham WM (1987) Supervisory control of a class of discrete event processes. *SIAM J Control Optim* 25:206–230
- Seatzu C et al (ed) (2013) Control of discrete-event systems. Springer, London/New York
- Wonham WM, Cai K, Rudie K (2018) Supervisory control of discrete-event systems: a brief history. *Ann Rev Control* 45:250–256
- Wonham WM, Cai K (2019) Supervisory control of discrete-event systems. Communications and Control Engineering, Springer International Publishing

Surgical Robotics

Fanny Ficuciello

Department of Electrical Engineering and Information Technology, Università degli Studi di Napoli Federico II, Napoli, Italy

Abstract

Surgical robotics has not yet reached a level of maturity in terms of system autonomy. Autonomous control methods are not sufficiently trusted because of safety-critical and high-consequence tasks to perform. On the other hand, remote teleoperation of surgical robotic systems imposes extreme cognitive loading to the surgeon, causing severe fatigue and, consequently, a progressive degeneration in performance. This entry discusses the recent advancements and future expectations of automation in surgical robotics.

Keywords

Surgical robotics · Minimally invasive surgery · Microsurgery · Shared autonomy · Autonomous control

Introduction

Twenty years after its introduction, robotic surgery has become a reality accepted worldwide in different fields, but it is still confined to the role of robot assistance, therefore limiting its potential benefit. In robot-assisted surgery (RAS), one usually operates in a master-slave control mode, so that the behavior of surgical robots unfolds under the surgeon's hands-on supervision and real-time overriding authority.

Thirty years ago the introduction of concept of minimally invasive therapy represented a real revolution in the surgical field. Acquiring knowledge in endoscopic techniques, surgeons became able to perform complex procedures through small cannulas. The current trend is nowadays represented by the use of natural orifices, combined

with local excision surgery, as screening programs allow a diagnosis at an early stage in a growing number of cases. Despite growing indications, advanced minimally invasive surgery (MIS) and microsurgery recognize the remaining significant challenges regarding retraction, suturing, ergonomics, and control of the operative field. MIS and microsurgery are driven by endoscope or microscope imaging. To this date the entire manipulation and control of robotic surgical instruments is carried out manually, resulting in a significant workload for the practicing surgeon, operating in a limited workspace. The reason is that endoscopic stereo vision is quite limited for environment awareness, and thus a robotic system still needs to be fully controlled by the surgeon.

Besides technological limitations there are also other important reasons, namely, ethical themes related to the increasing machine authority and ensuing threats to surgeon–patient trust and surgeon deskilling. It is widely debated how autonomous robots give rise to novel ethical issues concerning responsibility ascription for their wrongdoings and moral constraints to impose on their autonomous behavior.

Robotic Autonomy in Surgery

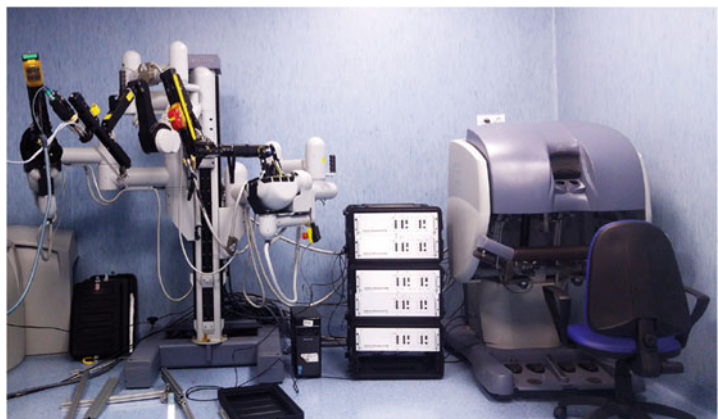
In the master–slave control mode, robotic assistance usually improves surgical procedures by scaling motion, attenuating tremor, and enhancing precision. This is the current

standard use of the da Vinci system (<https://intuitivesurgical.com/products/da-vinci-xi/>)

for robotic surgery, which is controlled in telemanipulation, so that the surgeon has direct control over each procedure. To enhance technology-oriented research for robot-aided surgery, Intuitive Surgical (<https://www.intuitive.com/en-us>) distributed a platform of the first generation of the da Vinci system, which can be used for research purposes only. The research platform, called da Vinci research kit (dVRK) (<http://research.intusurg.com/dvrkwiki>) (Fig. 1), is available in more than 30 research labs across the world. It is an open-source mechatronic platform provided with controllers and software developed at Johns Hopkins University LCSR and Worcester Polytechnic Institute AIM Lab (Kazanzides et al. 2014; Fontanelli et al. 2017). The research kit is suitable to enhance research in both haptic teleoperation and semi-autonomous control, and it is also a valuable tool for safely testing out new technology, e.g., new surgical tools and sensors (Fontanelli et al. 2018, 2019, 2020; Selvaggio et al. 2019a).

Even though teleoperation is the primary control mode (Hirche and Buss 2012), sensor-based shared control of robot motions has been extensively investigated and successfully used to augment surgeon abilities. In addition to da Vinci and similar robotic systems basically working in teleoperation by providing only limited assistance to the surgeon, there are other robotic systems which, although they are not completely autonomous, provide a much higher

Surgical Robotics,
Fig. 1 da Vinci Research
Kit



level of assistance than the da Vinci system. In particular, they carry out preoperative planning on the basis of image processing and assisted execution of the task by means of virtual fixtures (VFs) or active constraints. VFs can be classified into two main classes: forbidden-region virtual fixtures (FRVFs) and guidance virtual fixtures (GVFs). Generally speaking, FRVFs are suitable for simulating barriers, constraining surfaces or delicate regions that the user should be forbidden to enter. In contrast, a GVF has an attractive behavior that pulls the instrument toward a particular path.

Autonomous robotic planning and semi-autonomous execution of surgical tasks have been selectively achieved in orthopedics and other areas where the surgical environment is mainly constituted by rigid anatomical parts. A comprehensive review of surgical robots available on the market or as research platforms since the mid-1990s is provided in Hoeckelmann et al. (2015).

Various surgical robots are more than slave devices and have been successfully deployed in operating rooms. For example, ACROBOT (Jakopec et al. 2002) and MAKO RIO (Hagag et al. 2011), used for precise bone cutting, operate with active constraints in a shared control modality. ROBODOC (Netravali et al. 2016), used for orthopedic surgery, and CyberKnife (Dieterich and Gibbs 2011), used for stereotactic radiosurgery, are examples of robotic systems working in supervised autonomy by carrying out image-based preoperative plans without human intervention except in emergency situations. NeuroMate (Varma and Eldridge 2006), a robotic system for minimally invasive neurosurgery, is endowed with supervised planning capabilities, taking advantage of fiducial markers located in the patient's head using ultrasound, while its trajectory planning is based on high-resolution preoperative images. Other advanced robotic systems are used for high-precision assistance in eye surgery. The PRECEYES surgical system (<http://www.preceyes.nl/>) is a robotic assistant for vitreoretinal surgery. It supports surgeons in inserting and manipulating instruments inside the eye. The system is guided by positioning

commands by the surgeon through an intuitive motion controller. The system provides surgeons with a precision better than 20 μm to position and hold instruments steady for an extended period of time. The ARTAS iX (<https://www.venustreatments.com/en-gl/artas-ix.htm>) robotic hair restoration device is an advanced, minimally invasive hair transplant system that uses artificial intelligence (AI) technology to deliver precise results. Using advanced image-guided robotics, ARTAS iX can precisely analyze and dissect the best grafts from the donor area thousands of times per session and then accurately identify where they should be implanted to achieve a personalized hairline. Finally Yomi (<https://www.neocis.com/>) is the first and only FDA-cleared robotic device for dental implant surgery. It provides haptic robotic guidance to augment the clinical expertise of skilled implant dentists and deliver repeatable surgical precision.

Levels of Autonomy

A hierarchy of six different levels of autonomy for medical robots, including surgical robots, was introduced in Yang et al. (2017): starting from medical robots having no autonomy (Level 0), to robotic assistants constraining or correcting human action (Level 1 autonomy), robotic systems carrying out tasks designated by humans and under human supervision (Level 2), and robotic systems generating task execution strategies under human supervision (Level 3); this hierarchy is ideally rounded out by reference to technologically more distant perspectives. At Level 4 one envisages robots performing under human supervision an entire medical procedure (e.g., an entire intraoperative surgical intervention), while full autonomy without any human supervision characterizes culminating Level 5. At Level 0, surgeons govern the entire surgical procedure in master–slave control mode, from data analysis and planning to decision-making and actual execution. By scaling motion and attenuating tremors, robots compensate for human sensory-motor limitations and improve the execution of imperfectly conveyed motion commands. This correction, however, is applied

solely with the goal of bringing resulting actions closer to the actual intentions of human surgeons. The perfect example is da Vinci and similar robotic systems recently introduced on the market, such as TransEnterix (<https://transenterix.com/>), CMR Surgical (<https://cmrsurgical.com/>), and so forth. In contrast to Level 0 autonomous systems, robotic surgery assistants placed at autonomy Level 1 are not invariably bound to bring about a better fit between human intentions on one hand and imperfectly conveyed human commands on the other hand. Indeed, robotic assistants applying VFs may act against some consciously entertained intentions of human surgeons, by actively constraining and modifying humanly determined trajectories of surgical instruments. This is the case of Yomi. Analogous VF functions are developed in robotic microsurgery and implemented in robotic systems such as PRECEYES. At Level 2, the robotic systems plan a task under the supervision of humans that can adjust the planning. Afterward, the robot performs the task in a shared modality or autonomously under human supervision. The surgeon's supervising role consists in the hands-free monitoring of robotic task execution and prompt overriding if needed. Hence, the robotic system is under the surgeon's discrete, rather than continuous control. ROBODOC and MAKO RIO are examples of systems that fall into this category. Robotic systems endowed with Level 3 autonomy generate task strategies and conditionally rely on humans to select from among different strategies or to approve an autonomously selected strategy (Yang et al. 2017). To a limited extent, systems dynamically identifying VFs and generating optimal control parameters or trajectories already achieve this level of conditional autonomy. The system NeuroMate (Varma and Eldridge 2006) mentioned in the previous section performs trajectory planning based on high-resolution preoperative images. Other pertinent examples include CyberKnife and ARTAS iX, which generate execution plans that the surgeon is required to review and approve (Yip and Das 2017). Human control for Level 3 autonomy distinctively requires surgeons to

decide competently whether to approve one of the robot-generated strategies.

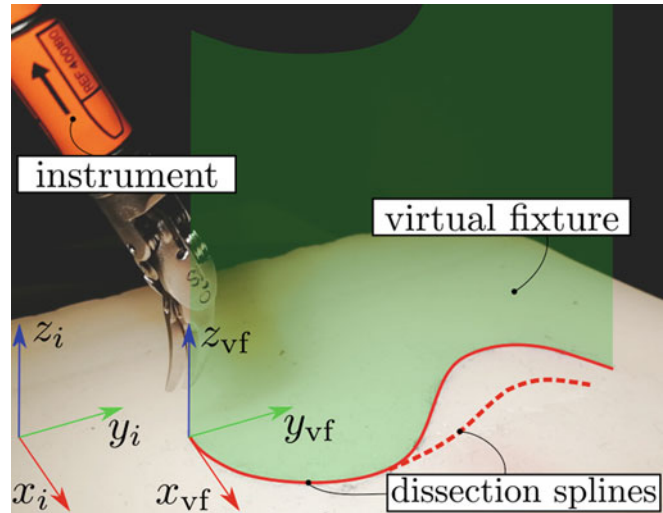
The kind of refinement proposed in Yang et al. (2017) for autonomy levels is solely based on the “what” of surgical robot autonomy. In Ficuciello et al. (2019) intra-level refinements are based on the “how” and on the “where” of surgical robot autonomy. The “where” referred to the surgical site or anatomy where the task is executed. The autonomy capabilities of a robotic system depend on where the task is performed, whether in a rigid and invariable environment (ROBODOC, MAKO RIO, or ARTAS) or on tissues and organs, deformable and subject to variations due to periodic movements of the patient (breath). The latter is the case of da Vinci and similar systems. In addition, the autonomy of the robot depends on the “how” the task is performed in the sense of cognitive and sensorimotor skills employed for its execution. For example, a VF can be realized off-line or online if there are reliable sensors and efficient algorithms. Therefore, depending on the sensorimotor and cognitive skills, a level of autonomy can be more or less pushed even if the concept behind it is the same.

Research on Sensor-Based Shared Autonomy

In most surgical robots, the basic shared control consists of aiding surgeons in a master-slave architecture by scaling the motion, reducing tremors, and enhancing precision. As discussed above, the step forward is the use of VFs to monitor and enhance the surgeon's performance improving safety, accuracy, and speed (Becker et al. 2011; Selvaggio et al. 2018) (Fig. 2). A state of the art on VFs is reported in Bowyer et al. (2014). The most significant challenge is the generation of constrained geometries in an effective and dynamic way (Moustris et al. 2011) taking into account tissue motion (Moustris et al. 2011). A vision-based method for robot-aided polyp dissection using the dVRK robot is proposed in Moccia et al. (2019). The work defines a functioning pipeline to assist the surgeon in the dissection task performing path planning of the cutting task and impedance control to enforce

Surgical Robotics,

Fig. 2 Curvilinear dissection task with VF geometry adaptation. $\mathcal{F}_{vf} : \{\mathbf{O}_{vf}; \mathbf{x}_{vf}, \mathbf{y}_{vf}, \mathbf{z}_{vf}\}$ is the VF reference frame, $\mathcal{F}_i : \{\mathbf{O}_i; \mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i\}$ is the inertial frame



guidance VF constraints. VF can also be used to avoid collision between instruments (Moccia et al. 2020).

A review of the state of the art regarding the autonomous systems and algorithms for MIS is presented in Moustiris et al. (2011). The automation of surgical tasks has been demonstrated using white light image (WLI) information in constrained phantom environments for automated stitching and excision, driven by image inference in visual servo control (Murali et al. 2016). Intraoperative hyperspectral imaging (HSI) has been shown to provide tissue features which are not visible in white light (Clancy et al. 2015). Additionally, HSI can intraoperatively monitor the perfusion state of various types of anastomoses by detecting and quantifying parameters such as tissue hemoglobin index (THI), tissue oxygen saturation (StO₂), near-infrared perfusion index (NIR-Index), and tissue water index (TWI), which can be displayed as pseudo-color images (Jansen-Winkel et al. 2018).

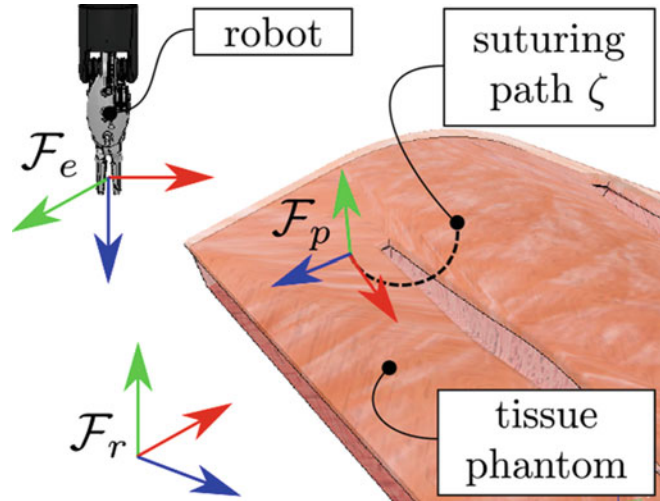
Suturing (Fig. 3) is one of the most significant tasks to be automated due to the difficulty in performing it using telerobotic systems without force feedback (Sen et al. 2016). Surgical threads are highly dynamic and deformable, and consequently they present a significant challenge. In addition, the surgical needle needs to be detected and tracked precisely (Jackson et al. 2015; Selvaggio et al. 2019b).

A survey on the application of machine learning techniques in the context of surgery is reported in Kassahun et al. (2016). Some techniques exploiting the modern learning methods aim at acquiring information from the surgeons and adapt, as a consequence, robot behavior in the future (Reiley et al. 2010). Learning-by-demonstration approaches are used to automate surgical subtasks (van den Berg et al. 2010) (e.g., suturing) with increased performance in terms of speed and smoothness and to define a finite state machine (FSM) (Murali et al. 2015) that models some surgical subtasks. Recently, a joint European group has developed several machine learning algorithms allowing for the autonomous discrimination between healthy and cancerous brain tissue (Fabelo et al. 2018). A human-machine collaborative (HMC) system (Padoy and Hager 2011) is proposed to learn, from surgical demonstration, how to collaboratively assist the surgeon during a teleoperated task. To do this it is necessary to find a cost function to be optimized for motion planning. To determine the cost function, inverse reinforcement learning (IRL) is a suitable machine learning framework. The goal of this is indeed to infer the underlying reward structure guiding an agent's behavior based on observation.

Cooperative inverse reinforcement learning (CIRL) formalizes the problem as a two-player game between the human and the robot, in which

Surgical Robotics,

Fig. 3 A schematic surgical setting during the suturing task: $\mathcal{F}_r$, $\mathcal{F}_e$ denote the inertial and end-effector reference frames, respectively. $\mathcal{F}_p$ denotes a generic needle-tip trajectory pose, i.e., position and orientation



only the human knows the parameters of the reward function: the robot needs to learn them as the interaction unfolds (Malik et al. 2018).

Summary and Future Directions

The future of surgical robotics is in the development of miniaturized robotic systems that can navigate the human body and execute surgical procedures in very narrow spaces using autonomous and semi-autonomous manipulations skills. Smart materials, innovative sensing, AI, and advanced control architecture are the keywords for the evolution of robotic technology in surgery. Over the last decade, extensive efforts have been made to miniaturize surgical devices to improve endoluminal theranostics, i.e., to perform simultaneously diagnostics and treatment. The principles of soft robotics (Grazioso et al. 2019) are the key to success. The effort is to create soft, compliant, and adaptable eversion robots (Ataka et al. 2020) whose structure is fused with function, bringing together capabilities of locomotion, perception, and physical interaction with the environment in one holistic approach.

Generally speaking, the current limitations of automation of surgical gesture not only in miniaturized robots but also in robotic laparoscopy can be overcome by using new vision sensors. Some significative examples are

Terahertz (THz) imaging and HSI to avoid, if possible, the use of dye injection. THz imaging provides enhanced contrast compared to 3D WLI, along with the capability to sense physiological information crucial for tissue characterization allowing the inspection of multi-layered structures and the evaluation of biological material properties. THz holds great potential for cancer diagnosis. HSI is a complementary imaging technique, providing spatial and spectral information simultaneously, allowing for a real-time intraoperative quantitative analysis of tissue properties with high spatial resolution. Also HSI has been proven to enhance the diagnostic ability of various neoplasia including colorectal cancer (CRC) (Kumashiro et al. 2016). Additionally, by discriminating various tissues based on spectral signatures, HSI has the potential to be a useful surgical guidance tool.

With the aid of new sensing technologies, such as the ones described above, it is possible to apply multimodal sensing for surgical application by integrating different image sources with white light view. This can solve the limitations of endoscopic vision for 3D surgical site reconstruction, for the detection and tracking of surgical instruments, for the identification of different tissue types or structures such as blood vessels or nerves, and for the tracking of deformable materials including layers and fiber detection. Using innovative sensing, advanced control strategies can be developed by exploiting

multisensory information and algorithms for data elaboration. A new paradigm should be created for robot-aided surgery toward the safe semi-autonomous execution of complex tasks requiring surgeon-robot interaction and workflow monitoring. In particular, suturing and dissection procedures can be automatized in microsurgery thanks to an enhanced perception and reconstruction of the environment. A high-level control architecture can be designed to manage and switch between different modalities, such as vision-based motion planning and control, virtual fixture control, impedance-based control, and optimal control integrating learning from human demonstration.

As for advanced control applied to surgical robots, a question may arise: What is the need to automate the surgical gesture in some contexts and circumstances? The answer is in the following: thanks to the diffusion of automated robotic surgery and microsurgery, less experienced surgeons will be able to perform more complex procedures. The patients will benefit from having more centers offering the procedures closer to home at high-target level, which is particularly important in the case of emergency care, but this technology will also drastically reduce the numbers of untreated cases. In the long run, we expect surgery to gradually become more and more automated, draining inspiration from the evolution of autonomous robotic systems endowed with AI technologies.

These forthcoming developments suggest the opportunity to start an in-depth discussion about meaningful human control (MHC) in semi-autonomous and autonomous surgical systems (Ficuciello et al. 2019). The notion of MHC was originally introduced in ethical and legal discussions of autonomous weapons systems and was more recently investigated in connection with other autonomous systems (de Sio and van den Hoven 2018). MHC is broadly understood as an ethical policy approach which is supposed to ensure human responsibilities and accountability while admitting limited forms of robotic autonomy. What MHC should distinctively amount to in robotic surgery and how one should modulate MHC considering

autonomous task assignments, contexts of use, and related robot skills are today open problems.

Cross-References

- ▶ [Deep Learning in a System Identification Perspective](#)
- ▶ [Image-Based Estimation for Robotics and Autonomous Systems](#)
- ▶ [Image-Based Robot Control](#)
- ▶ [Iterative Learning Control](#)
- ▶ [Optimal Control and the Dynamic Programming Principle](#)
- ▶ [Redundant Robots](#)
- ▶ [Reinforcement Learning and Adaptive Control](#)
- ▶ [Robust Adaptive Control](#)
- ▶ [Robot Learning](#)
- ▶ [Robot Motion Control](#)
- ▶ [Robot Teleoperation](#)
- ▶ [Robot Visual Control](#)

Recommended Reading

In addition to the works cited in the text, further reading providing a more comprehensive review of this rapidly expanding field can be found in Taylor et al. (2016). Here, basic concepts of computer integrated surgery are introduced, critical factors affecting the deployment and acceptance of medical robots are discussed, basic system paradigms of surgical computer-assisted planning, execution, monitoring, and assessment are described, as well as additional topics such as remote telesurgery and robotic surgical simulators are discussed. Another interesting publication mainly dedicated to medical image computing can be found in Zhou et al. (2019). The book provides a comprehensive reference on current technical approaches and solutions in medical image computing and computer-assisted intervention. About machine learning techniques and their role in intelligent and autonomous surgical actions, an interesting survey can be found in Kassahun et al. (2015). Recent publications on new sensing technologies for surgery are Dhillon et al. (2017) and Lu and Fei (2014). Finally,

an interesting reading on trusted autonomous surgical robots and an in-depth analysis of the levels of autonomy are given by Attia et al. (2018) and Haidegger (2019), respectively.

Bibliography

- Ataka A, Abrar T, Putzu F, Godaba H, Althoefer K (2020) Model-based pose control of inflatable eversion robot with variable stiffness. *IEEE Robot Autom Lett* 5(2):3398–3405
- Attia M, Hossny M, Nahavandi S, Dalvand M, Asadi H (2018) Towards trusted autonomous surgical robots. In: *IEEE international conference on systems, man, and cybernetics*, pp 4083–4088
- Becker BC, MacLachlan RA, Hager GD, Riviere CN (2011) Handheld micromanipulation with vision-based virtual fixtures. In: *IEEE international conference on robotics and automation*, pp 4127–4132
- Bowyer SA, Davies BL, Rodriguez y Baena F (2014) Active constraints/virtual fixtures: a survey. *IEEE Trans Robot* 30(1):138–157
- Clancy N, Arya S, Stoyanov D, Singh M, Hanna G, Elson D (2015) Intraoperative measurement of bowel oxygen saturation using a multispectral imaging laparoscope. *Biomed Opt Express* 6(10):4179–4190
- de Sio FS, van den Hoven J (2018) Meaningful human control over autonomous systems: a philosophical account. *Front Robot AI* 5:15
- Dhillon SS, Vitiello MS, Linfield EH, Davies AG, Hoffmann MC, Booske J, Paoloni C, Gensch M, Weightman P, Williams GP, Castro-Camus E, Cumming DRS, Simoens F, Escorcia-Carranza I, Grant J, Lucyszyn S, Kuwata-Gonokami M, Konishi K, Koch M, Schmuttenmaer CA, Cocker TL, Huber R, Markelz AG, Taylor ZD, Wallace VP, Zeitler JA, Sibik J, Korter TM, Ellison B, Rea S, Goldsmith P, Cooper KB, Appleby R, Pardo D, Huggard PG, Krozer V, Shams H, Fice M, Renaud C, Seeds A, Stöhr A, Naftaly M, Ridler N, Clarke R, Cunningham JE, Johnston MB (2017) The 2017 terahertz science and technology roadmap. *J Phys D Appl Phys* 50(4):043001
- Dieterich S, Gibbs IC (2011) The cyberknife in clinical use: current roles, future expectations. *Front Radiation Therapy Oncology* 43:181–194
- Fabelo H, Ortega S, Ravi D, Kiran B, Sosa C, Bulters D, Callicó G, Bulstrode H, Szolna A, Piñeiro J, Kabwama S, Madroñal D, Lazcano R, J-O'Shanahan A, Bisshopp S, Hernández M, Báez A, Yang G, Stanculescu B, Salvador R, Juárez E, Sarmiento R (2018) Spatio-spectral classification of hyperspectral images for brain cancer detection during surgical operations. *PLoS One* 13(3):e0193721
- Ficuciello F, Tamburrini G, Arezzo A, Villani L, Siciliano B (2019) Autonomy in surgical robots and its meaningful human control. *Paladyn J Behav Robot* 10(1):30–43
- Fontanelli GA, Ficuciello F, Villani L, Siciliano B (2017) Modelling and identification of the da vinci research kit robotic arms. In: *IEEE/RSJ international conference on intelligent robots and systems*, pp 1464–1469
- Fontanelli GA, Selvaggio M, Buonocore LR, Ficuciello F, Villani L, Siciliano B (2018) A new laparoscopic tool with in-hand rolling capabilities for needle reorientation. *IEEE Robot Autom Lett* 3(3):2354–2361
- Fontanelli G, Selvaggio M, Ferro M, Ficuciello F, Venditelli M, Siciliano B (2019) Portable dVRK: an augmented v-rep simulator of the da vinci research kit. *Acta Polytech Hung* 16(8):79–98
- Fontanelli GA, Buonocore LR, Ficuciello F, Villani L, Siciliano B (2020) An external force sensing system for minimally invasive robotic surgery. *IEEE/ASME Trans Mechatron* 25(3):1543–1554
- Grazioso S, Di Gironimo G, Siciliano S (2019) A geometrically exact model for soft continuum robots: the finite element deformation space formulation. *Soft Robot* 6(6):790–811
- Hagag B, Abovitz R, Kang H, Schmitz B, Conditt M (2011) RIO: robotic-arm interactive orthopedic system MAKOpasty: user interactive haptic orthopedic robotics. In: *Surgical robotics*. Springer, Boston
- Haidegger T (2019) Autonomy for surgical robots: concepts and paradigms. *IEEE Trans Med Robot Bionics* 1(2):65–76
- Hirche S, Buss M (2012) Human-oriented control for haptic teleoperation. *Proc IEEE* 100(3):623–647
- Hoeckelmann M, Rudas IJ, Fiorini P, Kirchner F, Haidegger T (2015) Current capabilities and development potential in surgical robotics. *Int J Adv Robot Syst* 12(5):61–100
- Jackson RC, Yuan R, Chow D, Newman W, Çavuşoglu MC (2015) Automatic initialization and dynamic tracking of surgical suture threads. In: *IEEE international conference on robotics and automation*, pp 4710–4716
- Jakopcic M, Harris SJ, y Baena FR, Gomes P, Davies BL (2002) Acrobot: a “hands-on” robot for total knee replacement surgery. In: *7th international workshop on advanced motion control*, pp 116–120
- Jansen-Winkel B, Maktabi M, Takoh J, Rabe S, Barberio M, Köhler H, Neumuth T, Melzer A, Chalopin C, Gockel I (2018) Hyperspectral imaging of gastrointestinal anastomoses. *Der Chirurg; Zeitschrift für alle Gebiete der operativen Medizin* 89(9):717–725
- Kassahun Y, Yu B, Tibebu AT, Stoyanov D, Giannarou S, Metzen JH, Poorten EBV (2015) Surgical robotics beyond enhanced dexterity instrumentation: a survey of machine learning techniques and their role in intelligent and autonomous surgical actions. *Int J Comput Assist Radiol Surg* 11:553–568
- Kassahun Y, Yu B, Tibebu A, Stoyanov D, Giannarou S, Metzen JH, Poorten EV (2016) Surgical robotics beyond enhanced dexterity instrumentation: a survey of machine learning techniques and their role in intelligent and autonomous surgical actions. *Int J Comput Assist Radiol Surg* 11(4):553–568
- Kazanzides P, Chen Z, Deguet A, Fischer GS, Taylor RH, DiMaio SP (2014) An open-source research

- kit for the da vinci surgical system. In: IEEE international conference on robotics and automation, pp 6434–6439
- Kumashiro R, Konishi K, Chiba T, Akahoshi T, Nakamura S, Murata M, Tomikawa M, Matsumoto T, Maehara Y, Hashizume M (2016) Integrated endoscopic system based on optical imaging and hyperspectral data analysis for colorectal cancer detection. *Anticancer Res* 36(8):3925–3932
- Lu G, Fei B (2014) Medical hyperspectral imaging: a review. *J Biomed Opt* 19(1):1–24
- Malik D, Palaniappan M, Fisac J, Hadfield-Menell D, Russell S, Dragan A (2018) An efficient, generalized Bellman update for cooperative inverse reinforcement learning. In: Dy J, Krause A (eds) *Proceedings of the 35th international conference on machine learning*, PMLR, Stockholm, vol 80, pp 3394–3402
- Moccia R, Selvaggio M, Villani L, Siciliano B, Ficuciello F (2019) Vision-based virtual fixtures generation for robotic-assisted polyp dissection procedures. In: *IEEE/RSJ international conference on intelligent robots and systems*, pp 7934–7939
- Moccia R, Iacono C, Siciliano B, Ficuciello F (2020) Vision-based dynamic virtual fixtures for tools collision avoidance in robotic surgery. *IEEE Robot Autom Lett* 5(2):1650–1655
- Moustiris G, Hirisidis S, Deliparaschos K, Konstantinidis K (2011) Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature. *Int J Med Robot Comput Assist Surg* 7(4):375–392
- Murali A, Sen S, Kehoe B, Garg A, McFarland S, Patil S, Boyd WD, Lim S, Abbeel P, Goldberg K (2015) Learning by observation for surgical subtasks: multi-lateral cutting of 3D viscoelastic and 2D orthotropic tissue phantoms. In: *IEEE international conference on robotics and automation*, pp 1202–1209
- Murali A, Garg A, Krishnan S, Pokorny FT, Abbeel P, Darrell T, Goldberg K (2016) Tsc-dl: unsupervised trajectory segmentation of multi-modal surgical demonstrations with deep learning. In: *IEEE international conference on robotics and automation*, pp 4150–4157
- Netravali N, Börner M, Bargar W (2016) The use of ROBODOC in total hip and knee arthroplasty. In: *Computer-assisted musculoskeletal surgery*. Springer, Cham
- Padoy N, Hager GD (2011) Human-machine collaborative surgery using learned models. In: *IEEE international conference on robotics and automation*, pp 5285–5292
- Reiley CE, Plaku E, Hager GD (2010) Motion generation of robotic surgical tasks: learning from expert demonstrations. In: *Annual international conference of the IEEE engineering in medicine and biology*, pp 967–970
- Selvaggio M, Fontanelli GA, Ficuciello F, Villani L, Siciliano B (2018) Passive virtual fixtures adaptation in minimally invasive robotic surgery. *IEEE Robot Autom Lett* 3(4):3129–3136
- Selvaggio M, Fontanelli G, Marrazzo V, Bracale U, Irace A, Breglio G, Villani L, Siciliano B, Ficuciello F (2019a) The MUSHa underactuated hand for robot-aided minimally invasive surgery. *Int J Med Robot Comput Assist Surg* 15(3):e1981
- Selvaggio M, Ghalamzan A, Moccia R, Ficuciello F, Siciliano B (2019b) Haptic-guided shared control for needle grasping optimization in minimally invasive robotic surgery. In: *IEEE/RSJ international conference on intelligent robots and systems*, pp 3617–3623
- Sen S, Garg A, Gealy DV, McKinley S, Jen Y, Goldberg K (2016) Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization. In: *IEEE international conference on robotics and automation*, pp 4178–4185
- Taylor RH, Menciassi A, Fichtinger G, Fiorini P, Dario P (2016) Medical robotics and computer-integrated surgery. In: Siciliano B, Khatib O (eds) *Springer handbook of robotics*. Springer International Publishing, Cham, pp 1657–1683
- van den Berg J, Miller S, Duckworth D, Hu H, Wan A, Fu X, Goldberg K, Abbeel P (2010) Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In: *IEEE international conference on robotics and automation*, pp 2074–2081
- Varma T, Eldridge P (2006) Use of the neuromate stereotactic robot in a frameless mode for functional neurosurgery. *Int J Med Robot Comput Assist Surg* 2(2):107–113
- Yang G-Z, Cambias J, Cleary K, Daimler E, Drake J, Dupont PE, Hata N, Kazanides P, Martel S, Patel RV, Santos VJ, Taylor RH (2017) Medical robotics—regulatory, ethical, and legal considerations for increasing levels of autonomy. *Sci Robot* 2(4):eaam8638
- Yip M, Das N (2017) Robot autonomy for surgery, CoRR, abs/1707.03080. <http://arxiv.org/abs/1707.03080>
- Zhou SK, Rueckert D, Fichtinger G (eds) (2019) *Handbook of medical image computing and computer assisted intervention*. Academic Press, p 1072

Switching Adaptive Control

Minyue Fu

School of Electrical Engineering and Computing, University of Newcastle, Callaghan, NSW, Australia

Abstract

Switching adaptive control is one of the advanced approaches to adaptive control. By employing an array of simple candidate controllers, a properly designed monitoring function, and a supervisory switching law,

this approach is capable to search in real time for a correct candidate controller to achieve the given control objective such as stabilization and set-point regulation. This approach can deal with large parameter uncertainties and offers good robustness against unmodelled dynamics. This entry offers a brief introduction to switching adaptive control, including some historical background, basic concepts, key design components, and technical issues.

Keywords

Adaptive control · Hybrid systems ·
Multiple models · Supervisory control ·
Switching logic · Uncertain systems

Introduction

Switching adaptive control, also known as *switched adaptive control* or *multiple model adaptive control*, refers to an *adaptive control* technique which deploys a set of controllers and a switching law to achieve a given control objective. The concept of switching adaptive control is generalized from the traditional *gain scheduling* technique (Leith and Leithead 2000). As in the standard adaptive control setting, the model for the controlled plant is assumed to contain uncertain parameters, and the control objective is to stabilize the system and, in many cases, to deliver certain performance using real-time information in the measured output. What differentiates switching adaptive control from gain scheduling is that the uncertain parameters are not directly measured, and the switching is determined by the system response. This seemingly minor difference is very important because parameter estimation may not be possible due to the lack of persistent excitation; moreover, the sensitivity of the measured output is often suppressed by the feedback control which makes closed-loop identification of the uncertain parameters difficult. Compared with classical adaptive control, switching adaptive control has better inherent robustness against parameter uncertainties and unmodelled dynamics.

By early 1980s, the classical adaptive control theory for linear systems had been well established under a set of so-called classical assumptions, which include:

- Known order of the plant (or known maximum order of the plant)
- Known relative degree of the plant
- Minimum phase dynamics
- Known sign of the high-frequency gain (which is the gain of the plant when the input is high-frequency sinusoidal signal)

At the same time, it was recognized that the classical adaptive control approach has inherent robustness problems against even miniature unmodelled dynamics (Rohrs et al. 1985). While this generated a wave of research aiming at robustification of the classical adaptive control theory (see, e.g., Ioannou and Sun 1996), a new line of research took place aiming at relaxing the classical assumptions. Nussbaum (1983) paved the way by showing that knowledge of the sign of the high-frequency gain can be avoided for a first-order linear system. Morse (1985) developed a “universal controller” which can adaptively stabilize any strictly proper, minimum-phase system with relative degree not exceeding two. Martensson (1985) gave a very surprising result by showing that asymptotic stabilization can be achieved adaptively by simply assuming that there exists a finite order stabilizer. But Martensson’s controller is impractical due to the need for exhaustive online search of the stabilizer and subsequent excessively high overshoots. Switching adaptive control was then introduced in Fu and Barmish (1986), aiming at achieving adaptive stabilization with minimal assumptions and a guarantee of exponential convergence rate for the state. In contrast to the work of Martensson, a compactness requirement is made on the set of possible plants and an upper bound on the order of the plant is assumed. These assumptions allow a set of possible plants to be partitioned into a finite number of subsets, with each stabilizable by a single controller. A monitoring function and a switching law are then designed to sequentially eliminate incorrect candidate controllers until an

appropriate controller is found. Due to the fact that the number of candidate controllers may be large, many follow-up works on switching adaptive control focused on speeding up the switching process by eliminating incorrect candidate controllers without trying them (Zhivoglyadov et al. 2000, 2001). These results can also deal with slowly time-varying parameters and infrequent parameter jumps.

Another major breakthrough came from the works of Morse (1996, 1997) under the term of *supervisory control*. His work considers set-point regulation for uncertain linear systems. A different compactness requirement is used to allow unmodelled dynamics in the system. More specifically, the given uncertain linear system is assumed to belong to a union of subfamilies of systems, with each subfamily having a linear controller capable to achieve set-point regulation. Suitably defined output-squared estimation errors are used as monitoring functions, and a candidate controller is selected whose corresponding performance signal is the smallest. The major advantages of this switching law are that the “correct” controller can usually be quickly identified without cycling through all possible candidate controllers, leading to a good closed-loop performance.

More recent research on switching adaptive control focuses on more systematic and alternative approaches to the design of candidate controllers and switching laws; see, e.g., Anderson et al. (2000), Hespanha et al. (2001), and Morse (2004). Generalizations to nonlinear systems are also found (Battistelli et al. 2012).

Design of Switching Adaptive Control

A switching adaptive controller consists of the following key ingredients:

- Design of control covering
- Design of monitoring function
- Selection of dwell time

For illustrative purposes, we consider an adaptive stabilization problem where the system has the following model:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}$$

with state $x(t) \in R^n$ for some $1 \leq n \leq n_{\max}$ and the measured output $y(t) \in R^r$. The given set of uncertain plants Σ consists of triplets (A, B, C) , and we use the notation $\Sigma^{(n)}$ to denote the subset of Σ consisting of those plants having order n . It is assumed that every possible plant $(A, B, C) \in \Sigma$ is a minimal realization (i.e., both controllable and observable) and that every $\Sigma^{(n)}$ is a compact set (i.e., it is closed and bounded). The control objective is to design an adaptive controller to drive the state to zero asymptotically, i.e., $x(t) \rightarrow 0$ as $t \rightarrow \infty$. It is clear that each possible plant in Σ admits a linear dynamic stabilizer. An alternative description of the uncertain plant is introduced in Morse (1996, 1997) where its transfer function is a member of a continuously parameterized set of admissible transfer functions of the form

$$\Sigma \subset \bigcup_{p \in \mathcal{P}} \{v_p + \delta : \|\delta\| \leq \varepsilon_p\}$$

In the above, $\mathcal{P}$ is a compact set in a finite dimensional space, v_p is a nominal transfer function with its coefficients depending continuously on p , δ is the transfer function of some unmodelled dynamics, $\|\delta\|$ represents a shifted H_∞ norm (obtained by first shifting the poles of δ slightly to the right and then computing its H_∞ norm), and ε_p is sufficiently small so that each set of plants $\{v_p + \delta : \|\delta\| \leq \varepsilon\}$ is stabilizable by a single controller for all $p \in \mathcal{P}$.

Control covering The purpose is to decompose the given set of plants into a union of subsets such that each subset P_i admits a single controller K_i (called candidate controller) to achieve the given control objective. This is typically done using two properties: inherent robustness of linear controllers and the existence of a finite cover for any compact set. More specifically, if a candidate

controller renders a desired control objective for a given plant, then the same objective is maintained when the plant is perturbed slightly. For example, Fu and Barmish (1986) uses the fact that if a given plant is stabilized by a controller then the same controller stabilizes all the plants with sufficiently small parameter perturbations. Similarly, Morse (1996, 1997) uses the fact that the same controller achieves set-point regulation for a small neighborhood of plants. Combining this property with the finite covering property yields

$$\Sigma = \bigcup_{i=1}^N \Sigma_i$$

such that each subset Σ_i admits a single controller K_i .

Monitoring Function The generation of the adaptive switching controller is accomplished using a *switching law* or *switching logic* whose task is to determine, at each time instant, which candidate controller is to be applied. The core of the switching law is a monitoring function. Its very basic role is to be able to detect whether the applied candidate controller is consistent with the corresponding plant subset so that wrong candidate controllers can be eliminated one by one until an appropriate controller is found. A major difficulty for switching adaptive control design is that persistent excitation is not assumed. Consequently, it is not always possible to detect the correct plant subset using the measured output. The key idea is to check which plant subsets are consistent with the generated output.

One simple monitoring function uses a finite-time L_2 norm of the measured output:

$$V(t, \tau) = \int_{t-\tau}^t \|y(s)\|^2 ds$$

where τ is the so-called dwell time. It turns out that for some properly chosen dwell time, a correctly applied candidate controller is able to guarantee some decay property for the monitoring function, i.e., $V(t, \tau) \leq e^{-\lambda\tau} V(t - \tau, \tau)$ for

some $\lambda > 0$. This property is sufficient to allow a wrong candidate controller to be eliminated. However, much smarter monitoring functions can be designed so that infeasible candidate controllers (those not corresponding to the true plant) can be eliminated without even being applied. This can be done using the *falsification* approach in parameter estimation where the basic idea is to eliminate all plant subsets Σ_i inconsistent with the measured output signal. For example, consider the following discrete-time model:

$$\begin{aligned} y(t) = & -a_1 y(t-1) - a_2 y(t-2) \\ & + b_1 u(t-1) + b_2 u(t-2) + w(t) \end{aligned}$$

where a_i and b_i are uncertain parameters and $w(t)$ is a bounded disturbance, i.e., $|w(t)| \leq \delta$ for some δ . For this example, we may eliminate all the uncertain parameter subsets which violate the following constraint (Zhivoglyadov et al. 2000):

$$\begin{aligned} |y(t) + a_1 y(t-1) + a_2 y(t-2) \\ - b_1 u(t-1) - b_2 u(t-2)| \leq \delta \end{aligned}$$

More generally, one can use the so-called multi-estimator (Morse 1996, 1997) which involves an array of estimators, one for each plant subset Σ_i using its nominal model. The output estimation error $e_i(l)$ for each such estimator is then used to construct a monitoring function, e.g.,

$$V_i(t, \tau) = \int_{t-\tau}^t e^{-2\lambda(t-s)} \|e_i(s)\|^2 ds$$

where τ is the dwell time as before and $\lambda > 0$ is an exponential weighting parameter used to guarantee the decay rate of the monitoring function as before. Instead of using the monitoring functions to eliminate infeasible candidate controllers, the candidate controller corresponding to the least estimation error, as measured by the least monitoring function, is selected. The main advantage of the multi-estimator-based monitoring functions is that falsification of candidate controllers is done implicitly, and a “correct” controller can be quickly reached, leading to good performance.

Dwell Time The dwell time τ as defined above is a critical component in switching adaptive control. Serving in the monitoring function, this is the minimum nonzero amount of time for a candidate controller to be applied before switching. That is, this provides a sufficient time lag to build the monitoring function so that its exponential decay property is detected when a correct candidate controller is applied. This will allow detection of infeasible plant subsets and selection of a “correct” controller. The use of a dwell time also avoids arbitrarily fast switching, thus guaranteeing the solvability of the system dynamics.

The dwell time can be selected a priori by using the fact that if a matrix A is stable, then there exist some positive values λ and τ such that $\|e^{At}\| \leq e^{-\lambda\tau}$ for all $t > \tau$. This leads to the desired exponential decaying property

$$V(t, \tau) \leq e^{-\lambda\tau V}(t - \tau, \tau)$$

for the aforementioned monitoring function for adaptive stabilization.

Alternatively, the dwell time can be chosen implicitly. Hespanha et al. (2001) suggest a *hysteresis switching logic* method. This method employs a hysteresis parameter $h > 0$. Suppose the candidate controller K_j is applied at time t_i , then K_j is kept until the next switching time t_{i+1} which is the minimum $t \leq t_i$, such that

$$(1 + h) \min_{1 \leq k \leq N} V_k(t, t - t_i) \leq V_j(t, t - t_i)$$

Because $h > 0$, the time difference $t_{i+1} - t_i > 0$ is lower bounded, which implies the existence of a dwell time.

Summary and Future Directions

Switching adaptive control is a conceptually simple control technique capable to deal with large parameter uncertainties. The use of simple candidate controllers (typically linear) imply good closed-loop behavior and good robustness against unmodelled dynamics. Although the discussion above assumes that the number of plant subsets is finite, this assumption is not essential; see Anderson et al. (2000).

Switching adaptive control renders the closed-loop system a switched system or hybrid system, for which a wide range of tools are available to aid the analysis of such a system; see, e.g., Liberzon (2003). However, unique features of such a system arise from the fact that the switching mechanism is chosen by the designer, rather than being a part of the given plant. How to best design the switching mechanism is an interesting issue.

Future works for switching adaptive control include:

1. How to simplify the design of candidate controllers. Finite covering-based design often yields a large number of plant subsets, hence a large number of candidate controllers. Since most of the candidate controllers do not need to apply (which is the case when falsification based switching logic is used, for example), smarter ways are needed for the design of candidate controllers.
2. Wider applications. Most of the research so far focuses on stabilization and set-point regulation (which is essentially a stabilization problem). How to incorporate general performance criteria is an essential and yet challenging issue.
3. Better design of monitoring functions and the corresponding switching logic. Most existing monitoring functions use a finite-time L_2 norm of the output (or regulation error), with the key feature that some exponential decay property is guaranteed when the candidate controller is “correct.” Note that the key purpose of the monitoring function and the corresponding switching logic is to allow fast falsification of infeasible candidate controllers. Thus, a much wider range of monitoring functions can possibly be used. In particular, how to incorporate set membership identification techniques (Milanese and Taragna 2005) may be of particular interest.

Cross-References

- [Adaptive Control: Overview](#)
- [Robust Model Predictive Control](#)

► **Stability and Performance of Complex Systems Affected by Parametric Uncertainty**

Bibliography

- Anderson BDO, Brinsmead T, Bruyne FD, Hespanha JP, Liberzon D, Morse AS (2000) Multiple model adaptive control. Part 1: finite controller coverings. *Int J Robust Nonlinear Control* 10(11–12):909–929
- Battistelli G, Hespanha JP, Tesi P (2012) Supervisory control of switched nonlinear systems. *Int J Adapt Control Signal Process* 26(8):723–738. Special issue on Recent Trends on the Use of Switching and Mixing in Adaptive Control
- Fu M, Barmish BR (1986) Adaptive stabilization of linear systems via switching control. *IEEE Trans Autom Control* 31(12):1097–1103
- Hespanha JP, Liberzon D, Morse AS, Anderson BDO, Brinsmead T, Bruyne FD (2001) Multiple model adaptive control. Part 2: switching. *Int J Robust Nonlinear Control* 11:479–496
- Ioannou P, Sun J (1996) Robust adaptive control. Prentice Hall, Upper Saddle River
- Leith DJ, Leithead WE (2000) Survey of gain-scheduling analysis and design. *Int J Control* 73(11):1001–1025
- Liberzon D (2003) Switching in systems and control. Birkhäuser, Boston
- Martensson B (1985) The order of any stabilizing regulator is sufficient information for adaptive stabilization. *Syst Control Lett* 6:87–91
- Milanese M, Taragna M (2005) H -infinity set membership identification: a survey. *Automatica* 41:2019–2032
- Morse AS (1985) A three-dimensional universal controller for the adaptive stabilization of any strictly proper minimum-phase system with relative degree not exceeding two. *IEEE Trans Autom Control* 30(12):1188–1191
- Morse AS (1996) Supervisory control of families of linear set-point controllers part I: exact matching. *IEEE Trans Autom Control* 41(10):1413–1431
- Morse AS (1997) Supervisory control of families of linear set-point controllers part II: robustness. *IEEE Trans Autom Control* 42(11):1500–1515
- Morse AS (2004) Lecture notes on logically switched dynamical systems. In: Nistri P, Stefani G (eds) Nonlinear and optimal control theory. Springer, Berlin, pp 61–162
- Nussbaum RD (1983) Some remarks on a conjecture in parameter adaptive control. *Syst Control Lett* 3: 243–246
- Rohrs CE, Valavani L, Athans M, Stein G (1985) Robustness of continuous-time adaptive control algorithms in the presence of un-modeled dynamics. *IEEE Trans Autom Control* 30(9):881–889
- Zhivoglyadov PV, Middleton RH, Fu M (2000) Localization based switching adaptive control for time-varying

discrete-time systems. *IEEE Trans Autom Control* 45(4):752–755

Zhivoglyadov PV, Middleton RH, Fu M (2001) Further results on localization based switching adaptive control. *Automatica* 37:257–263

Synthesis Theory in Optimal Control

Ugo Boscain¹ and Benedetto Piccoli²

¹CNRS, Sorbonne Université, Université de Paris, INRIA team CAGE, Laboratoire Jacques-Louis Lions, Paris, France

²Mathematical Sciences and Center for Computational and Integrative Biology, Rutgers University, Camden, NJ, USA

Abstract

In this entry we review the theory of optimal synthesis. We describe the steps necessary to solve an optimal control problem and the sufficient conditions for optimality given by the theory. We describe some relevant examples that have important applications in mechanics, in the theory of hypoelliptic operators, and for the study of models of geometry of vision. Finally, we discuss the problem of optimal stabilization and the difficulties encountered if one tries to give the solution to the problem in feedback form.

Keywords

Affine control systems · Extremals · Pontryagin maximum principle · Sub-riemannian geometry · Time-optimal synthesis

Optimal Control

An optimal control problem with fixed initial and terminal conditions can be seen as a problem of calculus of variations under nonholonomic constraints:

$$\dot{q}(t) = f(q(t), u(t)), \quad (1)$$

$$\int_0^T L(q(t), u(t)) dt \rightarrow \min \quad (T \text{ fixed or free}), \quad (2)$$

$$q(0) = q_0, \quad q(T) = q_1. \quad (3)$$

Here we make the following set of assumptions:

(H) q belongs to a finite-dimensional smooth manifold M of dimension n . As a function of time, $q(\cdot)$ is assumed to be Lipschitz continuous. The control $u(\cdot)$ is a L^∞ function taking values in a set $U \subset \mathbb{R}^m$. For simplicity, we assume that the functions f and L , defined on $M \times \mathbb{R}^m$, are smooth.

The dynamics $\dot{q}(t) = f(q(t), u(t))$ play the role of the nonholonomic constraint (nonholonomic means that it is a constraint on the velocity but not necessarily on the position).

Solving an optimal control problem in general is a very difficult task. Usually, to attack such a problem, the steps are the following:

- Step 0: Existence. First, one has to guarantee the existence of a solution to (1)–(3). The most important sufficient condition for the existence of minimizers (in the class of Lipschitz functions) is the famous Filippov theorem (see, for instance, Agrachev and Sachkov 2004 for proof). In the following, we state it in the case of T fixed. Introduce a new variable q^0 , and consider the so-called augmented state $\hat{q} := (q^0, q) \in \mathbb{R} \times M$ satisfying the following dynamics:

$$\begin{aligned} \dot{\hat{q}}(t) &= \begin{pmatrix} \dot{q}^0(t) \\ \dot{q}(t) \end{pmatrix} = \begin{pmatrix} L(q(t), u(t)) \\ f(q(t), u(t)) \end{pmatrix} \\ &=: \hat{f}(\hat{q}(t), u(t)) \end{aligned} \quad (4)$$

then if (i) U is compact; (ii) the set of velocities $F(\hat{q}) := \{\hat{f}(\hat{q}, u) \mid u \in U\}$ is convex for every $\hat{q}$; (iii) there exists a compact set $K \subset \mathbb{R} \times M$ such that all solutions of (4) starting from $\hat{q}_0 := (0, q_0)$ stay in K for $t \in [0, T]$; then there exists a minimizer for (1), (2), and (3). Other theorems that can be applied in more general functional classes or under less restrictive hypotheses can be found

in the literature. See, for instance, Bressan and Piccoli (2007), Cesari (1983), and Vinter (2010).

- Step 1: First-order necessary conditions. In optimal control, first-order necessary conditions for optimality are given by the celebrated Pontryagin maximum principle (Pontryagin et al. 1961) (see also Agrachev and Sachkov 2004 for a more recent viewpoint). The Pontryagin maximum principle (PMP for short) extends the (Hamiltonian version of the) Euler-Lagrange equations of calculus of variations to problems with nonholonomic constraints. For a discussion about the relation between variational problems under nonholonomic constraints and variational principles in nonholonomic mechanics, see Bloch (2003).

The PMP restricts the set of candidate optimal trajectories starting from q_0 to a family of trajectories, called *extremals*, parameterized by a co-vector $p(0) \in T_{q_0}^* M$. In addition, there are two kinds of special extremals: (i) the *singular extremals* for which the maximization condition given by the PMP does not permit directly obtaining the control and (ii) the *abnormal extremals* which are candidate optimal trajectories for any cost function. For certain classes of problems, abnormal extremals and singular trajectories coincide.

The set of all trajectories satisfying the PMP (in general having intersections and not being all optimal forever) is called an *extremal synthesis*. The requirement that the trajectories starting from q_0 reach the final point q_1 (at time T , fixed or free) is usually not very useful at this step. This requirement is rather made at Step 4.

- Step 2. Higher-order conditions. Higher-order conditions are used to restrict further the set of candidate optimal trajectories. The most important conditions are those used to eliminate singular extremals (which usually are very hard to treat) as the Goh condition and the generalized Legendre-Clebsch conditions (see, for instance, Agrachev and Sachkov

2004). Other theories that provide higher-order conditions (which apply also to extremals that are not singular) are, for instance, higher-order maximum principles (Bressan 1985; Krener 1977), generalized Morse-Maslov index theories (Agrachev and Sachkov 2004), and envelope theory (Sussmann 1986, 1989; see also Boscaïn and Piccoli 2004, Cap. 1.3.2).

- Step 3. Selection of the optimal trajectories. This step is the most difficult one. Indeed, one should check that each extremal of the extremal synthesis does not intersect another extremal having a smaller cost at the intersection point. This comparison should be done not only among extremals which are close, one to the other, but among all of them. The problem is indeed global.

One of the techniques to address this problem in a very elegant way takes the name of *optimal synthesis theory* and was developed almost together with the birth of the Pontryagin maximum principle. This theory dates back to the paper of Boltyanskii (1966) and was further developed by Brunovsky (1980, 1978), Sussmann (1980, 1979), and Piccoli and Sussmann (2000).

Roughly speaking, the theory of optimal synthesis permits to conclude that if one has an extremal synthesis having certain regularity properties, then this extremal synthesis is indeed an optimal synthesis.

An optimal synthesis is a collection of optimal trajectories starting from q_0 and reaching the various points of the space:

$$\mathcal{S}_{q_0} = \{\gamma_q(\cdot) : [0, T_q] \rightarrow M \mid q \in M, \gamma_q \text{ is a trajectory of (1) minimizing the cost } \int_0^{T_q} L(q(t), u(t)) dt \text{ with } \gamma(0) = q_0, \gamma(T) = q\}.$$

An optimal synthesis should also verify the following condition: if γ_q defined on $[0, T]$ and $\gamma_{q'}$ defined on $[0, T']$ (with $T' \in]0, T[$) belong to $\mathcal{S}_{q_0}$ and we have $q' = \gamma_q(T')$ then $\gamma_{q'} = \gamma_q|_{[0, T']}$. More details are given in the next section.

- Step 4. Selection of the trajectory reaching the final point. Once an optimal synthesis is computed, one selects the optimal trajectory reaching the desired final point solving the equation $\gamma(T) = q_1$, in the set of all trajectories belonging to the optimal synthesis.

Remark 1 Notice that one could require that the final point is reached at Step 1. This would considerably reduce the set of candidate optimal trajectories already at Step 1, but would not permit to apply the powerful (global) theorems of Step 3. As a consequence, one would be obliged to compare by hands all extremals going from q_0 to q_1 .

Sufficient Conditions for Optimality: The Theory of Optimal Synthesis

There exists a general principle for which every synthesis formed by extremals is optimal under very mild regularity conditions. We will illustrate a classical case of a feedback smooth on a stratification, due to Boltyanskii and Brunovsky (see Boltyanskii 1966; Brunovsky 1980, 1978). More general results can be found in Piccoli and Sussmann (2000). This principle is very strong and is valid only because the synthesis is a global object; while given a single trajectory satisfying PMP, there is no regularity condition which ensures optimality.

For simplicity, from now on, we assume that $M = \mathbb{R}^n$ is the Euclidean space and $q_0 = 0$. Let us indicate by $\mathcal{S}$ a candidate optimal synthesis from 0, the general case follows easily. A set $P \subset M$ is said a *curvilinear open polytope* of dimension p , if there exists a polytope (i.e., bounded closed region intersection of a finite number of half-spaces) $P' \subset \mathbb{R}^p$ and a smooth map $\phi : \mathbb{R}^p \rightarrow \mathbb{R}^n$, injective with Jacobian having maximal rank at every point, such that $\phi(P' \setminus \partial P') = P$.

Let Ω be an open subset of M (for the induced topology) containing the origin in its interior. We say that $\mathcal{S}$ is a *Boltyanskii-Brunovsky regular synthesis*, briefly BB synthesis, if the following holds.

There exists a 6-tuple $\Xi = (\mathcal{P}, \mathcal{P}_1, \mathcal{P}_2, \prod, \Sigma, u)$ such that

- (BB1) $\mathcal{P}$ is a collection of curvilinear open polyhedra and Ω is disjoint union of elements of $\mathcal{P}$. If $P_j \neq P_k \in \mathcal{P}$ and $P_k \cap \overline{P_j} \neq \emptyset$, then $P_k \subset \partial P_j$, and $\dim(P_k) < \dim(P_j)$. $\{0\} \in \mathcal{P}$ and the elements of $\mathcal{P}$ are called “cells.”
- (BB2) $\mathcal{P} \setminus \{0\}$ is the disjoint union of $\mathcal{P}_1$ (the set of “type I cells”) and $\mathcal{P}_2$ (the set of “type II cells”),
- (BB3) the feedback $u : \{q : \exists P_1 \in \mathcal{P}_1, q \in P_1\} \rightarrow U$ and $\prod : \mathcal{P}_1 \rightarrow \mathcal{P}$ are maps, $\Sigma : \mathcal{P}_2 \rightarrow \mathcal{P}_1$ is a multifunction, with non-empty values, such that the following properties are satisfied:

- (i) The function u is of class $\mathcal{C}^1$ on each cell.
- (ii) If $P_1 \in \mathcal{P}_1$, then $f(q, u(q)) \in T_q P_1$ (the tangent space to P_1 at q) for every $q \in P_1$. In addition, for each $q \in P_1$, if we let ξ_q be the maximally defined solution to the initial value problem

$$\dot{\xi} = f(\xi, u(\xi)), \quad \xi(0) = x, \quad \xi \in P_1, \quad (5)$$

and define $t_q = \sup \text{Dom}(\xi_q)$, then the limit $\xi_q(t_q -) := \lim_{t \uparrow t_q} \xi_q(t)$ exists and belongs to $\prod(P_1)$.

- (iii) If $P_2 \in \mathcal{P}_2$, then for each $q \in P_2$ and $P \in \Sigma(P_2)$, there exists a unique curve $\xi_q^P : [0, t_q^P[\rightarrow \Omega$ such that the restriction of ξ_q^P to $]0, t_q^P[$ is a maximally defined integral curve of the vector field $f(\cdot, u(\cdot))$ on P , and $\xi_q^P(0) = q$.
- (iv) On every cell $P_1 \in \mathcal{P}_1$, $q \rightarrow t_q$ is a continuously differentiable function, and $(t, q) \rightarrow \xi_q(t)$ and $(t, q) \rightarrow u_q(t) := u(\xi_q(t))$ are continuously differentiable maps on the set

$$E(P) := \{(t, q) : q \in P_1, t \in [0, t_q]\}.$$

If $P_2 \in \mathcal{P}_2$, the same holds for every t_q^P, ξ_q^P, u_q^P , with $P \in \Sigma(P_2)$.

- (v) For every $q \in \Omega \setminus \{0\}$, the trajectory $\gamma_q : [0, T_q] \rightarrow M, \gamma_q \in \mathcal{S}$, is obtained by piecing together the trajectories on every single cell. Moreover, γ_q changes the cell a finite number of times.

Theorem 1 (Sufficiency Theorem for BB synthesis) *Let S be a BB synthesis on M formed by extremal trajectories; then S is optimal.*

Remark 2 Theorem 1 can be proved also for synthesis on an open subset Ω of M , under suitable conditions (see Piccoli and Sussmann 2000).

Some Relevant Examples

Even if the sufficient conditions for optimality given by the theory of optimal synthesis are very powerful, in general computing, explicitly, an optimal synthesis is very hard, and the complexity grows quickly with the dimension of the space. The main difficulties are:

- The integration of the Hamiltonian equations given by the PMP (which in general is not integrable, unless there are many symmetries);
- The characterization of singular and abnormal extremals;
- The verification of the hypotheses of the sufficient conditions for optimality given by synthesis theory.

For these reasons, the computation of optimal synthesis is already challenging in dimension 2, and a few examples have been solved in dimension 3. In higher dimensions, only very symmetric problems have been completely solved. In the following, we list some of the most relevant optimal syntheses that have been computed up to now.

Time-Optimal Synthesis for Affine Control Systems on 2D Manifolds

Let M be a 2D manifold, and consider the problem of finding the time-optimal synthesis starting

from a point q_0 for a system of the type

$$\dot{q} = F(q) + uG(q), \quad |u| \leq 1, \quad F(q_0) = 0 \quad (6)$$

The condition $F(q_0) = 0$ guarantees local controllability around q_0 , for a generic pair (F, G) . A complete theory for this kind of systems was developed in Bressan and Piccoli (1998), Piccoli (1996), and Boscain and Piccoli (2004), under generic conditions on the vector fields F and G . More precisely, in Boscain and Piccoli (2004), (i) an algorithm building explicitly the time-optimal synthesis; (ii) a classification of synthesis in terms of graphs; (iii) a classification of synthesis singularities; and (iv) an analysis of the properties of the minimum time function were provided.

Here we just recall that optimal trajectories are a finite concatenation of bang (trajectories corresponding to constant control $+1$ or -1) and singular arcs (for which the control may correspond to something different from $+1$ or -1).

Under generic conditions, the optimal synthesis provides a stratification of M . In the regions of dimension 2, the control is either $+1$ or -1 . The regions of dimension 1 called *Frame Curves* can be (i) arcs of optimal trajectories (that may be bang or singular); (ii) switching curves (i.e., curves made of points in which the control switches from $+1$ or -1 , or vice versa); and (iii) overlap curves (i.e., curves made of points where the extremals lose their optimality). The regions of dimension 0 called *Frame Points* are points where frame curves intersect. Generically, they can be of 23 types. See Boscain and Piccoli (2004, p. 60).

Some Relevant Time-Optimal Synthesis for 3D Problems

As we saw in the previous section, for minimum time problems in dimension 2, many results can be obtained, and in most cases, a time-optimal synthesis can be constructed. The situation is different for time-optimal problems in dimension 3. Indeed, besides trivial cases, the time-optimal synthesis was computed in full details for a few examples only. One is the Reeds and Shepp's car,

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = u_1 \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} + u_2 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad |u_1|, |u_2| \leq 1. \quad (7)$$

The time-optimal synthesis for this problem was computed in Soueres and Laumond (1996). The extreme complexity of the optimal synthesis obtained for this simple example had the effect that no other time-optimal synthesis in dimension 3 or larger, with one or two bounded controls, were computed up to the last 7 years.

Very recently, the interest in time-optimal synthesis for systems of the type

$$\dot{q} = \sum_{i=1}^m u_i F_i(q), \quad |u_i| \leq 1, \quad (i = 1, \dots, m) \quad (8)$$

where q belongs to a n -dimensional manifold and $2 \leq m \leq n$ has attracted new attention.

This is indeed a problem of nonstrictly convex sub-Finsler geometry that appears in the study of asymptotic cones of nilpotent groups in geometric group theory (Barilari et al. 2017; Gromov 1981; Breuillard and Le Donne 2012).

Sub-Riemannian Geometry

A very important class of optimal control problems is the one called sub-Riemannian. Let M be a n -dimensional manifold ($n \geq 2$), and consider the problem of finding the time-optimal synthesis starting from a point q_0 for the problem

$$\dot{q} = \sum_{i=1}^m u_i F_i(q), \quad \int_0^1 \sqrt{\sum_{i=1}^m u_i^2} dt \rightarrow \min, \quad (m \geq 2). \quad (9)$$

Here we assume that the family of vector fields $\{F_i\}_{i=1, \dots, m}$ is Lie-bracket generating. This kind of optimal-control problem includes Riemannian geometry and many of its generalizations that

usually take the name of sub-Riemannian geometry (see Bellaïche 1996, Montgomery 2002, Agrachev et al. 2019, and the pioneering work by Brockett 1982). The complete time-optimal synthesis was computed in a few relevant cases:

- The Heisenberg group (Gaveau 1977; Gershkovich and Vershik 1988).
- The local three-dimensional contact case, under generic conditions (Agrachev 1996; El-Alaoui et al. 1996).
- Some relevant left-invariant problem on simple Lie groups, i.e., $SO(3)$, $SU(2)$, $Sl(2)$ (see Boscaïn and Rossi 2008).
- The left-invariant problem on the group of rototranslation $SE(2)$ that has important applications in models of geometry of vision (Boscaïn et al. 2012; Sachkov 2011; Petitot 2008).
- In dimension bigger than 3, only the quasi-Heisenberg case (Charlot 2002) and certain multidimensional generalizations of the Heisenberg case have been computed (Beals et al. 1996).
- In dimension 2, problems of type (8) are called problems of *almost-Riemannian geometry*. The basic example (the so-called Grushin case) was studied in Bellaïche (1996) and the study of the synthesis in the generic case, permitted to obtain some generalizations of the Gauss-Bonnet theorem (Agrachev et al. 2008).

Some of the syntheses mentioned above permitted to obtain important results for the theory of hypoelliptic operators (Hormander 1967). Moreover, they permitted to clarify the relation between small-time heat kernel asymptotics and the properties of the value function for the problem (9). See, for instance, Barilari et al. (2012) and references therein.

Remark 3 It should be mentioned that, due to its specific structure, for sub-Riemannian geometry, the general theory for optimal syntheses described above has never been applied. Instead a generalization of a classical technique due

to Hadamard, used in Riemannian geometry, appears of more direct application. See Agrachev et al. 2019, Cap. 1.3.2.

Connections with the Stabilization Problem

Consider now the control system $\dot{q}(t) = f(q(t), u(t))$, under hypothesis (H). Fix $q_0 \in M$, and assume that there exists $u_0 \in U$ such that $f(q_0, u_0) = 0$. A stabilization problem can be stated as follows:

(P): For every $\bar{q} \in M$, find a trajectory of the control system $\dot{q}(t) = f(q(t), u(t))$, with boundary conditions $q(0) = \bar{q}$, $q(T) = q_0$. (Here T could be required to be fixed or free, finite or not, depending on the problem.)

An elegant way of giving a solution to the problem (P) is to give a stabilizing feedback, namely, a function $K(q)$ such that for every $\bar{q} \in M$ the solution of

$$\dot{q}(t) = f(q(t), K(q(t))), \quad q(0) = \bar{q}, \quad (10)$$

steers $\bar{q}$ to q_0 .

It is well-known that in general it is not possible to give the solution to (P) in feedback form. Indeed there may be topological constraints (in the sense of Brockett, see, for instance, Brockett 1983) that prevent such a feedback to be continuous. Hence, in general, one cannot guarantee existence and uniqueness of classical or Caratheodory solutions to the ODE (10). This problem attracted a lot of attention since the pioneering work of Brockett, and several approaches have been proposed, e.g., via generalized concept of solutions, patchy feedback, time-varying feedback, etc. (see, for instance, Clarke et al. 1997; Ancona and Bressan 1999; Coron 1992).

Sometimes one considers an “optimal control” variant of the problem (P):

(Po): For every $\bar{q} \in M$, find the trajectory of the control system $\dot{q}(t) = f(q(t), u(t))$

(under hypothesis (H)), minimizing the cost $\int_0^T L(q(t), u(t)) dt$, with boundary conditions $q(0) = \bar{q}$, $q(T) = q_0$.

The cost can be an additional constraint given by the problem or can be added artificially to have a method and a good concept of solution to solve problem (P).

A way of giving the solution to problem (Po) (and hence to (P)) is to find the optimal synthesis starting from q_0 for the problem

(-Po): for every $\bar{q} \in M$, solve

$$\begin{cases} \dot{q} = -f(q, u), & u \in U \\ \int_0^T L(q(t), u(t)) dt \rightarrow \min \\ q(0) = q_0, q(T) = \bar{q}, \end{cases} \quad (11)$$

and then to reverse the time. In other words if $\gamma : [0, T] \rightarrow M$ is the solution of (-Po) steering q_0 to $\bar{q}$, then $\gamma(T-t)$ is the solution to (Po) steering $\bar{q}$ in q_0 .

If

$$S_{q_0} = \{\gamma_q(\cdot) : [0, T_q] \rightarrow M\}$$

is an optimal synthesis for (-Po), then

$$S_{q_0}^- = \{\gamma_q(T_q - t) : [0, T_q] \rightarrow M\}$$

is called an “optimal stabilizing synthesis” for (Po).

Extracting a Feedback from an Optimal Synthesis

It is interesting to see what happens if one tries to extract a feedback from an optimal stabilizing synthesis.

If each optimal trajectory of the optimal synthesis corresponds to a regular enough control (e.g., smooth or piecewise), the feedback corresponding to the optimal synthesis can be defined easily in the following way: if $(\gamma(\cdot), u(\cdot))$ defined in $[0, T]$ is a pair trajectory-control of the optimal synthesis, then $K(\gamma(t)) = u(t)$ for every $t \in [0, T]$.

However, as already mentioned, in most of the situations, $K(q)$ is not continuous. (Notice that

even in the case in which all trajectories of the optimal synthesis are smooth, it may happen that $K(q)$ is not continuous.) Hence, in general, one cannot guarantee existence and uniqueness of classical or Caratheodory solutions to the ODE (10).

One could think of enlarging the concept of the solution of (10) by using Filippov, Krasowski, or CLSS (Clarke et al. 1997) solutions (see, for instance, Marigo and Piccoli 2002, Piccoli and Sussmann 2000, and references therein). However none of these types of solutions are adapted to give the solution of an optimal stabilization problem in feedback form. To fix the ideas, let us consider the case of Filippov solutions. In Piccoli and Sussmann (2000), the authors build examples of optimal synthesis for which the corresponding feedbacks generate solutions that are either Filippov but non-optimal or optimal but not Filippov. The same can be done with the other types of solutions mentioned above. Also, it is possible to build an example showing an optimal stabilizing synthesis for which the corresponding feedback generates non-optimal trajectories even in classical sense. This is presented in the next section.

Hence, at the moment, an optimal stabilizing synthesis remains the only possible concept of solution for an optimal stabilizing problem.

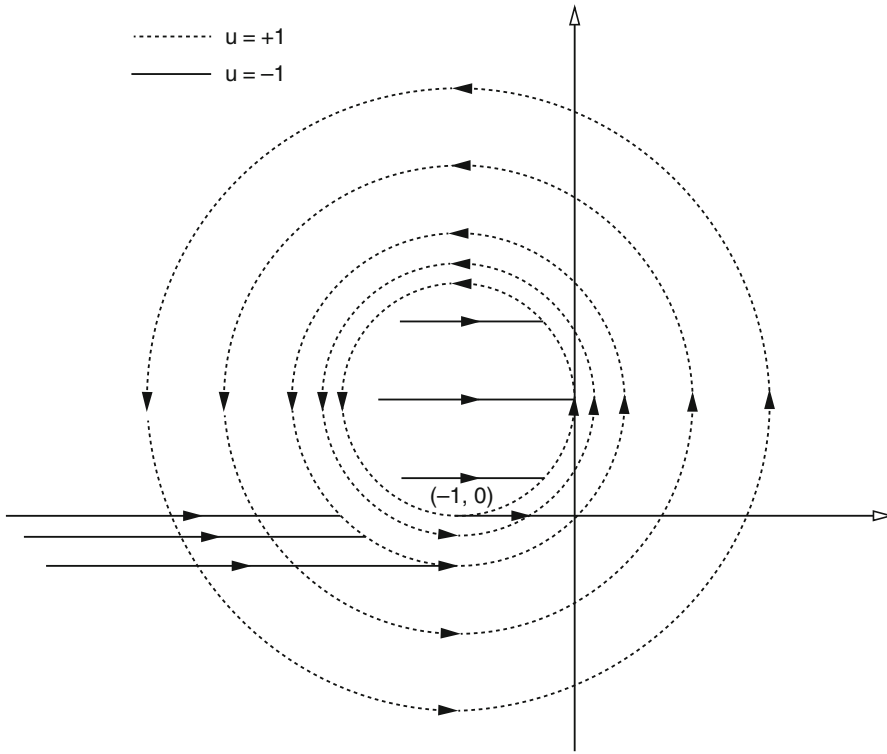
An Example of a Time-Optimal Synthesis Whose Feedback Generates Non-optimal Trajectories

We present an example exhibiting the phenomenon of non-uniqueness of trajectories for the closed-loop equation arising from the feedback extracted from an optimal synthesis. In particular the optimal feedback admits non-optimal (classical) solutions. This well illustrates the importance of using the synthesis as concept of solution for an optimal stabilization problem.

Consider the planar system:

$$\dot{q} = F(q) + u G(q), \quad |u| \leq 1,$$

where $q = (x, y)$ and:



Synthesis Theory in Optimal Control, Fig. 1 An optimal stabilizing synthesis for which the corresponding feedback generates non-optimal trajectories

$$F(q) = \left(1 - \frac{y}{2} \right), \quad G(q) = \left(\frac{-y}{x+1} \right),$$

and the target is the origin.

The trajectories corresponding to the constant control equal to -1 are straight horizontal lines going from left to right, while those corresponding to $+1$ are circles centered at the point $(-1, 1)$, running counterclockwise. The optimal synthesis is described in Fig. 1. For a proof of optimality, see Piccoli and Sussmann (2000).

Starting from the point $(-1, 0)$, we have an infinite number of classical solutions to the discontinuous optimal feedback. Indeed at that point, we have $F + G = F - G$, so given any natural number n , the trajectory running n times on the circle centered at $(-1, 1)$ and then going to the origin with control -1 is a classical solution to the discontinuous optimal feedback. However, only the one corresponding to $n = 0$ is optimal.

Concerning other concepts of solutions starting from $(-1, 0)$, one can prove the following. Krasowski or CLSS includes classical solutions (and hence produces many non-optimal trajectories). There is only one Filippov solution, that is, the one that rotates indefinitely on the circle and never goes to the origin. This trajectory is not a solution to the stabilization problem since it does not reach the target.

Cross-References

- [Optimal Control and Mechanics](#)
- [Optimal Control and Pontryagin's Maximum Principle](#)
- [Sub-Riemannian Optimization](#)

Acknowledgments The first author has been supported by the ANR projects SRGI ANR-15-CE40-0018 and Quaco ANR-17-CE40-0007-01.

Bibliography

- Agrachev A (1996) Exponential mappings for contact sub-Riemannian structures. *J Dyn Control Syst* 2(3): 321–358
- Agrachev AA, Sachkov YuL (2004) Control theory from the geometric viewpoint. *Encyclopedia of mathematical sciences*, vol 87. Springer, Berlin/New York
- Agrachev A, Boscaïn U, Sigalotti M (2008) A Gauss-Bonnet-like formula on two-dimensional almost-Riemannian manifolds. *Discret Contin Dyn Syst A* 20:801–822
- Agrachev A, Barilari D, Boscaïn U (2019) A comprehensive introduction to sub-Riemannian geometry. *Cambridge studies in advanced mathematics*, vol 181. Cambridge University Press, Cambridge
- Ancona F, Bressan A (1999) Patchy vector fields and asymptotic stabilization. *ESAIM Control Optim Calc Var* 4:445–471
- Barilari D, Boscaïn U, Neel RW (2012) Small time heat asymptotics at the sub-Riemannian cut locus. *J Differ Geom* 92(3):373–416
- Beals R, Gaveau B, Greiner P (1996) The Green function of model step two hypoelliptic operators and the analysis of certain tangential Cauchy Riemann complexes. *Adv Math* 121(2):288–345
- Bellaïche A (1996) The tangent space in sub-Riemannian geometry. In: Bellaïche A, Risler J-J (eds) *Sub-Riemannian geometry*. *Progress in mathematics*, vol 144. Birkhäuser, Basel, pp 1–78
- Bloch A (2003) Nonholonomic mechanics and control. *Interdisciplinary applied mathematics*, vol 24. Springer, New York
- Boltyanskii V (1966) Sufficient condition for optimality and the justification of the dynamic programming principle. *SIAM J Control Optim* 4:326–361
- Boscaïn U, Piccoli B (2004) Optimal synthesis for control systems on 2-D manifolds. *SMAI*, vol 43. Springer, Berlin/New York
- Boscaïn U, Rossi F (2008) Invariant Carnot-Carathéodory metrics on S^3 , $SO(3)$, $SL(2)$ and Lens Spaces. *SIAM J Control Optim* 47:1851–1878
- Boscaïn U, Duplaix J, Gauthier JP, Rossi F (2012) Anthropomorphic image reconstruction via hypoelliptic diffusion. *SIAM J Control Optim* 50(3):1309–1336
- Barilari D, Boscaïn U, Le Donne E, Sigalotti M (2017) Sub-Finsler structures from the time-optimal control viewpoint for some nilpotent distributions. *J Dyn Control Syst* 23(3):547–575
- Breuillard E, Le Donne E (2012) On the rate of convergence to the asymptotic cone for nilpotent groups and sub-Finsler geometry. *PNAS*. <https://doi.org/10.1073/pnas.1203854109>
- Bressan A (1985) A high order test for optimality of bang-bang controls. *SIAM J Control Optim* 23(1): 38–48
- Bressan A, Piccoli B (1998) A generic classification of time optimal planar stabilizing feedbacks. *SIAM J Control Optim* 36(1):12–32
- Bressan A, Piccoli B (2007) Introduction to the mathematical theory of control. *AIMS series on applied mathematics*, vol 2. American Institute of Mathematical Sciences, Springfield
- Brockett R (1982) Control theory and singular Riemannian geometry. In: *New directions in applied mathematics* (Cleveland, 1980). Springer, New York/Berlin, pp 11–27
- Brockett R (1983) Asymptotic stability and feedback stabilization. In: Brockett RW, Millman RS, Sussmann HJ (eds) *Differential geometric control theory*. Birkhäuser, Boston, pp 181–191
- Brunovsky P (1978) Every normal linear system has a regular time-optimal synthesis. *Math Slovaca* 28: 81–100
- Brunovsky P (1980) Existence of regular syntheses for general problems. *J Differ Equ* 38:317–343
- Cesari L (1983) Optimization-theory and applications: problems with ordinary differential equations. Springer, New York
- Clarke F, Ledyaev Yu, Subbotin A, Sontag E (1997) Asymptotic controllability implies feedback stabilization. *IEEE Trans Autom Control* 42:1394–1407
- Charlot G (2002) Quasi-contact S-R metrics: normal form in $\mathbb{R}^{2n}$, wave front and caustic in $\mathbb{R}^4$. *Acta Appl Math* 74(3):217–263
- Coron JM (1992) Global asymptotic stabilization for controllable systems without drift. *Math Control Signals Syst* 5:295–312
- Dubins LE (1957) On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal position and tangents. *Am J Math* 79:497–516
- El-Alaoui El-H Ch, Gauthier J-P, Kupka I (1996) Small sub-Riemannian balls on $\mathbb{R}^3$. *J Dyn Control Sys* 2(3):359–421
- Gaveau B (1977) Principe de moindre action, propagation de la chaleur et estimées sous elliptiques sur certains groupes nilpotents. *Acta Math* 139(1–2):95–153
- Gershkovich V, Vershik A (1988) Nonholonomic manifolds and nilpotent analysis. *J Geom Phys* 5:407–452
- Gromov M (1981) Groups of polynomial growth and expanding maps. *Inst Hautes Études Sci Publ Math* 53:53–73
- Hörmander L (1967) Hypoelliptic second order differential equations. *Acta Math* 119:147–171
- Krener AJ (1977) The high order maximal principle and its application to singular extremals. *SIAM J Control Optim* 15(2):256–293
- Marigo A, Piccoli B (2002) Regular syntheses and solutions to discontinuous ODEs. *ESAIM Control Optim Calc Var* 7:291–308
- Montgomery R (2002) A tour of sub-Riemannian geometries, their geodesics and applications. *Mathematical surveys and monographs*, vol 91. American Mathematical Society, Providence
- Petitot J (2008) Neurogéométrie de la vision, Modèles mathématiques et physiques des architectures fonctionnelles. *Les Éditions de l'École Polytechnique*

- Piccoli B (1996) Classifications of generic singularities for the planar time-optimal synthesis. *SIAM J Control Optim* 34(6):1914–1946
- Piccoli B, Sussmann HJ (2000) Regular synthesis and sufficiency conditions for optimality. *SIAM J Control Optim* 39(2):359–410
- Pontryagin LS et al (1961) *The mathematical theory of optimal processes*. Wiley, New York
- Reeds JA, Shepp LA (1990) Optimal Path for a car that goes both forwards and backwards. *Pac J Math* 145:367–393
- Sachkov Yu (2011) Cut locus and optimal synthesis in the sub-Riemannian problem on the group of motions of a plane. *ESAIM COCV* 17:293–321
- Sigalotti M, Chitour Y (2006) Dubins' problem on surfaces II: nonpositive curvature. *SIAM J Control Optim* 45:457–482
- Soueres P, Laumond JP (1996) Shortest paths synthesis for a car-like robot. *IEEE Trans Autom Control* 41(5):672–688
- Sussmann HJ (1979) Subanalytic sets and feedback control. *J Differ Equ* 31(1):31–52
- Sussmann HJ (1980) Analytic stratifications and control theory. In: *Proceedings of the international congress of mathematicians (Helsinki, 1978)*, Academia Scientiarum Fennica, Helsinki, pp 865–871
- Sussmann HJ (1986) Envelopes, conjugate points, and optimal bang-bang extremals. In: *Algebraic and geometric methods in nonlinear control theory. Mathematics and its applications*, vol 29. Reidel, Dordrecht, pp 325–346
- Sussmann HJ (1989) Envelopes, higher-order optimality conditions and Lie Brackets. In: *Proceedings of the 1989 IEEE conference on decision and control*, Tampa
- Vinter R (2010) *Optimal control*. Birkhäuser, Basel/Boston

Synthetic Biology

Domitilla Del Vecchio
Department of Mechanical Engineering,
Massachusetts Institute of Technology,
Cambridge, MA, USA

Abstract

Synthetic biology, that is, the ability to engineer biology for useful functionalities, will have remarkable impact in a number of applications ranging from health and medicine to environment and energy. Examples include engineered bacteria that recognize and kill

cancer cells, neutralize radioactive waste, and transform feedstock into fuel and programmable mammalian cells that control the differentiation of tissue in vivo. Engineering biological circuits that are sufficiently sophisticated to accomplish these functions are becoming possible but at the same time are proving challenging due to a number of obstacles. Many of these obstacles require a system-level understanding of the dynamical and robustness properties of interacting systems, and hence the field of control and dynamical systems theory may highly contribute. In this entry, we review the basic technology employed in engineering biology, simple example modules, and complex systems created using this technology and discuss key system-level problems along with challenging research questions for the field of control theory.

Keywords

Biomolecular systems · Gene expression · Robustness · Scalability · Modularity

Introduction to Synthetic Biology

Synthetic biology is an engineering discipline in which the biochemical and biophysical principles present in living organisms are used to engineer new systems (Cameron et al. 2014). These systems will have the ability to accomplish a number of remarkable tasks, such as turning waste into energy sources (Peralta-Yahya et al. 2012; Zhang et al. 2012), detecting pathogens (Pardee et al. 2016), or recognizing cancer cells with the aim of targeting them for deletion (Danino et al. 2017). While synthetic biology can be employed to create new functionalities, it can also enable the understanding of fundamental design principles of living systems. In fact, implementing a circuit with a prescribed behavior provides a powerful means to test hypotheses regarding the underlying biological mechanisms.

The functions of living organisms are controlled by biomolecular circuits, in which, at

the most basic level, proteins and genes interact with each other through activation and repression interactions forming complex networks. A common signal carrier is the concentration of the active form of a protein, which can be controlled through a number of mechanisms, including gene expression regulation and posttranscriptional and posttranslational modification. Through the process of gene expression, proteins are produced by their corresponding genes, whose production rates can be activated or repressed by other proteins (transcription factors). Once the proteins are produced, they can be activated or inhibited, by other proteins or small molecules, through posttranscriptional modification of mRNA and posttranslational modification to proteins, such as phosphorylation, and allosteric modification (Alon 2007; Del Vecchio and Murray 2014). We next describe some salient aspects of gene expression focusing, for simplicity, on prokaryotic systems.

A gene is a piece of DNA whose expression rate can often be controlled by a DNA sequence upstream of the gene itself, called promoter. The promoter contains the binding regions for the RNA polymerase, an enzyme that transcribes the gene into a messenger RNA molecule, which is then translated into protein by the ribosomes (central dogma of molecular biology Alberts et al. 2007). The promoter also contains operator sites, which are binding regions where other proteins, called transcription factors, can bind. If these proteins are activators, they will help the RNA polymerase in binding the promoter to start transcription. By contrast, if these proteins are repressors, they will prevent the RNA polymerase from binding the promoter. These activation and repression interactions are nonlinear and stochastic; therefore the most commonly used modeling frameworks include systems of nonlinear ordinary differential equations, stochastic differential equations, or the chemical master equation (Del Vecchio and Murray 2014).

The basic technique for constructing synthetic circuits is that of assembling, through the process of cloning, DNA sequences with prescribed combinations of promoters and genes such that a desired network of activation and repression

interaction is created. For example, if we would like to create an “inverter” where protein A represses protein B, we can simply place the gene of B under the control of a promoter repressed by protein A. Currently, there is an increasing library of parts that one can use to assemble a desired circuit this way ([BioBricks Foundation](#); [Registry of Standard Biological Parts](#)). The set of parts includes promoters, gene-coding sequences, terminators, and ribosome binding sites. Terminators are DNA sequences placed at the end of a gene to make the RNA polymerase terminate transcription, while ribosome binding sites are DNA sequences placed at the beginning of a gene, which establish the rate at which ribosomes will bind to the mRNA, determining the overall translation rate (Del Vecchio and Murray 2014). An area of intense research is expansion of the library by creating mutations of existing parts or by assembling new ones.

Once a DNA sequence is created that encodes the desired circuit, it is inserted in a cell either on the chromosome itself or on DNA plasmids. Once in the cell, the circuit can “run” using the cellular resources required for gene expression, including RNA polymerase and ribosomes, amino acids, and ATP. Alternatively, a cell extract can be created for running the circuits in cell-free setups (Perez et al. 2016; Pardee et al. 2014), a technology that has open the way to a number of applications, including diagnostic (Pardee et al. 2016; Takahashi et al. 2018). In this sense, the cell, or the cell extract, can be viewed as a chassis for the synthetic genetic circuits. The operation of the circuit can then be observed by monitoring the concentration of reporters, that is, of proteins that are easy to detect and quantify. These include fluorescent proteins, that is, proteins that exhibit bright fluorescence when exposed to light of a specific wavelength. Examples include the green, red, blue, and yellow fluorescent proteins. These fluorescent proteins are mainly employed in two different ways to measure the amount of a protein of interest. Specifically, one can fuse the gene of the fluorescent protein with the gene expressing the protein of interest. Alternatively, one can use the protein of interest as a transcription factor of the fluorescent protein. In both cases, the con-

centration of the fluorescent protein will provide an indirect measurement of the concentration of the protein of interest.

It is also possible to apply external inputs to a circuit to control the activity of transcription factors. This is accomplished through the use of inducers, which are small signaling molecules that can be injected in the cell culture and enter the cell wall. These inducers bind specific transcription factors and either activate them, allowing the transcription factor to bind the promoter operator sites, or inhibit them, reducing the transcription factor ability to bind the promoter operator sites (Meyer et al. 2019).

Early Synthetic Biology Modules

A number of modules comprising two or three genes have been fabricated in the earlier days of synthetic biology (Atkinson et al. 2003; Becskei and Serrano 2000; Elowitz and Leibler 2000; Gardner et al. 2000; Stricker et al. 2008). We can group them into oscillators (Atkinson et al. 2003; Elowitz and Leibler 2000; Stricker et al. 2008), mono-stable systems (Becskei and Serrano 2000), and bistable systems called toggle switches (Gardner et al. 2000).

Oscillators The creation of circuits whose protein concentrations oscillate periodically in time has been a major focus. In fact, the ability of creating an oscillator has the potential of shedding light into the mechanisms at the basis of natural clocks, such as circadian rhythms and the cell cycle. Oscillator designs can be divided into two types: loop oscillators (Eloitz and Leibler 2000), in which repression/activation interactions occur in a loop topology or oscillators based on the interplay between an autocatalytic loop and negative feedback (Atkinson et al. 2003; Stricker et al. 2008) (see Fig. 1).

The design requirements of synthetic circuits are usually explored through models of varying details, starting with the use of low-dimensional models, which are composed of a set of non-linear ordinary differential equations describing the rate of change of the circuit's proteins. These

models allow application of a number of tools from dynamical systems theory to infer parameter or structural requirements for a desired behavior. After simple models are analyzed, larger-scale mechanistic models are constructed, which include all the intermediate species taking part in the biochemical reactions. These models can be either deterministic or stochastic. Simulation is usually required for the study of these more complicated models, and the Gillespie algorithm is often employed for stochastic simulations (Gillespie 1977).

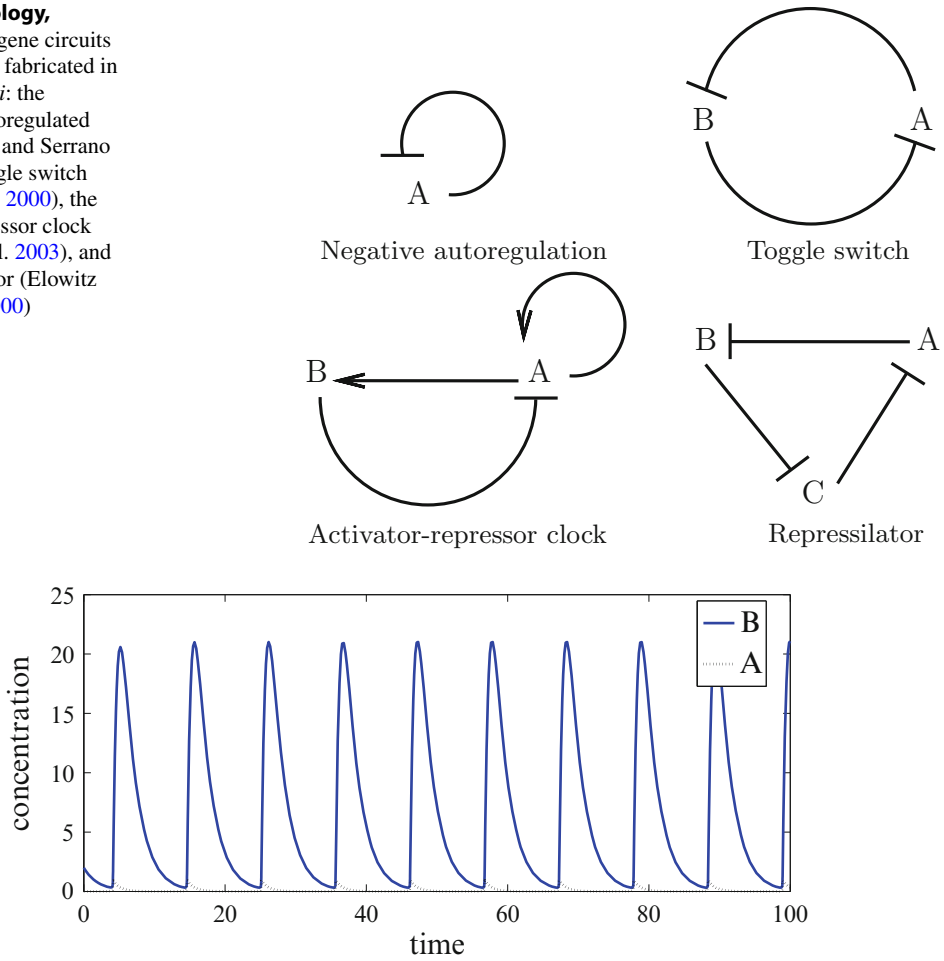
As an example of a simple model and related analysis, consider the activator-repressor clock of Atkinson et al. (2003) shown in Fig. 1. This oscillator is composed of an activator *A* activating itself and a repressor *B*, which, in turn, represses the activator *A*. Both activation and repression occur through transcription regulation. Denoting in italics the concentration of species, a toy model of this clock can be written as

$$\begin{aligned}\dot{A} &= \frac{\beta_A (A/K_a)^n + \beta_{0,A}}{1 + (A/K_a)^n + (B/K_b)^m} - \gamma_A A, \\ \dot{B} &= \frac{\beta_B (A/K_a)^n + \beta_{0,B}}{1 + (A/K_a)^n} - \gamma_B B,\end{aligned}\quad (1)$$

in which γ_A and γ_B represent protein decay (due to dilution and/or degradation). The functions $(\beta_A (A/K_a)^n + \beta_{0,A}) / (1 + (A/K_a)^n + (B/K_b)^m)$ and $(\beta_B (A/K_a)^n + \beta_{0,B}) / (1 + (A/K_a)^n)$ are called Hill functions and are the most commonly used models for transcription regulation (Del Vecchio and Murray 2014). The first Hill function in system (1) increases with *A* and decreases with *B*, while the second one increases with *A*, as expected since *A* is an activator and *B* is a repressor. The key mechanism by which this system displays sustained oscillations is a supercritical Hopf bifurcation with bifurcation parameter the relative time scale of the activator dynamics with respect to the repressor dynamics (Del Vecchio and Murray 2014). Specifically, as the activator dynamics become faster than the repressor dynamics, the system goes through a supercritical Hopf bifurcation, and a stable periodic orbit appears (Fig. 2).

Synthetic Biology,

Fig. 1 Early gene circuits that have been fabricated in bacteria *E. coli*: the negatively autoregulated gene (Becskei and Serrano 2000), the toggle switch (Gardner et al. 2000), the activator-repressor clock (Atkinson et al. 2003), and the repressilator (Elowitz and Leibler 2000)



Synthetic Biology, Fig. 2 Activator-repressor clock time trajectory

Mono-stable systems The first mono-stable system engineered through negative autoregulation was fabricated with the aim of understanding the role of negative feedback in attenuating biological noise. The results of Becskei and Serrano (2000) clearly showed that negative autoregulation can reduce intrinsic noise. Furthermore, the results of Austin et al. (2005) demonstrated that while low-frequency noise is attenuated, noise at high frequency can be amplified by negative autoregulation in accordance with Bode's integral formula (Åström and Murray 2008).

Bistable systems The toggle switch of Gardner et al. (2000) was the first bistable system con-

structed. It constitutes the simplest circuit with memory, in which the state of the system can be switched from one equilibrium (low, high) to the other (high, low) by external inputs. Once the system state is switched to one of these two equilibria, it will stay there unless another external perturbation is applied. Based on this circuit, a number of applications have been developed, including systems to control endogenous cell factors (Kobayashi et al. 2004) and kill switches for programmed cell death (Chan et al. 2016) and programmable bacteria that detect and record events in the guts (Kotulaa et al. 2014).

While the early circuits described so far were fabricated mainly to investigate design principles for limit cycles, memory, and robustness to noise,

many more circuits after these have been fabricated with the aim of solving concrete engineering problems, including incoherent feedforward loops for robustness to DNA copy number (Bleris et al. 2011; Segall-Shapiro et al. 2018), logic gates (Moon et al. 2012), communication modules (Chen and Weiss 2005), load drivers (Mishra et al. 2014; Nilgiriwala et al. 2015), and various types of feedback controllers (Hsiao et al. 2015; Kelly et al. 2018; Aoki et al. 2019; Shopera et al. 2017; Huang et al. 2018; Lillacci et al. 2018). For a more extensive review, the reader is referred to (Del Vecchio et al. 2016; Qian et al. 2018; Hsiao et al. 2018; McBride et al. 2019).

From Modules to Systems: Opportunities and Challenges

One approach to creating systems that can accomplish sophisticated tasks is to assemble together simpler modules, such as those described in the previous section (Purnick and Weiss 2009). Increasingly large systems are being built by composing modules together in a layered architecture (Moon et al. 2012), and software tools such as Cello are being developed in order to automate the process that goes from design concept to choice of genetic parts (Nielsen et al. 2016). Layered logic gates are often necessary in order to integrate multiple signals in several applications. One such application is a cell type classifier based on RNA signatures (Xie et al. 2011). Here, a synthetic gene circuit is created that integrates sensory information from a number of molecular markers to determine whether a cell is in a specific state, that is, cancer, and, in such a case, produces a protein output triggering cell death. The design of this circuit is based on the composition of three key modules. Specifically, a double inversion module senses high levels of a molecular marker, a single inversion module senses low levels of a molecular marker, and a logical “and” module finally integrates the outputs of the other two modules to produce the output protein.

While increasingly sophisticated circuits composed of multiple modules are being built, the

behavior of these circuits becomes more difficult to predict and engineer as their size increases. A major problem is that circuits characterized in isolation often fail to behave as intended once they are part of a larger system. This type of failure is commonly referred to as *context dependence* of genetic circuits (Cardinale and Arkin 2012; Del Vecchio 2015), that is, the fact that modules behave in a poorly predictable way once interacting together in the cell environment. This is a bottleneck to creating larger circuits that behave predictably.

Problems of context dependence can be divided into three qualitatively different types: genetic context; direct inter-module interactions; and indirect interactions among genetic modules. While the first one can be dealt with via appropriate genetic engineering of DNA parts (Del Vecchio 2015; Yeung et al. 2017; Meyer et al. 2019), the other two forms of context dependence require a system-level understanding of dynamic interactions among circuit components. We therefore focus mostly on the latter two aspects here.

Direct inter-module interactions These are unwanted interactions that occur directly between modules. A well-known example is that of off-target binding of transcription factors to promoters. These undesirable bindings couple genetic modules that should be decoupled and can be tackled by co-optimizing the genetic sequences of promoters and transcription factors (Meyer et al. 2019). A different type of undesirable interaction occurs when modules are connected to each other and a protein in an upstream module is used as an “input” to a downstream module. This creates a “load” on the upstream system due to the fact that the output protein cannot take part in the upstream module’s reactions whenever it is taking part in the downstream module’s reactions. As a consequence, the behavior of the upstream module changes compared to when the system functions in isolation (Del Vecchio et al. 2008; Jayanthi et al. 2013). These loading effects have been called retroactivity to extend the notion of loading and impedance to biomolecular systems.

Solutions to mitigate this problem have appeared (Nilgiriwala et al. 2015; Mishra et al. 2014). These are based on the design of “insulation devices,” systems that are placed between a sending module and receiving load modules such that arbitrarily large loads can be driven without a deterioration of the transmitted signal. The design of these load drivers uses principles of disturbance attenuation from control theory in order to reject retroactivity (Jayanthi and Del Vecchio 2011).

Indirect interactions among genetic modules

Ideally, the cell should function as a “chassis” for engineered biological circuits. In practice, this is not the case because the cellular endogenous circuitry interacts with externally introduced circuits at multiple levels, chiefly by sequestering resources such as ATP, RNA polymerase, and ribosomes, which are required for the operation of synthetic circuits. This sequestration often reduces cell fitness, with deleterious consequences also for synthetic circuits, a phenomenon that has been broadly called “metabolic burden” (Bentley et al. 1990). This phenomenon has been studied and characterized more precisely in recent years, by showing that it is possible to create a “burden monitor” that senses the level of stress by cells and correlates with growth rate (Ceroni et al. 2015). A more subtle phenomenon than purely reducing cell fitness is that synthetic circuits compete with each other for the same resources. This fact creates implicit and unwanted coupling among circuits with unpredictable consequences. Specifically, it was demonstrated that competition for shared gene expression resources, chiefly ribosomes, by multiple synthetic genes, couples the expression levels of these genes in such a way that inducing one gene causes a drop in the expression of the other (Gyorgy et al. 2015). These “hidden” interactions are a major cause of lack of modularity in the design of genetic circuits and have been shown to lead to emergent genetic circuit behaviors that are arbitrarily far from the theoretically prescribed one (Qian et al. 2017).

Approaches to mitigate these problems have recently appeared in the literature. Specifically, in order to mitigate defects on cell growth rate imparted by induction of synthetic genes, a control system has been built, which uses the burden monitor to sense burden and, based on this, reduces the expression of synthetic genes (Ceroni et al. 2018). Concurrently, solutions have appeared to mitigate the coupling among synthetic genetic circuits due to competition for ribosomes. Two types of solutions have appeared. The first type is a decentralized control architecture in which each genetic module is equipped with a feedback controller that aims at making output protein level robust to fluctuations in available ribosomes (Shopera et al. 2017; Huang et al. 2018). Specifically, in Huang et al. (2018), the authors built a quasi-integral controller that is embedded within a genetic module via posttranscriptional modifications, which allows to keep the inputs and outputs of the genetic module unchanged while reaching adaptation to variable ribosome demand. In the second type of solution, a centralized controller was proposed to dynamically allocate orthogonal ribosomes to synthetic genetic circuits (Darlington et al. 2018).

Finally, the external environment where a cell operates has a number of physical attributes, which may also be subject to perturbations. These physical attributes include temperature, acidity, nutrients’ level, etc. Perturbations in these attributes often lead to poor cell fitness or to non-standard growth conditions, ultimately leading to synthetic circuits malfunctions. Recently, a universal integral control architecture has been proposed to mitigate some of these disturbances, such as growth rate changes (Aoki et al. 2019).

Summary and Future Directions

The future of synthetic biology highly depends on the ability of scaling up the complexity of design to create more sophisticated functions, which yet are robust and reliable. While a number of issues can be successfully addressed by (nontrivial) fabrication of new parts, issues such

as context dependence require a system-level dynamic understanding of circuits and their interactions. Here is where control and dynamical systems theory could greatly contribute. Control theory has proven critical to reason about and engineer robustness in a number of concrete applications including aerospace and automotive systems, robotics and intelligent machines, manufacturing chains, electrical, power, and information networks. Similarly, control theory could enable the understanding of principles that ensure robust behavior of synthetic genetic circuits. An extensive review on this topic can be found in Del Vecchio et al. (2018).

The author would like to thank Richard M. Murray for having provided feedback on the initial form of the current manuscript.

Cross-References

- [Reverse Engineering and Feedback Control of Gene Networks](#)

Bibliography

- BioBricks Foundation. <https://biobricks.org>
Registry of Standard Biological Parts. <http://parts.igem.org>
- Åström KJ, Murray RM (2008) *Feedback Systems*. Princeton University Press, Princeton
- Alberts B, Johnson A, Lewis J, Raff M, Roberts K (2007) *Molecular biology of the cell*, 5th edn. Garland Science, New York
- Alon U (2007) *An introduction to systems biology. Design principles of biological circuits*. Chapman-Hall, Boca Raton
- Aoki SK, Lillacci G, Gupta A, Baumschlager A, Schweingruber D, Khammash M (2019) A universal biomolecular integral feedback controller for robust perfect adaptation. *Nature* 570:533–537
- Atkinson MR, Savageau MA, Meyers JT, Ninfa AJ (2003) Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in *Escherichia coli*. *Cell* 113:597–607
- Austin DW, Allen MS, McCollum JM, Dar RD, Wilgus JR, Sayler GS, Samatova NF, Cox CD, Simpson ML (2005) Gene network shaping of inherent noise spectra. *Nature* 439:608–611
- Becskei A, Serrano L (2000) Engineering stability in gene networks by autoregulation. *Nature* 405:590–593
- Bentley WE, Mirjalili N, Andersen DC, Davis RH, Kompala DS (1990) Plasmid-encoded protein: the principal factor in the “metabolic burden” associated with recombinant bacteria. *Biotechnol Bioeng* 35(7):668–81
- Bleris L, Xie Z, Glass D, Adadey A, Sontag E, Benenson Y (2011) Synthetic incoherent feedforward circuits show adaptation to the amount of their genetic template. *Mol Syst Biol* 7:519
- Cameron DE, Bashor CJ, Collins JJ (2014) A brief history of synthetic biology. *Nat Rev Microbiol* 12:381–390
- Cardinale S, Arkin AP (2012) Contextualizing context for synthetic biology – identifying causes of failure of synthetic biological systems. *Biotechnol J* 7:856–866
- Ceroni F, Algar R, Stan GB, Ellis T (2015) Quantifying cellular capacity identifies gene expression designs with reduced burden. *Nat Methods* 12(5):415–422. <https://doi.org/10.1038/nmeth.3339>
- Ceroni F, Boo A, Furini S, Gorochowski TE, Borkowski O, Ladak YN, Awan AR, Gilbert C, Stan GB, Ellis T (2018) Burden-driven feedback control of gene expression. *Nat Methods* 15:387–393
- Chan CT, Lee JW, Cameron DE, Bashor CJ, Collins JJ (2016) ‘deadman’ and ‘passcode’ microbial kill switches for bacterial containment. *Nat Chem Biol* 12:82–86
- Chen M-T, Weiss R (2005) Artificial cell-cell communication in yeast *Saccharomyces cerevisiae* using signaling elements from *Arabidopsis thaliana*. *Nat Biotechnol* 23(12):1551–1555
- Danino T, Prindle A, Kwong GA, Skalak M, Li H, Allen K, Hasty J, Bhatia SN (2017) Programmable probiotics for detection of cancer in urine. *Sci Transl Med* 7(289):289ra84
- Darlington APS, Kim J, Jiménez JI, Bates DG (2018) Dynamic allocation of orthogonal ribosomes facilitates uncoupling of co-expressed genes. *Nat Commun* 9(1):695
- Del Vecchio D (2015) Modularity, context-dependence, and insulation in engineered biological circuits. *Trends Biotechnol*
- Del Vecchio D, Murray RM (2014) *Biomolecular feedback systems*, 1st edn. Princeton University Press, Princeton
- Del Vecchio D, Ninfa AJ, Sontag ED (2008) Modular cell biology: retroactivity and insulation. *Mol Syst Biol* 4:161
- Del Vecchio D, Dy AJ, Qian Y (2016) Control theory meets synthetic biology. *J R Soc Interface* 13(120). <https://doi.org/10.1098/rsif.2016.0380>
- Del Vecchio D, Qian Y, Murray RM, Sontag ED (2018) Future systems and control research in synthetic biology. *IFAC Rev Control* 45:5–17
- Elowitz MB, Leibler S (2000) A synthetic oscillatory network of transcriptional regulators. *Nature* 403:339–342
- Gardner TS, Cantor CR, Collins JJ (2000) Construction of the genetic toggle switch in *Escherichia Coli*. *Nature* 403:339–342
- Gillespie DT (1977) Exact stochastic simulation of coupled chemical reactions. *J Phys Chem* 81:2340–2361

- Gyorgy A, Jiménez JI, Yazbek J, Huang HH, Chung H, Weiss R, Del Vecchio D (2015) Isocost lines describe the cellular economy of gengine circuits. *Biophys J* 109(3):639–646. <https://doi.org/10.1016/j.bpj.2015.06.034>
- Hsiao V, De Los Santos E, Whitaker WR, Dueber JE, Murray RM (2015) Design and implementation of a biomolecular concentration tracker. *ACS Syn Biol* 4(2):150–161. <https://doi.org/10.1021/sb500024b>
- Hsiao V, Swaminathan A, Murray RM (2018) Control theory for synthetic biology: recent advances in system characterization, control design, and controller implementation for synthetic biology. *IEEE Control Syst Mag* 38:32–62
- Huang HH, Qian Y, Del Vecchio D (2018) A quasi-integral controller for adaptation of genetic modules to variable ribosome demand. *Nat Commun* 9(1). <https://doi.org/10.1038/s41467-018-07899-z>
- Jayanthi S, Del Vecchio D (2011) Retroactivity attenuation in bio-molecular systems based on timescale separation. *IEEE Trans Autom Control* 56:748–761
- Jayanthi S, Nilgiriwala K, Del Vecchio D (2013) Retroactivity controls the temporal dynamics of gene transcription. *ACS Synth Biol*. <https://doi.org/10.1021/sb300098w>
- Kelly CL, Harris A, Steel H, Hancock EJ, Heap JT, Papachristodoulou A (2018) Synthetic negative feedback circuits using engineered small RNAs. *Nucl Acids Res* 46:9875–9889. <https://doi.org/10.1021/sb500024b>
- Kobayashi H, Kaern M, Araki M, Chung K, Gardner TS, Cantor CR, Collins JJ (2004) Programmable cells: interfacing natural and engineered gene networks. *Proc Natl Acad Sci* 101:8414–8419
- Kotulaa JW, Kernsa SJ, Shaketb LA, Sirajb L, Collins JJ, Wayb JC, Silver PA (2014) Programmable bacteria detect and record an environmental signal in the mammalian gut. *Proc Natl Acad Sci USA* 111:4838–4843
- Lillacci G, Benenson Y, Khammash M (2018) Synthetic control systems for high performance gene expression in mammalian cells. *Nucl Acids Res* 46(18):9855–9863. <https://doi.org/10.1093/nar/gky795>
- McBride C, Shah R, Del Vecchio D (2019) The effect of loads in molecular communications. *Proc IEEE* 107:1369–1386
- Meyer AJ, Segall-Shapiro TH, Glassey E, Zhang J, Voigt CA (2019) *Escherichia coli* “marionette” strains with 12 highly optimized small-molecule sensors. *Nat Chem Biol* 15:196–204
- Mishra D, Rivera-Ortiz PM, Lin A, Del Vecchio D, Weiss R (2014) A load driver device to engineer modularity in biological circuits. *Nat Biotechnol* 32:1268–1275
- Moon TS, Lou C, Tamsir A, Stanton BC, Voigt CA (2012) Genetic programs constructed from layered logic gates in single cells. *Nature* 491:249–253
- Nielsen AAK, Der BS, Shin J, Vaidyanathan P, Paralanov V, Strychalsk EA, Ross D, Densmore D, Voigt CA (2016) Genetic circuit design automation. *Science* 352:aac7341
- Nilgiriwala K, Rivera-Ortiz PM, Jimenez J, Del Vecchio D (2015) A tunable amplifying buffer circuit in *e. coli*. *ACS Synth Biol* 4:577–584
- Pardee K, Green A, Ferrante T, Cameron DE, Daleykeyser A, Yin P, Collins JJ (2014) Paper-based synthetic gene networks. *Cell* 159(4):940–954. ISSN 10974172. <https://doi.org/10.1016/j.cell.2014.10.004>
- Pardee K, Green A, Takahashi MK, Braff D, Lambert G, Lee JW, Ferrante T, Ma D, Donghia N, Fan M, Daringer NM, Bosch I, Dudley DM, O'Connor DH, Gehrke L, Collins JJ (2016) Rapid, low-cost detection of Zika virus using programmable biomolecular components. *Cell* 165(5):1255–1266. ISSN 10974172. <https://doi.org/10.1016/j.cell.2016.04.059>
- Peralta-Yahya PP, Zhang F, del Cardayre SB, Keasling JD (2012) Microbial engineering for the production of advanced biofuels. *Nature* 488:320–328
- Perez JG, Stark JC, Jewett MC (2016) Cell-free synthetic biology: engineering beyond the cell. *Cold Spring Harb Perspect Biol* 12:a023853. <https://doi.org/10.1101/cshperspect.a023853>
- Purnick P, Weiss R (2009) The second wave of synthetic biology: from modules to systems. *Nat Rev Mol Cell Biol* 10:410–22
- Qian Y, Huang HH, Jiménez JI, Del Vecchio D (2017) Resource competition shapes the response of genetic circuits. *ACS Synth Biol* 6:1263–1272
- Qian Y, McBride C, Del Vecchio D (2018) Programming cells to work for us. *Ann Rev Control Robot Auton Syst* 1(1). <https://doi.org/10.1146/annurev-control-060117-105052>
- Segall-Shapiro TH, Sontag ED, Voigt CA (2018) Engineered promoters enable constant gene expression at any copy number in bacteria. *Nat Biotechnol* 36(4):352–358. <https://doi.org/10.1038/nbt.4111>
- Shopera T, He L, Oyetunde T, Tang YJ, Moon TS (2017) Decoupling resource-coupled gene expression in living cells. *ACS Synth Biol* 6(8):1596–1604
- Stricker J, Cookson S, Bennett MR, Mather WH, Tsimring LS, Hasty J (2008) A fast, robust and tunable synthetic gene oscillator. *Nature* 456:516–519
- Takahashi MK, Tan X, Dy AJ, Braff D, Akana RT, Furuta Y, Donghia N, Ananthakrishnan A, Collins JJ (2018) A low-cost paper-based synthetic biology platform for analyzing gut microbiota and host biomarkers. *Nature Commun* 9(1):3347. ISSN 2041-1723. <https://doi.org/10.1038/s41467-018-05864-4>
- Xie Z, Wroblewska L, Prochazka L, Weiss R, Benenson K (2011) Multi-input RNAi-based logic circuit for identification of specific cancer cells. *Science* 333:1307–11
- Yeung E, Dy AJ, Martin KB, Ng AH, Del Vecchio D, Beck JL, Collins JJ, Murray RM (2017) Biophysical constraints arising from compositional context in synthetic gene networks. *Cell Syst* 5:11–24
- Zhang F, Carothers JM, Keasling JD (2012) Design of a dynamic sensor-regulator system for production of chemicals and fuels derived from fatty acids. *Nat Biotechnol* 30:354–359

System Identification Software

Brett Ninness

School of Electrical and Computer Engineering,
University of Newcastle, Newcastle, Australia

Abstract

This contribution discusses various aspects important to software for system identification. Essential functionality for existing practice and the algorithmic fundamentals this relies on are considered together with a brief discussion of additional commonly useful support tools. Since software is intimately tied to the hardware that it runs on, a discussion on this topic follows with an emphasis on considering how future system identification software developments might best align with clear current and future trends in computer architecture developments.

Keywords

System identification · Computer-aided design · Parameter estimation · Software

Introduction

Fundamental to the practice of system identification is the employment of appropriate software to compute system estimates and evaluate their properties. One option is for the user to code the necessary routines themselves in their computer language of choice. For simple situations, such as least-squares estimation with a linearly parametrized model, this approach is feasible.

However, it quickly becomes onerous and time consuming as one moves even slightly beyond this simple example. In response to this, researchers have developed a number of software packages designed to accommodate classes of data formats, model structures, and estimation methods.

The purpose of this contribution is to profile the support that available system identification software provides, the underlying foundations on which this software depends, and the future capabilities that may be expected due to trends in desktop and portable computer capacity.

The material to follow depends on explanations, definitions, and background presented in ► [System Identification: An Overview](#), by Ljung, which should be read in conjunction with this contribution.

Essential Functionality

The essence of system identification software packages is that they implement an identification method $\mathcal{I}$ as defined in ► [System Identification: An Overview](#).

Typically, this involves taking a model structure specification $\mathcal{M}(\theta)$ together with N observed data points Z_N and translating that to a cost function $V_N(\theta)$ for which a minimizer

$$\hat{\theta} \triangleq \arg \min_{\theta \in D_{\mathcal{M}}} V_N(\theta) \quad (1)$$

is then computed in order to deliver a system estimate $\mathcal{M}(\hat{\theta})$.

While the details of these fundamental operations vary according to the chosen model structure and method, there are some shared aspects. To pick a starting point, subspace-based estimation methods (► [Subspace Techniques in System Identification](#)) have been one of the most significant developments in the near history of system identification, and they fundamentally involve a first stage of setting up and solving the optimization problem

$$\hat{\beta} = \arg \min_{\beta} \|Y - \Phi\beta\|_F^2, \quad (2)$$

where Y, Φ are data-dependent matrices, β is a θ -dependent matrix, and $\|\cdot\|_F$ is the Frobenius norm, which, for an $m \times n$ matrix A , is defined as

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}. \quad (3)$$

This is a classic least-squares optimization problem, which also arises in other system identification contexts, particularly when the prediction $\hat{y}(t | \theta)$ is a linear function of θ .

As is well known Golub and Loan (1989), the minimizer $\hat{\beta}$ satisfies the “normal equations”

$$(\Phi^T \Phi) \hat{\beta} = \Phi^T Y, \quad (4)$$

and if $\Phi^T \Phi$ is invertible, this allows for a closed-form solution

$$\hat{\beta} = (\Phi^T \Phi)^{-1} \Phi^T Y. \quad (5)$$

While formally correct, no system identification software packages would compute $\hat{\beta}$ in this manner since it is computationally inefficient and sensitive to numerical rounding errors.

Drawing on decades of study on this topic in the numerical computations literature (Golub and Loan 1989), system identification software packages rely on the QR factorization

$$\Phi = QR = [Q_1 | Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix}, \quad (6)$$

where Q is square and satisfies $Q^T Q = I$ (the identity matrix) and R contains the upper triangular square and invertible block R_1 . This decomposition of Φ allows the normal Eq. (4) to be re-expressed as

$$R_1 \hat{\beta} = Q_1^T Y. \quad (7)$$

Since R_1 is upper triangular, the solution $\hat{\beta}$ may then be found by elementary and numerically robust backward substitution (Golub and Loan 1989).

The importance of efficient and accurate solution of normal equations to any system identification software is not limited to these subspace or linearly parametrized cases. For instance, the very general class of prediction error methods

encompassed by the formulation (1) involves a cost $V_N(\theta)$ that depends on the vector

$$E(\theta) \triangleq [\varepsilon(t_1, \theta), \dots, \varepsilon(t_N, \theta)]^T \quad (8)$$

of differences between the observed data and the response of a model parametrized by θ . In the case of time-domain data, the elements of (8) are defined by

$$\varepsilon(t, \theta) \triangleq y(t) - \hat{y}(t | \theta). \quad (9)$$

In this general situation, it is most commonly the case that no closed-form solution for the optimization problem (1) exists.

The strategy then taken by most system identification software packages is to employ a gradient-based search for a minimizer. These methods are motivated by the use of a linear approximation of $E(\theta)$ about a current putative minimizer θ_k according to

$$E(\theta) \approx E(\theta_k) + J(\theta_k)(\theta - \theta_k), \quad (10)$$

where $J(\theta_k)$ denotes the Jacobian matrix

$$J(\theta_k) \triangleq \left. \frac{\partial E(\theta)}{\partial \theta} \right|_{\theta=\theta_k}. \quad (11)$$

In the very common situation where $V_N(\theta)$ is a quadratic function of $E(\theta)$, this implies the associated approximation

$$\begin{aligned} V_N(\theta) &= \text{Trace}\{E^T(\theta)E(\theta)\} \\ &= \|E\|_F^2 \approx \|E(\theta_k) + J(\theta_k)(\theta - \theta_k)\|_F^2. \end{aligned} \quad (12)$$

Via this reasoning, computation of an appropriate “search direction” $p = \theta - \theta_k$ again involves the efficient solution of a linear least-squares problem of the form (2), namely,

$$p = \arg \min_p \|E(\theta_k) + J(\theta_k) p\|_F^2. \quad (13)$$

More generally, system identification software packages extend this rationale and solve (1) by

generating a sequence of iterations $\{\theta_k\}$, which are refined according to

$$\theta_{k+1} = \theta_k + \mu p, \quad (14)$$

where μ is a step length that at each iteration k may be altered until a cost decrease

$$V_N(\theta_{k+1}) < V_N(\theta_k) \quad (15)$$

is achieved and the search direction p again involves the solution of normal equations

$$\left[J(\theta_k)^T J(\theta_k) + \lambda I \right] p = -J(\theta_k)^T E(\theta_k). \quad (16)$$

The choice $\lambda > 0$ implies what is called a Levenberg–Marquardt method, while $\lambda = 0$ leads to a so-called Gauss–Newton update strategy, and there are further variants such as “trust region” methods that are typically offered as options.

Via (16) we see that again system identification software comes to fundamentally depend on underpinning numerical linear algebra, in this case, again via the QR decomposition.

Another decomposition, the singular value decomposition (SVD), also has a significant role to play, particularly with respect to subspace-based methods where it is essential to the extraction of an estimated system parametrization $\hat{\theta}$ from $\hat{\beta}$ referred to in (2).

In addition to matrix decompositions, other system identification methods depend on many other even more fundamental linear algebra tools such as basic matrix/vector operations, matrix inversion, and eigen-decomposition. Because of this dependence, most (Ljung 2012; Kollár et al. 2006; Young and Taylor 2012; Garnier et al. 2012; Ninness et al. 2013) but not all (Hjalmarsson and Sjöberg 2012) currently available system identification software packages are built upon the MathWorks MATLAB (originally short for “matrix laboratory”) package, which provides an efficient interface to the widely accepted standard numerical linear algebra libraries LAPACK and EISPACK. For example, solving (2) efficiently and robustly via QR decomposition and back-substitution of (7) is achieved transparently using

the MATLAB backslash operator with the simple command: `beta = Phi\Y`.

Additional Functionality and the Decision-Making Process

As emphasized in ► [System Identification: An Overview](#), the provision of an estimated model is typically an iterative process (illustrated diagrammatically in Fig. 4 of ► [System Identification: An Overview](#)) of which just one component is the implementation of an identification method $\mathcal{I}$ to deliver a system estimate $\mathcal{M}(\hat{\theta})$.

In addition to this “essential functionality,” system identification software must also provide tools and a logistical support for the decision-making process of assessing $\mathcal{M}(\hat{\theta})$ and, based on this, perhaps altering aspects such as the choice of model structure $\mathcal{M}$, the experiment design $\mathcal{X}$, or indeed the identification method $\mathcal{I}$.

To support this, system identification software packages may offer further capabilities such as:

1. Nonparametric estimation methods that deliver estimates of linear system frequency response without involving a parametrized model structure $\mathcal{M}(\theta)$ and hence not involving (1)
2. Data preprocessing tools, such as to remove trends and to frequency selectively prefilter data before use
3. Visualization tools to display and compare the time- and frequency-domain response of estimated models
4. Model validation tools to determine if estimated models can be falsified by observed data
5. Model accuracy measures that deliver statistical confidence bounds on estimated parameters
6. Additional data processing tools such as Kalman filtering and smoothing routines and sequential Monte Carlo (particle filter) routines that are used to compute $V_N(\theta)$ but have many other applications
7. Graphical user interface (GUI) support in order to aid organization of the various

aspects of data preprocessing, model structure selection, algorithm selection, estimate computation, model validation, and model visualization

8. The employment of symbolic computation capabilities to aid complex model structure specification and preprocessing for efficient numerical implementation (Hjalmarsson and Sjöberg 2012)

Note that with the exception of this last point (8), the computations associated with this additional functionality again depend fundamentally on efficient numerical linear algebra software.

Computing Platforms

Currently available system identification software packages are designed for standard desktop computing environments, and as such their capabilities are intimately tied to those of the central processing unit (CPU), memory, and other architectural features of this hardware.

For instance, the linear algebra underpinnings just discussed are typically implemented in serially coded form, and hence bus bandwidth, together with memory and CPU speed, will be the fundamental factor affecting software

performance. Taking CPU speed as an example, the evolution of clock speed for the very commonly used Intel architecture CPUs is shown as the red curve in Fig. 1 and, as can be seen, has largely plateaued over the last decade after two orders of magnitude growth in the decade preceding it.

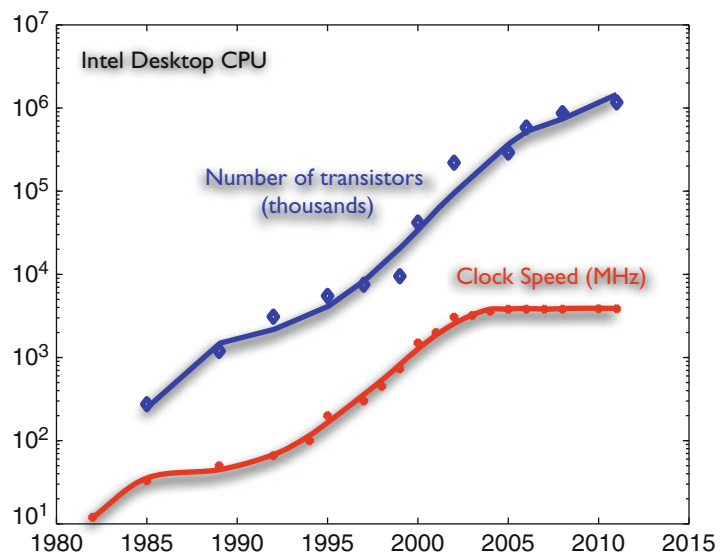
As a result, and roughly speaking, system estimates that took a minute to compute in the early 1990s took under a second to compute in the early part of this century, but are essentially no faster to compute now, a further decade later.

As a result, while system identification software has continued to grow in sophistication, in areas that involve high computational burdens, such as estimation of complex and high-dimensional model structures, or the implementation of compute intensive algorithms, the capability of system identification software has been hardware limited for some time.

At the same time, as the blue line in Fig. 1 illustrates, Moore's law continues to hold, and transistor densities continue to increase. While this is delivering no greater serial CPU speed, it is delivering multiple CPU core availability. Future advances in system identification software capability will therefore need to exploit the potential for parallel computation.

Indeed, in current MATLAB, the fundamental numerical linear algebra routines previously

System Identification Software, Fig. 1 Trends in desktop CPU capacity taking Intel as an example. Serial throughput speeds have long plateaued, but transistor density continues to grow, which delivers growing multiple cores



mentioned such as QR-based solution of normal equations, eigenvalue, and SVD decompositions will all automatically execute on multiple computational threads on multicore-enabled machines. Expanding this to take advantage of even higher levels of parallelism is the subject of current research.

While these developments will deliver performance enhancements for existing system identification methods, they will also open up the possibility for new tools to be added to system identification software suites.

For example, in addition to the existing subspace, prediction error, and maximum likelihood methods just mentioned, there is another important estimation approach that does not involve the solution of an optimization problem such as (1) or (2) and for which there is always a closed-form expression for the parameter estimate. It is the conditional mean estimate

$$\hat{\theta} = \mathbf{E} \{ \theta \mid Y \}, \quad (17)$$

which is a Bayesian approach that depends on the calculation of the posterior density of the parameters θ given the data Y according to

$$p(\theta \mid Y) = \frac{p(Y \mid \theta)p(\theta)}{p(Y)}, \quad (18)$$

where $p(\theta)$ is a prior that allows for incorporation of user knowledge (before observing the data) and $p(Y \mid \theta)$ is the usual data likelihood.

Not only does this estimate have an explicit formulation; it is also the minimum mean square error estimate in that for any other estimate $\hat{\beta} = f(Y)$ computed as any other measurable function f of the data Y , it holds that

$$\mathbf{E} \left\{ \|\theta - \hat{\theta}\|^2 \right\} \leq \mathbf{E} \left\{ \|\theta - \hat{\beta}\|^2 \right\}. \quad (19)$$

In this sense, the conditional mean (17) is the most accurate estimate. Furthermore, quantifications of estimation accuracy may be directly obtained via the marginal densities $p(\theta_i \mid Y)$ of individual parameter vector values θ_i .

Nevertheless, it is currently not widely used. There are no doubt philosophical reasons for this

stemming from the well-known debate between frequentist and Bayesian perspectives on inference (Efron 2013).

Another key reason is that it is difficult to compute. It requires the evaluation of a multidimensional integral,

$$\mathbf{E} \{ \theta \mid Y \} = \int \int \cdots \int \theta p(\theta \mid Y_N) d\theta_1 \cdots d\theta_n \quad (20)$$

as does the computation of the marginal densities

$$p(\theta_i \mid Y_N) = \int \cdots \int p(\theta \mid Y_N) d\theta_1 \cdots d\theta_{i-1} d\theta_{i+1} \cdots d\theta_n. \quad (21)$$

Evaluating these quantities requires adding fundamentally new capability beyond efficient linear algebra support to system identification software. It involves adding capability for numerical integration.

Integration in one dimension is straightforward. The well-known and used Simpson's rule is remarkably efficient in that the relationship between the computational error and the number of grid points m obeys

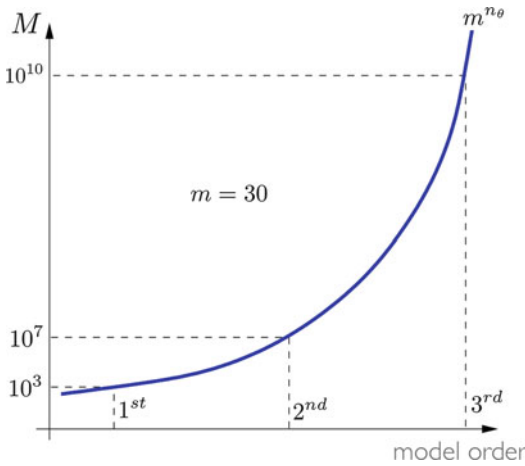
$$\text{Error} = O(m^{-4}) \quad (22)$$

so that every order of magnitude increase in m delivers four extra digits of precision. However, (20) is an $n_\theta = \dim\{\theta\}$ dimensional integral, and m grid points on each of n_θ axes imply

$$M = m^{n_\theta} \quad (23)$$

function evaluations. This can blow up quite quickly, as illustrated in Fig. 2 for the case of only modest $m = 30$ grid points and with respect to the very simple problem of estimating a straightforward linear output-error model of increasing order.

On a serial CPU platform, there is an upper limit of time available to wait for a result and hence an upper limit M of function evaluations that are tolerable. Viewed as a function of this, the accuracy of simple Simpson's rule methods is



System Identification Software, Fig. 2 Increase in number of function evaluations M required for Simpson's rule integration with $m = 30$ grid points on each parameter axis associated with linear output-error models of increasing order. Note that accounting for both numerator and denominator parameters, $n_\theta = 2 \times \text{model order} + 1$

$$\text{Error} = O(M^{-4/n_\theta}), \quad (24)$$

which is not attractive as model complexity and hence n_θ grows.

A further and vitally important problem is that it will generally not be clear where to allocate the m grid points on each axis since the support of the posterior $p(\theta | Y)$ is not readily known. Indeed, a main point of computing the multidimensional integrals associated with the marginals (21) is to determine this support.

A strategy to address these difficulties is based on the strong law of large numbers (SLLN). Namely, if random draws $x^i \sim p(x)$ from a density $p(x)$ can be obtained, then sample averages of functions of them converge with probability one to the ensemble average expectation, which is an integral:

$$\frac{1}{M} \sum_{i=1}^M f(x^i) \xrightarrow{\text{w.p.1}} \mathbf{E}\{f(x^i)\} = \int f(x)p(x) dx. \quad (25)$$

This principle may then be used as a “randomized” method to compute an estimate $\hat{I}_M$ of an integral I ; viz.,

$$I = \int f(x)p(x) dx \approx \hat{I}_M \triangleq \frac{1}{M} \sum_{i=1}^M f(x^i). \quad (26)$$

Furthermore, if the x^i are independent draws, then

$$\text{Var}\{\hat{I}_M\} = \frac{1}{M^2} \sum_{i=1}^M \text{Var}\{f(x^i)\} = \frac{1}{M} \text{Var}\{f(x)\}, \quad (27)$$

and hence the absolute error in integral evaluation is

$$O(|I - \hat{I}_M|) \approx O(M^{-1/2}). \quad (28)$$

The vital point is that as opposed to (24), this error is *independent* of the dimension of x and hence *independent* of the dimension of the integral I . Furthermore, the grid points are the realizations $\{x^i\}$, which naturally will lie within the support of the integrand $f(x)p(x)$ and do not need to be otherwise designed.

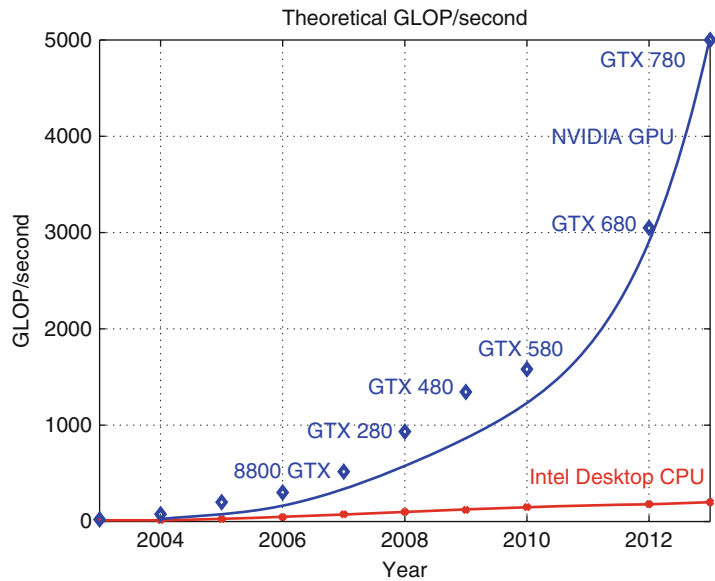
Of course, this depends on a means to draw samples from an arbitrary density $p(\cdot)$ of interest, but simple methods such as the Metropolis–Hastings methods and “slice sampler” exist to achieve this Mackay (2003).

Importantly too, these randomized methods are ideally suited to exploiting the growing availability of desktop multicore computing platforms. Generating M realizations to form the integral approximation $\hat{I}_M$ in (26) may be achieved in one-tenth the time simply by running ten independently initialized random number generators in parallel, each generating one $M/10$ length realization. The method (26) is thus (in principal) trivial to parallelize.

Furthermore, much greater parallelization and hence also speedup may be achieved by employing the “graphics processing units” (GPUs) in desktop computers. These GPUs are inexpensive because they service a high volume consumer demand for interactive gaming, which requires high-speed numerical computation for 3D-projected graphics. As such these GPUs have evolved to provide hundreds of parallel processing cores, each clocked in the gigahertz range.

System Identification Software,

Fig. 3 Historical trend of theoretical single-precision giga- FLOPS performance of commodity NVIDIA brand GPUs versus Intel architecture CPUs designed for desktop computing



To give an impression of the computational capability of GPU-based platforms, the single-precision giga-FLOPS (floating-point operations per second) performance history for NVIDIA brand GPUs and Intel architecture processors designed for desktop applications is profiled in Fig. 3.

This shows theoretical performance, assuming all cores may be fully utilized constantly. In reality, this is never possible due to communication and architecture restrictions. For example, GPU architectures are based on an SIMD (single instruction, multiple data) design, so at any one time many cores must execute the identical instruction, but may do so on different data. Analysis of these and other aspects relevant for system identification software implementation requires detailed study (Lee et al. 2010).

The fact that desktop hardware architectures have and will continue to offer more but not faster processing cores may be exploited in system identification software beyond this Bayesian setting. For example, the last decade has seen great interest in delivering estimation methods for an increasingly broad range of nonlinear model structures, a quite general version of which can be expressed in the nonlinear state-space form

$$x(t+1) \sim p(x(t+1) | x(t), \theta) \quad (29)$$

$$y(t) \sim p(y(t) | x(t), \theta). \quad (30)$$

In principle, there is no reason why this cannot be straightforwardly addressed by the usual maximum likelihood approach of forming the likelihood

$$p(Y_N | \theta) = \prod_{t=1}^N p(y(t) | Y_{t-1}, \theta),$$

$$Y_t = \{y(1), \dots, y(t)\} \quad (31)$$

and then using this as the cost function $V_N(\theta)$ in (1) and then proceeding with the usual gradient-based search. Indeed, there exist explicit formulae for computing the predictive densities $p(y(t) | Y_{t-1}, \theta)$ required in (31). Namely, the coupled measurement update

$$p(x(t) | Y_t, \theta) = \frac{p(y(t) | x(t), \theta) p(x(t) | Y_{t-1}, \theta)}{p(y(t) | Y_{t-1}, \theta)}$$

$$p(y(t) | Y_{t-1}, \theta) =$$

$$\int p(y(t) | x(t), \theta) p(x(t) | Y_{t-1}) dx(t)$$

and time update

$$p(x(t+1) | Y_t, \theta) =$$

$$\int p(x(t+1) | x(t), \theta) p(x(t) | Y(t), \theta) dx(t) \quad (32)$$

equations.

However, again we are faced with the problem of numerically evaluating multidimensional integrals. The integral dimension this time is that of the state vector $x(t)$, which may be less than that of the parameter vector θ just discussed, but $2N$ of these integrals needs to be evaluated in order to compute the likelihood (31), and this needs to be redone for each step of any associated gradient-based search.

Again, a randomized algorithm approach based on the SLLN could be considered as a way forward in system identification software development. Indeed, sequential Monte Carlo (SMC) algorithms (aka particle filtering) (Doucet and Johansen 2011) have been specifically developed to compute the above integrals involved in the time and measurement update, and there has been recent work (Schön et al. 2011; Andrieu et al. 2010) on employing this to develop software for the estimation of the general nonlinear model (29) and (30).

The resulting algorithms are computationally intensive, to the point where implementation on serial CPU architectures means they are limited to deployment on nonlinear model structures of very low state dimension. However, again because the SLLN is at the heart of the methods, and averaging over one long run on a serial machine is numerically equivalent (but potentially much faster) to averaging over multiple shorter runs computed in parallel, there is scope for future system identification software to employ these approaches.

Examples of Available System Identification Software

With the features of current and perhaps future system identification software packages profiled, it may be useful to make specific mention of

particular system identification software packages that have been under active development for a substantial period of time. These include the following commercially available packages:

1. The *MathWorks System Identification Toolbox* (Ljung 2012), which is arguably the most mature and comprehensive system identification software available
2. The *GAMAX Frequency Domain System Identification Toolbox* (Kollár et al. 2006), which specializes in estimation of models based on measurements in the frequency domain
3. The *Adaptx* software (Larimore 2000) specializing in the estimation of state-space models using subspace-based methods

Noncommercial and freely available system identification software packages that are relevant include:

1. The “computer-aided program for time-series analysis and identification of noisy systems” (CAPTAIN) toolbox (Young and Taylor 2012), which provides a platform supporting the “refined instrumental variable” (RIV) algorithm for linear system estimation;
2. The “continuous-time system identification” (CONTSID) toolbox (Garnier et al. 2012), which specializes in the estimation of continuous-time models
3. The “interactive software tool for system identification education” (ITSIE) toolbox (Guzmán et al. 2012), which has an emphasis on education and training in system identification principles
4. The “University of Newcastle identification toolbox” (UNIT) software (Ninness et al. 2013) that is designed as an open platform for researchers to evaluate the performance of new methods relative to established ones

Summary and Future Directions

A case can be mounted that at its heart, system identification is about the design of software and the understanding of the results provided by it.

Certainly, the field has been built on decades of deep theoretical contributions, but this has been very practically focused either on delivering new algorithms that may be directly implemented or on better understanding the performance of existing algorithms.

Efficient numerical linear algebra routines have traditionally been the foundation of the resulting proven and effective system identification methods and software to date, and these have scaled in effectiveness as desktop computing clock speeds have scaled.

However, the recent past and the foreseeable future see CPU speed as static and with an increasing number of available processor cores. Delivering greater system identification capacity will require the development of methods whose software implementations can harness this growing availability of multiple processor cores.

Cross-References

- [Frequency Domain System Identification](#)
- [Nonlinear System Identification Using Particle Filters](#)
- [System Identification: An Overview](#)
- [System Identification Techniques: Convexification, Regularization, Relaxation](#)

Recommended Reading

For readers wishing to gain a deeper understanding of the numerical linear algebra aspects discussed here, the classic text (Golub and Loan 1989) is recommended. Those wishing further background on the calculation of multidimensional integrals via randomized algorithms such as Metropolis–Hastings and slice sampling will find (Mackay 2003) useful. The particle filtering methods mentioned here for nonlinear estimation problems are clearly explained in Doucet and Johansen (2011). Readers interested in further detail on numerical computations on GPU-

based platforms supporting these computations will find (Lee et al. 2010) useful.

Bibliography

- Andrieu C, Doucet A, Holenstein R (2010) Particle Markov chain Monte Carlo methods. *J R Stat Soc Ser B* 72:1–33
- Doucet A, Johansen AM (2011) A tutorial on particle filtering and smoothing: fifteen years later. In: Crisan D, Rozovsky B (eds) *Nonlinear filtering handbook*. Oxford University Press, London
- Efron B (2013) A 250 year argument: belief, behaviour and the bootstrap. *Bull Am Math Soc* 50:129–146
- Garnier H, Gilson M, Laurain V (2012) Developments for the CONTSID toolbox. In: *Proceedings of the 16th IFAC symposium on system identification, Brussels*. ISBN:978–3–902823–06–9
- Golub G, Loan CV (1989) *Matrix computations*. Johns Hopkins University Press, Baltimore
- Guzmán J, Rivera D, Dormido S, Berenguel M (2012) An interactive software tool for system identification. *Adv Eng Softw* 45:115–123
- Hjalmarsson H, Sjöberg J (2012) A mathematica toolbox for signals, systems and identification. In: *Proceedings of the 16th IFAC symposium on system identification, Brussels*
- Kollár I, Pintelon R, Schoukens J (2006) Frequency domain system identification toolbox for Matlab: characterizing nonlinear errors of linear models. In: *Proceedings of the 14th IFAC symposium on system identification, Newcastle*, pp 726–731
- Larimore WE (2000) The adaptx software for automated multivariable system identification. In: *Proceedings of the 12th IFAC symposium on system identification, Santa Barbara*
- Lee A, Yau C, Giles M, Doucet A, Holmes C (2010) On the utility of graphics cards to perform massively parallel simulation of advanced monte-carlo methods. *J Comput Graph Stat* 19: 769–789
- Ljung L (1999) *System identification: theory for the user*, 2nd edn. Prentice-Hall, New Jersey
- Ljung L (2012) *MATLAB system identification toolbox users guide, version 8*. The Mathworks
- Mackay DJ (2003) *Information theory, inference, and learning algorithms*. Cambridge University Press, Cambridge/New York
- Ninness B, Wills A, Mills A (2013) Unit: a freely available system identification toolbox. *Control Eng Pract* 21:631–644
- Schön T, Wills A, Ninness B (2011) System identification of nonlinear state-space models. *Automatica* 37:39–49
- Young PC, Taylor CJ (2012) Recent developments in the CAPTAIN toolbox for Matlab. In: *Proceedings of the 16th IFAC symposium on system identification, Brussels*. ISBN:978–3–902823–06–9

System Identification Techniques: Convexification, Regularization, Relaxation

Alessandro Chiuso

Department of Information Engineering,
University of Padova, Padova, Italy

Abstract

System identification has been developed, by and large, following the classical parametric approach. In this entry we discuss how regularization theory can be employed to tackle the system identification problem from a nonparametric (or semi-parametric) point of view. Both regularization for smoothness and regularization for sparseness are discussed, as flexible means to face the bias/variance dilemma and to perform model selection. These techniques have also advantages from the computational point of view, leading sometimes to convex optimization problems.

Keywords

Kernel methods · Nonparametric methods · Optimization · Sparse bayesian learning · Sparsity

Introduction

System identification is concerned with automatic model building from measured data. Under this unifying umbrella, this field spans a rather broad spectrum of topics, considering different model classes (linear, hybrid, nonlinear, continuous, and discrete time) as well as a variety of methodologies and algorithms, bringing together in a nontrivial way concepts from classical statistics, machine learning, and dynamical systems.

Even though considerable effort has been devoted to specific areas, such as parametric methods for linear system identification which are by now well developed (see the introductory article ► [“System Identification: An Overview”](#)),

it is fair to say that modeling still is, by far, the most time-consuming and costly step in advanced process control applications. As such, the demand for fast and reliable automated procedures for system identification makes this exciting field still a very active and lively one.

Suffices here to recall that, following this classic parametric maximum likelihood (ML)/prediction error (PE) framework, the candidate models are described using a finite number of parameters $\theta \in \mathbb{R}^n$. After the model classes have been specified, the following two steps have to be undertaken:

- (i) Estimate the model complexity $\hat{n}$.
- (ii) Find the estimator $\hat{\theta} \in \mathbb{R}^{\hat{n}}$ minimizing a cost function $J(\theta)$, e.g., the prediction error or (minus) the log-likelihood.

Both of these steps are critical, yet for different reasons: step (ii) boils down to an optimization problem which, in general, is non-convex, and as such it is very hard to guarantee that a global minimum is achieved. The regularization techniques discussed in this entry sometimes allow to reformulate the identification problem as a convex program, thus solving the issue of local minima.

In addition fixing the system complexity equal to the “true” one is a rather unrealistic assumption, and in practice the complexity n has to be estimated as per step (i). In practice there is never a “true” model, certainly not in the model class considered. The problem of statistical modeling is first of all an approximation problem; one seeks for an approximate description of “reality” which is at the same time simple enough to be learned with the available data and also accurate enough for the purpose at hand. On this issue see also the section “Trade-off Between Bias and Variance” in ► [“System Identification: An Overview”](#). This has nontrivial implications, chiefly the facts that classical order selection criteria are based on asymptotic arguments and that the statistical properties of estimators $\hat{\theta}$ after model selection, called post-model-selection estimators (PMSEs), are in general difficult to study (Leeb and Pötscher 2005) and may lead to unde-

sirable behavior. Experimental evidence shows that this is not only a theoretical problem but also a practical one (Pillonetto et al. 2011; Chen et al. 2012). On top of this statistical aspect, there is also a computational one. In fact the model selection step, which includes as special cases also variable selection and structure selection, may lead to computationally intractable combinatorial problems. Two simple examples which reveal the combinatorial explosion of candidate models are the following:

- (a) *Variable selection*: consider a high-dimensional time series (MIMO) where not all inputs/outputs are relevant and one would like to select k out of m available input signals where k is not known and needs to be inferred from data (see, e.g., Banbura et al. 2010 and Chiuso and Pillonetto 2012).
- (b) *Structure selection*: consider all autoregressive models of maximal lag p with only $p_0 < p$ nonzero coefficients and one would like to estimate how many (p_0) and which coefficients are nonzero.

The same combinatorial problem arises in hybrid system identification (e.g., switching ARX models). Given that enumeration of all possible models is essentially impossible due the combinatorial explosion of candidates, selection could be performed using greedy approaches from multivariate statistics, such as stepwise methods (Hocking 1976).

The system identification community, inspired by work in statistics (Tibshirani 1996; Mackay 1994), machine learning (Rasmussen and Williams 2006; Tipping 2001; Bach et al. 2004), and signal processing (Donoho 2006; Wipf et al. 2011), has recently developed and adapted methods based on regularization to jointly perform model selection and estimation in a computationally efficient and statistically robust manner. Different regularization strategies have been employed which can be classified in two main classes: regularization induced by so-called smoothness priors (aka Tikhonov regularization; see Kitagawa and Gersh (1984) and Doan et al. (1984) for early references in the

field of dynamical systems) and regularization for selection. This latter is usually achieved by convex relaxation of the ℓ_0 quasinorm (such as ℓ_1 norm and variations thereof such as sum of norms, nuclear norm, etc.) or other non-convex sparsity-inducing penalties which can be conveniently derived in a Bayesian framework, aka sparse Bayesian learning (SBL) (Mackay 1994; Tipping 2001; Wipf et al. 2011).

The purpose of this entry is to guide the reader through the most interesting and promising results on this topic as well as areas of active research; of course this subjective view only reflects the author's opinion, and of course different authors could have offered a different perspective.

While, as mentioned above, system identification studies various classes of models (ranging from linear to general “nonlinear” models), in this entry, we shall restrict our attention to specific ones, namely, linear and hybrid dynamical systems. The field of nonlinear system identification is so vast (a quote sometimes attributed to S. Ulam has it that the study of nonlinear systems is a sort of “non-elephant zoology”) that even though it has largely benefitted from the use of regularization, it cannot be addressed within the limited space of this contribution. The reader is referred to the Encyclopedia chapters ▶ “Nonlinear System Identification: An Overview of Common Approaches” and ▶ “Nonlinear System Identification Using Particle Filters” for more details on nonlinear model identification.

System Identification

Let $u_t \in \mathbb{R}^m$ and $y_t \in \mathbb{R}^p$ be, respectively, the measured *input* and *output* signals in a dynamical system; the purpose of system identification is to find, from a finite collection of input-output data $\{u_t, y_t\}_{t \in [1, N]}$, a “good” dynamical model which describes the phenomenon under observation. The candidate model will be searched for within a so-called model set denoted by $\mathcal{M}$. This set can be described in parametric form

(see, e.g., Eq. (3) in ► “[System Identification: An Overview](#)”) or in a nonparametric form. In this entry we shall use the symbol $\mathcal{M}_n(\theta)$ for parametric model classes where the subscript n denotes the model complexity, i.e., the number of free parameters.

Linear Models

The first part of the entry will address identification of linear models, i.e., models described by a convolution

$$y_t = \sum_{k=1}^{\infty} g_{t-k} u_k + \sum_{k=0}^{\infty} h_{t-k} e_k \quad t \in \mathbb{Z} \quad (1)$$

where g and h are the so-called impulse responses of the system and $\{e_t\}_{t \in \mathbb{Z}}$ is a zero-mean white noise process which under suitable assumptions is the one-step-ahead prediction error; a convenient description of the linear system (1) is given in terms of the transfer functions

$$G(q) := \sum_{k=1}^{\infty} g_k q^{-k} \quad H(q) := \sum_{k=0}^{\infty} h_k q^{-k}$$

The linear model (1) naturally yields an “optimal” (in the mean square sense) output predictor which shall be denoted later on by $\hat{y}_{t|t-1}$. As mentioned above, under suitable assumptions, the noise e_t in (1) is the so-called *innovation* process $e_t = y_t - \hat{y}_{t|t-1}$. See also Eq. (8) in ► “[System Identification: An Overview](#)”.

When g and h are described in a parametric form, we shall use the notation $g_k(\theta)$, $h_k(\theta)$, and, likewise, $G(q, \theta)$, $H(q, \theta)$, and $\hat{y}_{t|t-1}(\theta)$.

Example 1 Consider the so-called output-error model, i.e., assume $H(q) = 1$. An example of *parametric* model class is obtained restricting $G(q, \theta)$ to be a rational function

$$G(q, \theta) = K \prod_{i=1}^n \frac{q - z_i}{q - p_i}$$

where $\theta := [K, p_1, z_1, \dots, p_n, z_n]$ is the parameter vector. Note that the parameter vector θ may be subjected to constraints $\theta \in \Theta$, e.g., enforcing

that the system be bounded input and bounded output (BIBO) stable ($|p_i| < 1$) or that the impulse response be real ($K \in \mathbb{R}$ and poles p_i and zeros z_i appear in complex conjugate pairs).

An example of *nonparametric* model is obtained, e.g., postulating that g_k is a realization of a Gaussian process (Rasmussen and Williams 2006) with zero mean and a certain covariance function $R(t, s) = \text{cov}(g_t, g_s)$. For instance, the choice $R(t, s) = \lambda^t \delta_{t-s}$, where $|\lambda| < 1$ and δ_k is the Kronecker symbol, postulates that the g_t and g_s are uncorrelated for $t \neq s$ and that the variance of g_t decays exponentially in t ; this latter condition ensures that each realization g_k , $k > 0$, is BIBO stable with probability one. The exponential decay of g_t guarantees that, to any practical purpose, it can be considered zero for $t > T$ for a suitably large T . This allows to approximate the OE model with a “long” *finite impulse response* (FIR) model

$$G(q) = \sum_{k=1}^T g_k z^{-k} \quad (2)$$

where g_k , $k = 1, \dots, T$, is modeled as a zero-mean Gaussian vector with covariance Σ , with elements $[\Sigma]_{ts} = R(t, s)$.

Remark 1 Note that the model (2), which has been obtained from truncation of a nonparametric model, could in principle be thought as a parametric model in which the parameter vector θ contains all the entries of g_k , $k = 1, \dots, T$. Yet the truncation index T may have to be large even for relatively “simple” impulse responses; for instance, $\{g_k(\theta)\}_{k \in \mathbb{Z}^+}$ may be a simple decaying exponential, $g_k(\theta) = \alpha \rho^k$, which is described by two parameters (amplitude and decay rate), yet if $|\rho| \simeq 1$, the truncation index T needs to be large (ideally $T \rightarrow \infty$) to obtain sensible results (e.g., with low bias). Therefore, the number of parameters $T(m \times p)$ may be larger (and in fact much larger) than the available number of data points N . Under these conditions, the parameter θ cannot be estimated from any finite data segment unless further constraints are imposed.

The Role of Regularization in Linear System Identification

In order to simplify the presentation, we shall refer to the linear model (1) and assume that $H(q) = 1$, i.e., we consider the so-called linear output-error (OE) models. The extension to more general model classes can be found in Pillonetto et al. (2011), Chen et al. (2012), Chiuso and Pillonetto (2012), and references therein.

The main purpose of regularization is to control the model complexity in a flexible manner, moving from families of rigid, finite dimensional parametric model classes $\mathcal{M}_n(\theta)$ to flexible, possibly infinite dimensional, models. To this purpose one starts with a “suitably large” model class which is constrained through the use of so-called regularization functionals. To simplify the presentation, we consider the FIR (2). The estimator $\hat{\theta}$ is found as the solution of the following optimization problem

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^n} J_F(\theta) + J_R(\theta; \lambda) \quad (3)$$

where $J_F(\theta)$ is the “fit” term often measured in terms of average squared prediction errors:

$$J_F(\theta) := \frac{1}{N} \sum_{t=1}^N \|y_t - \hat{y}_{t|t-1}(\theta)\|^2 \quad (4)$$

while $J_R(\theta; \lambda)$ is a regularization term which penalizes certain parameter vectors θ associated to “unlikely” systems. Equation (3) can be seen as a way to deal with the *bias-variance trade-off*. The regularization term $J_R(\theta; \lambda)$ may depend upon some regularization parameters λ which need to be tuned using measured data. In its simplest instance,

$$J_R(\theta; \lambda) = \lambda J_R(\theta)$$

where λ is a scale factor that controls “how much” regularization is needed. We now discuss different forms of regularization $J_R(\theta; \lambda)$ which have been studied in the literature.

Example 2 Let us consider the FIR model in Eq. (2), and let θ be a vector containing all the unknown coefficients of the impulse response

$\{g_k\}_{k=1,\dots,T}$. The linear least squares estimator

$$\hat{\theta}_{LS} := \arg \min_{\theta} \frac{1}{N} \sum_{t=1}^N \|y_t - \hat{y}_{t|t-1}(\theta)\|^2 \quad (5)$$

is ill-posed unless the number of data N is larger (and in fact much larger) than the number of parameters T . From the statistical point of view, the estimator (5) would result for large T in small bias and large variance. The purpose of regularization is to render the inverse problem of finding θ from the data $\{y_t\}_{t=1,\dots,N}$ well posed, thus better trading bias versus variance. The simplest form of regularization is indeed the so-called ridge regression or its weighted version (aka generalized Tikhonov regularization), where the 2-norm of the vector θ is weighted w.r.t. a positive semidefinite matrix Q ,

$$\begin{aligned} \hat{\theta}_{\text{Reg}} := \arg \min_{\theta} \frac{1}{N} \sum_{t=1}^N \|y_t - \hat{y}_{t|t-1}(\theta)\|^2 \\ + \lambda \theta^\top Q \theta \end{aligned} \quad (6)$$

which result in so-called regularization for smoothness; see section “[Regularization for Smoothness](#).” The choice of the weighting Q is highly nontrivial in the system identification context, and the performance of the regularized estimator $\hat{\theta}_{\text{Reg}}$ heavily depends on this.

Remark 2 In order to formalize these ideas for nonparametric models or, equivalently, when the parameter θ is infinite dimensional, one has to bring in functional analytic tools, such as reproducing kernel Hilbert spaces (RKHS). This is rather standard in the literature on ill-posed inverse problems and has been recently introduced also in the system identification setting (Pillonetto et al. 2011). We shall not discuss these issues here because, we believe, the formalism would render the content less accessible.

Note that this regularization approach admits a completely equivalent Bayesian formulation simply setting

$$p(y|\theta) \propto e^{-J_F(\theta)} \quad p(\theta|\lambda) \propto e^{-J_R(\theta;\lambda)} \quad (7)$$

The densities $p(y|\theta)$ and $p(\theta|\lambda)$ are, respectively, the likelihood function and the prior, which in turn may depend on the unknown regularization parameters λ , aka hyperparameters in this Bayesian formulation. This is straightforward in the finite dimensional setting, while it requires some care when θ is infinite dimensional. With reference to Example 1 and assuming θ contains the impulse response coefficients g_k in (2), $p(\theta|\lambda)$ is a Gaussian density with zero mean and covariance Σ which may depend upon some regularization parameters λ . From the definitions (7), it follows that

$$p(\theta|y, \lambda) \propto p(y|\theta)p(\theta|\lambda) \quad (8)$$

from which point estimators of θ can be obtained (e.g., as posterior mean, MAP, etc.). As such, with some abuse of terminology, we shall indifferently refer to $J_R(\theta; \lambda)$ as the “regularization term” or the “prior.” The unknown parameter λ is used to introduce some flexibility in the regularization term $J_R(\theta; \lambda)$ or equivalently in the prior $p(\theta|\lambda)$ and is tuned based on measured data as discussed later on.

The regularization term $J_R(\theta; \lambda)$ can be roughly classified in *regularization for smoothness*, which attempts to control complexity in a smooth fashion and *regularization for sparseness* which, on top of estimation, also aims at selecting among a finite (yet possibly very large) number of candidate model classes.

Regularization for Smoothness

Let us consider a single-input, single-output FIR model of length T (arbitrarily large), and let $\theta := [g_1 \ g_2 \ \dots \ g_T]^\top \in \mathbb{R}^T$ be the (finite) impulse response; define also $y \in \mathbb{R}^N$ be the vector of output observations, Φ the regressor matrix with past input samples, and e the vector with innovations (zero mean, variance $\sigma^2 I$). With this notation the convolution input-output equation (1) takes the form

$$y = \Phi\theta + e$$

Following the prescriptions of ridge regression, a regularized estimator $\hat{\theta}$ can be found setting

$$J_R(\theta; \lambda) = \theta^\top K^{-1}(\lambda)\theta \quad (9)$$

where the matrix $K(\lambda)$, aka kernel, is tailored to capture specific properties of impulse responses (exponential decay, BIBO stability, smoothness, etc.). Early references include Doan et al. (1984) and Kitagawa and Gersh (1984), while more recent work can be found in Pillonetto and De Nicolao (2010), Pillonetto et al. (2011), and Chen et al. (2012) where several choices of kernels are discussed; the recent paper Zorzi and Chiuso (2018) provides a frequency domain interpretation of several kernel functions using tools from harmonic analysis.

Example 3 The simplest example of kernel is the so-called “exponentially decaying” kernel

$$K(\lambda) := \gamma D(\rho) \quad D(\rho) := \text{diag}\{\rho, \dots, \rho^T\} \quad (10)$$

where $\lambda := (\gamma, \rho)$ with $0 < \rho < 1$ and $\gamma \geq 0$.

For fixed λ , the estimator $\hat{\theta}(\lambda)$ is the solution of a quadratic problem and can be written in closed form (aka ridge regression):

$$\hat{\theta}(\lambda) = K(\lambda)\Phi^\top \left(\Phi K(\lambda)\Phi^\top + \sigma^2 I \right)^{-1} y \quad (11)$$

Two common strategies adopted to estimate the parameters λ are cross validation (Ljung 1999) and marginal likelihood maximization. This latter approach is based on the Bayesian interpretation given in Eq. (7) from which one can compute the so-called “empirical Bayes” estimator $\hat{\theta}_{\text{EB}} := \hat{\theta}(\hat{\lambda}_{\text{ML}})$ of θ plugging in (11) the estimator of λ which maximizes the marginal likelihood:

$$\begin{aligned} \hat{\lambda}_{\text{ML}} &:= \arg \max_{\lambda} p(\lambda|y) \\ &= \arg \max_{\lambda} \int p(\lambda, \theta|y) d\theta \end{aligned} \quad (12)$$

The main strength of the marginal likelihood is that by integrating the joint posterior over the unknown hyperparameters θ , it automatically accounts for the residual uncertainty in θ for

fixed λ . When both J_F and J_R are quadratic costs, which corresponds to assuming that e and θ are independent and Gaussian, the marginal likelihood in (12) can be computed in closed form so that

$$\hat{\lambda}_{\text{ML}} := \arg \min_{\lambda} \log(\det(\Sigma(\lambda))) + y^\top \Sigma^{-1}(\lambda) y$$

$$\Sigma(\lambda) := \Phi K(\lambda) \Phi^\top + \sigma^2 I \quad (13)$$

It is here interesting to observe that $\hat{\lambda}_{\text{ML}}$ which solves (12) under certain conditions leads to $K(\hat{\lambda}_{\text{ML}}) = 0$ (see Example 4), so that the estimator of θ in (11) satisfies $\hat{\theta}(\hat{\lambda}_{\text{ML}}) = 0$. This simple observation is the basis of so-called sparse Bayesian learning (SBL); we shall return to this issue in the next section when discussing regularization for sparsity and selection.

Unfortunately the optimization problem (12) (or (13)) is not convex and thus subjected to the issue of local minima. However, both experimental evidence and some theoretical results support the use of marginal likelihood maximization for estimating regularization parameters; see, e.g., Rasmussen and Williams (2006) and Aravkin et al. (2014).

Regularization for Sparsity: Variable Selection and Order Estimation

The main purpose of regularization for sparseness is to provide estimators $\hat{\theta}$ in which subsets or functions of the estimated parameters are equal to zero.

Consider the multi-input, multi-output OE model

$$y_{t,j} = \sum_{i=1}^m \sum_{k=1}^T g_{k,ij} u_{t-k,i} + e_{t,i} \quad j = 1, \dots, p \quad (14)$$

where $y_{t,j}$ denotes the j th component of $y_t \in \mathbb{R}^p$; let also $\theta \in \mathbb{R}^{T(m+p)}$ be the vector containing all the impulse response coefficients $g_{k,ij}$, $j = 1, \dots, p$, $i = 1, \dots, m$, and $k = 1, \dots, T$. With reference to Eq. (14), simple examples of sparsity one may be interested in are:

- (i) Single elements of the parameter vector θ , which corresponds to eliminating specific lags of some variables from the model (14).
- (ii) Groups of parameters such as the impulse response from i th input to the j th output $g_{k,ij}$, $k = 1, \dots, T$, thereby eliminating the i th input from the model for the j th output.
- (iii) The singular values of the Hankel matrix $\mathcal{H}(\theta)$ formed with the impulse response coefficients g_k ; in fact the rank of the Hankel matrix equals the order (i.e., the McMillan degree) of the system. (Strictly speaking any full rank FIR model of length T has McMillan degree $T \times p$. Yet, we consider $\{g_k\}_{k=1,\dots,T}$ to be the truncation of some “true” impulse response $\{g_k\}_{k=1,\dots,\infty}$, and, as such, the finite Hankel matrix built with the coefficients g_k will have rank equal to the McMillan degree of $G(q) = \sum_{k=1}^{\infty} g_k z^{-k}$.)

To this purpose one would like to penalize the number of nonzero terms; let them be entries of θ , groups, singular values, etc. This is measured by the ℓ_0 quasinorm or its variations: group ℓ_0 and ℓ_0 quasinorm of the Hankel singular values, i.e., the rank of the Hankel matrix. Unfortunately if J_R is a function of the ℓ_0 quasinorm, the resulting optimization problem is computationally intractable; as such one usually resorts to relaxations. Three common ones are described below.

One possibility is to resort to greedy algorithms such as orthogonal matching pursuit; generically it is not possible to guarantee convergence to a global minimum point.

A very popular alternative is to replace the ℓ_0 quasinorm by its *convex envelope*, i.e., the ℓ_1 norm, leading to algorithms known in statistics as LASSO (Tibshirani 1996) or its group version group LASSO (Yuan and Lin 2006):

$$J_R(\theta; \lambda) = \lambda \|\theta\|_1 \quad (15)$$

Similarly the convex relaxation of the rank (i.e., the ℓ_0 quasinorm of the singular values) is the so-called nuclear norm (aka Ky Fan n -norm or trace norm), which is the sum of the singular values $\|A\|_* := \text{trace}\{\sqrt{A^\top A}\}$ where $\sqrt{\cdot}$ denotes the matrix square root which is well defined for

positive semidefinite matrices. In order to control the order (McMillan degree) of a linear system, which is equal to the rank of the Hankel matrix $\mathcal{H}(\theta)$ built with the impulse response described by the parameter θ , it is then possible to use the regularization term

$$J_R(\theta; \lambda) = \lambda \|\mathcal{H}(\theta)\|_* \quad (16)$$

thus leading to convex optimization problems (Fazel et al. 2001). The implications of using nuclear and atomic norms in system identification have been studied in Pillonetto et al. (2016), while Prando et al. (2017a) introduces an iterative reweighted scheme to account for smoothness, stability, and complexity (MacMillan degree) at the same time.

Both (16) and (15) induce sparse or nearly sparse solutions (in terms of elements or groups of θ (15) or in terms of Hankel singular values (16)), making them attractive for selection. It is interesting to observe that both ℓ_1 and group ℓ_1 are special cases of the nuclear norm if one considers matrices with fixed eigenspaces. Yet, as well documented in the statistics literature, both (16) and (15) do not provide a satisfactory trade-off between sparsity and shrinking, which is controlled by the regularization parameter λ . As λ varies one obtains the so-called regularization path. Increasing λ the solution gets sparser, but, unfortunately, it suffers from shrinking of nonzero parameters. To overcome these problems, several variations of LASSO have been developed and studied, such as adaptive LASSO (Zou 2006), SCAD (Fan and Li 2001), and so on. We shall now discuss a Bayesian alternative which, to some extent, provides a better trade-off between sparsity and shrinking than the ℓ_1 norm.

This Bayesian procedure goes under the name of sparse Bayesian learning and can be seen as an extension of the Bayesian procedure for regularization described in the previous section. In order to illustrate the method, we consider its simplest instance. Consider an MIMO system as in (14) with $p = 1$ and $m = 2$, i.e.,

$$\begin{aligned} y_t &= \sum_{k=1}^T g_{k,1} u_{t-k,1} + \sum_{k=1}^T g_{k,2} u_{t-k,2} + e_t \\ &= \phi_{t,1}^\top g_1 + \phi_{t,2}^\top g_2 + e_t \end{aligned} \quad (17)$$

where $g_i := [g_{1,i}, \dots, g_{t,i}]$. Let $\theta := [g_1^\top \ g_2^\top]^\top$, and assume that the g_i s are independent Gaussian random vectors with zero mean and covariances $\lambda_i K$. Letting $\Phi_i := [\phi_{1,i}, \dots, \phi_{N,i}]^\top$ and following the formulation in (7) and (8), it follows that the marginal likelihood estimator of λ takes the form

$$\hat{\lambda}_{\text{ML}} := \arg \min_{\lambda_i \geq 0} \log(\det(\Sigma(\lambda))) + y^\top \Sigma^{-1}(\lambda) y$$

$$\Sigma(\lambda) := \lambda_1 \Phi_1 K \Phi_1^\top + \lambda_2 \Phi_2 K \Phi_2^\top + \sigma^2 I \quad (18)$$

After $\hat{\lambda}_{\text{ML}}$ has been found, the estimator of θ is found in closed form as per Eq. (11). It can be shown that under certain conditions on the observation vector y , the estimated hyperparameters $\hat{\lambda}_{\text{ML},i}$ lie at the boundary, i.e., are exactly equal to zero. If $\hat{\lambda}_{\text{ML},i} = 0$, then, from Eq. (11), also $\hat{g}_i = 0$; this reveals that in (17), the i th input does not enter into the model; see also Example 4 for a simple illustration.

These Bayesian methods for sparsity have been studied in a general regression framework in Wipf et al. (2011) under the name of “type-II” maximum likelihood. Further results can be found in Aravkin et al. (2014) which suggest that these Bayesian methods provide a better trade-off between sparsity and shrinking (i.e., are able to provide sparse solution without inducing excessive shrinkage on the nonzero parameters).

Remark 3 A more detailed analysis, see, for instance, Aravkin et al. (2014), shows that LASSO/GLASSO (i.e., ℓ_1 penalties) and SBL using the “empirical Bayes” approach can be derived under a common Bayesian framework starting from the joint posterior $p(\lambda, \theta|y)$. While SBL is derived from the maximization λ of the marginal posterior, LASSO/GLASSO corresponds to maximizing the joint posterior after a suitable change of variables. For reasons of space, we refer the interested reader to the literature for details.

Recent work on the use of sparseness for variable selection and model order estimation can be found in Wang et al. (2007), Chiuso and Pillonetto (2012), and references therein.

Example 4 In order to illustrate how sparse Bayesian learning leads to sparse solutions, we consider a very simplified scenario in which the measurements equation is

$$y_t = \theta u_{t-1} + e_t$$

where e_t is zero-mean, unit variance Gaussian, and white and u_t is a deterministic signal. The purpose is to estimate the coefficient θ , which could be possibly equal to zero. Thus, the estimator should reveal whether u_{t-1} influences y_t or not.

Following the SBL framework, we model θ as a Gaussian random variable, with zero mean and variance λ , independent of e_t . Therefore, y_t is also Gaussian, zero mean, and variance $u_{t-1}^2 \lambda + 1$. Therefore, assuming N data points are available, the likelihood function for λ is given by

$$L(\lambda) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi(u_{i-1}^2 \lambda + 1)}} e^{-\frac{1}{2} \sum_{i=1}^N \frac{y_i^2}{u_{i-1}^2 \lambda + 1}}$$

Defining now

$$\hat{\lambda}_{ML} := \arg \min_{\lambda \geq 0} -2 \log L(\lambda)$$

one obtains that

$$\hat{\lambda}_{ML} = \max(0, \lambda_*)$$

where λ_* is the solution of

$$\sum_{i=1}^N \frac{u_{i-1}^4 \lambda + u_{i-1}^2 (1 - y_i^2)}{(u_{i-1}^2 \lambda + 1)^2} = 0$$

which unfortunately is not available in closed form. If however we assume that the input u_t is constant (without loss of generality say that $u_t = 1$), we obtain that

$$\lambda_* = \frac{1}{N} \sum_{i=1}^N y_i^2 - 1$$

thus

$$\hat{\lambda}_{ML} = \max\left(0, \frac{1}{N} \sum_{i=1}^N y_i^2 - 1\right)$$

Clearly this is a threshold estimator which sets to zero $\hat{\lambda}_{ML}$ when the sample variance of y_t is smaller than the variance of e_t , which was assumed to be equal to 1. Defining $\Phi = [u_1, \dots, u_N]^\top$, the empirical Bayes estimator of θ , as per Eq. (11), is given by

$$\hat{\theta} = \hat{\lambda}_{ML} \Phi^\top [\hat{\lambda}_{ML} \Phi \Phi^\top + I]^{-1} y$$

which is clearly equal to zero when $\hat{\lambda}_{ML} = 0$.

Application of sparse learning is attracting enormous interest in the context of dynamic network models (Chiuso and Pillonetto 2012; Dankers et al. 2016; Hayden et al. 2016), which arise, for instance, in gene regulatory networks (Daniel et al. 2010) and neuroscience (Prando et al. 2017b; Razi et al. 2017), where structured dynamic dependences should be inferred from measured data; for high-dimensional processes, the combinatorial explosion of alternative structures renders the direct comparison intractable. Regularization allows to avoid such curse of dimensionality, returning sparse (and thus interpretable) models with reasonable computational effort.

Extensions: Regularization for Hybrid Systems Identification and Model Segmentation

An interesting extension of linear systems is a class of so-called hybrid models described by a relation of the form

$$\begin{aligned} y_t &= \hat{y}_{\theta_k}(t|t-1) + e_t \\ \hat{y}_{\theta_k}(t|t-1) &= L_{\theta_k}(y_t^-, u_t^-) \\ \theta_k &\in \mathbb{R}^{n_k} \quad k = 1, \dots, K \end{aligned} \quad (19)$$

where the predictor $\hat{y}_{\theta_k}(t|t-1)$, which is a linear function $L_{\theta_k}(y_t^-, u_t^-)$ of the “past” histories $y_t^- := \{y_{t-1}, y_{t-2}, \dots\}$ and $u_t^- := \{u_{t-1}, u_{t-2}, \dots\}$, is parametrized by a parameter vector $\theta_k \in \mathbb{R}^{n_k}$; there are K different parameter vectors θ_k , $k = 1, \dots, K$, whose evolution over time is determined by a so-called switching mechanism. The name *hybrid* hints at the fact that the model is described continuous-valued (y , u , and e) and discrete-valued (k) variables.

A well-studied subclass of (19) is composed by the so-called switching ARX models, where the predictor takes the special form

$$\hat{y}_{\theta_k}(t|t-1) = \phi_t^\top \theta_k \quad \theta_k \in \mathbb{R}^{n_k} \quad (20)$$

The regressor ϕ_t is a finite vector containing inputs u_s and outputs y_s in a finite past window $s \in [t-1, t-T]$, plus possibly a constant component to model changing “means.” The value of $k \in [1, K]$ is determined by the switching mechanism $p(\phi_t, t) : \mathbb{R}^{n_k} \times \mathbb{R} \rightarrow \{1, \dots, K\}$.

Two extreme but interesting cases are (i) $p(\phi_t, t) = p_t$, where $p(\cdot)$ is an exogenous and not measurable signal, and (ii) $p(\phi_t, t) = p(\phi_t)$, where $p(\cdot)$ is an endogenous unknown measurable function of the regression vector ϕ_t . In any case, from the identification point of view, k at time t is *not* assumed to be known, and, as such, the identification algorithm has to operate without knowledge of this switching mechanism.

Identification of systems in the form (20) requires to estimate (a) the number of models K and the position of the switches between different models, (b) the “dimension” of each model n_k , (c) the value of the parameters θ_k , and, possibly, (d) the function $p(\phi_t, t)$ which determines the switching mechanism.

Steps (b) and (c) are essentially as in section “[System Identification](#)” (see also the introductory paper ▶ “[System Identification: An Overview](#)”); however, this is complicated by steps (a) and (d), which in particular require that one is able to estimate, from data alone, which system is “active” at each time t .

Step (a), which is also related to the problem of *model segmentation*, has been tackled in the

literature; see, e.g., Ozay et al. (2012) and Ohlsson and Ljung (2013), and references therein, by applying suitable penalties on the number of different models K and/or on the number of switches. Note that $p(\phi_t, t) \neq p(\phi_s, s)$ if and only if $\theta_t \neq \theta_s$. Based on this simple observation, one can construct a regularization which counts either the number of switches, i.e.,

$$J_R(\theta; \gamma) := \gamma \sum_{t=2}^N \|\theta_t - \theta_{t-1}\|_0, \quad (21)$$

or attempts to approximate the total number of different models computing

$$J_R(\theta; \gamma) := \gamma \sum_{t,s=1}^N w(s, t) \|\theta_t - \theta_s\|_0 \quad (22)$$

for a suitable weighting $w(t, s)$; see Ohlsson and Ljung (2013).

As discussed above, these quasinorms lead, in general, to unfeasible optimization problems (NP-hard). An exception is the case where one considers bounded noise, i.e., solves a problem of the form

$$\min_{\theta_t} \sum_{t=2}^N \|\theta_t - \theta_{t-1}\|_0 \quad \text{s.t.} \quad \|y_t - \phi_t^\top \theta_t\|_\infty < \epsilon \quad (23)$$

which is shown to be a convex problem; see Ozay et al. (2012). In general relaxations are used, typically using the ℓ_1 /group- ℓ_1 penalties, thus relaxing (21) and (22) to

$$\begin{aligned} J_R(\theta; \lambda) &:= \lambda \sum_{t=2}^N \|\theta_t - \theta_{t-1}\|_1 \\ J_R(\theta; \lambda) &:= \lambda \sum_{t,s=1}^N w(s, t) \|\theta_t - \theta_s\|_1 \end{aligned} \quad (24)$$

This yields to the convex optimization problems:

$$\min_{\theta_t} \sum_t \left(y_t - \phi_t^\top \theta_t \right)^2 + \lambda \sum_{t=2}^N \|\theta_t - \theta_{t-1}\|_1 \quad (25)$$

or

$$\min_{\theta_t} \sum_t \left(y_t - \phi_t^\top \theta_t \right)^2 + \lambda \sum_{t,s=1}^N w(s, t) \|\theta_t - \theta_s\|_1 \quad (26)$$

Summary and Future Directions

We have presented a bird's-eye overview of regularization methods in system identification. By necessity this overview was certainly incomplete, and we encourage the reader to browse through the recent literature for new developments on this exciting topic; we hope the references we have provided are a good starting point. While regularization is quite an old topic, we believe it is fair to say that the nontrivial interaction between regularization and system theoretic concepts provides a wealth of interesting and challenging problems while also providing new powerful tools to tackle challenging practical problems.

Just to mention a few open questions, (i) a thorough analysis of the interaction between different type of priors (e.g., stability vs. complexity vs. structure) is still lacking; (ii) some preliminary results (Formentin and Chiuso 2018) suggest that tailoring regularization to control objectives leads to improved results in terms of control performance; further research in this direction is certainly needed; (iii) the statistical properties of Bayesian procedures such as SBL and its extensions in the context of system identification are not completely understood. Last but not least, while some results are available, nonlinear system identification still offers significant challenges.

Cross-References

- [Nonlinear System Identification Using Particle Filters](#)
- [Nonlinear System Identification: An Overview of Common Approaches](#)
- [Subspace Techniques in System Identification](#)
- [System Identification: An Overview](#)

Recommended Reading

The use of regularization methods for system identification can be traced back to the 1980s; see Doan et al. (1984) and Kitagawa and Gersh (1984); yet it is fair to say that the most significant developments are rather recent, and therefore the literature is not established yet. The reader may consult Fazel et al. (2001), Pillonetto et al. (2011), Chen et al. (2012), Chiuso and Pillonetto (2012), Chiuso (2016), Prando et al. (2017a), Pillonetto and Chiuso (2015), Zorzi and Chiuso (2018), Zorzi and Chiuso (2017), Romeres et al. (2019), and references therein. Clearly all this work has largely benefitted from cross fertilization with neighboring areas, and, as such, very relevant work can be found in the fields of machine learning (Bach et al. 2004; Mackay 1994; Tipping 2001; Rasmussen and Williams 2006), statistics (Hocking 1976; Tibshirani 1996; Fan and Li 2001; Wang et al. 2007; Yuan and Lin 2006; Zou 2006), signal processing (Donoho 2006; Wipf et al. 2011), and econometrics (Banbura et al. 2010).

References

- Aravkin A, Burke J, Chiuso A, Pillonetto G (2014) Convex vs non-convex estimators for regression and sparse estimation: the mean squared error properties of ARD and GLASSO. *J Mach Learn Res* 15:217–252
- Bach F, Lanckriet G, Jordan M (2004) Multiple kernel learning, conic duality, and the SMO algorithm. In: *Proceedings of the 21st international conference on machine learning*, Banff, pp 41–48
- Banbura M, Giannone D, Reichlin L (2010) Large Bayesian VARs. *J Appl Econom* 25:71–92
- Chen T, Ohlsson H, Ljung L (2012) On the estimation of transfer functions, regularizations and Gaussian processes – revisited. *Automatica* 48:1525–1535
- Chiuso A (2016) Regularization and Bayesian learning in dynamical systems: past, present and future. *Annu Rev Control* 41:24–38
- Chiuso A, Pillonetto G (2012) A Bayesian approach to sparse dynamic network identification. *Automatica* 48:1553–1565
- Daniel M, Robert JP, Thomas S, Claudio M, Dario F, Gustavo S (2010) Revealing strengths and weaknesses of methods for gene network inference. *Proc Natl Acad Sci* 107:6286–6291
- Dankers A, Van den Hof PMJ, Bombois X, Heuberger PSC (2016) Identification of dynamic models in com-

- plex networks with prediction error methods: predictor input selection. *IEEE Trans Autom Control* 61:937–952
- Doan T, Litterman R, Sims C (1984) Forecasting and conditional projection using realistic prior distributions. *Econom Rev* 3:1–100
- Donoho D (2006) Compressed sensing. *IEEE Trans Inf Theory* 52:1289–1306
- Fan J, Li R (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. *J Am Stat Assoc* 96:1348–1360
- Fazel M, Hindi H, Boyd S (2001) A rank minimization heuristic with application to minimum order system approximation. In: *Proceedings of the 2001 American control conference*, Arlington, vol 6, pp 4734–4739
- Formentin S, Chiuso A (2018) CoRe: control-oriented regularization for system identification. In: *2018 IEEE conference on decision and control (CDC)*, pp 2253–2258
- Hayden D, Chang YH, Goncalves J, Tomlin CJ (2016) Sparse network identifiability via compressed sensing. *Automatica* 68:9–17
- Hocking RR (1976) A biometrics invited paper. The analysis and selection of variables in linear regression. *Biometrics* 32:1–49
- Kitagawa G, Gersh H (1984) A smoothness priors-state space modeling of time series with trends and seasonalities. *J Am Stat Assoc* 79:378–389
- Leeb H, Pötscher B (2005) Model selection and inference: facts and fiction. *Econom Theory* 21:21–59
- Ljung L (1999) *System identification – theory for the user*. Prentice Hall, Upper Saddle River
- Mackay D (1994) Bayesian non-linear modelling for the prediction competition. *ASHRAE Trans* 100:3704–3716
- Ohlsson H, Ljung L (2013) Identification of switched linear regression models using sum-of-norms regularization. *Automatica* 49:1045–1050
- Ozay N, Szaiaier M, Lagoa C, Camps O (2012) A sparsification approach to set membership identification of switched affine systems. *IEEE Trans Autom Control* 57:634–648
- Pillonetto G, Chiuso A (2015) Tuning complexity in regularized kernel-based regression and linear system identification: the robustness of the marginal likelihood estimator. *Automatica* 58:106–117
- Pillonetto G, De Nicolao G (2010) A new kernel-based approach for linear system identification. *Automatica* 46:81–93
- Pillonetto G, Chiuso A, De Nicolao G (2011) Prediction error identification of linear systems: a nonparametric Gaussian regression approach. *Automatica* 47:291–305
- Pillonetto G, Chen T, Chiuso A, Nicolao GD, Ljung L (2016) Regularized linear system identification using atomic, nuclear and kernel-based norms: the role of the stability constraint. *Automatica* 69:137–149
- Prando G, Chiuso A, Pillonetto G (2017a) Maximum entropy vector kernels for MIMO system identification. *Automatica* 79:326–339
- Prando G, Zorzi M, Bertoldo A, Chiuso A (2017b) Estimating effective connectivity in linear brain network models. In: *2017 IEEE 56th annual conference on decision and control (CDC)*, pp 5931–5936
- Rasmussen C, Williams C (2006) *Gaussian processes for machine learning*. MIT, Cambridge
- Razi A, Seghier ML, Zhou Y, McColgan P, Zeidman P, Park H-J, Sporns O, Rees G, Friston KJ (2017) Large-scale DCMs for resting-state fMRI. *Netw Neurosci* 1:222–241
- Romeres D, Zorzi M, Camoriano R, Traversaro S, Chiuso A (2019, in press) Derivative-free online learning of inverse dynamics models. *IEEE Trans Control Syst Technol*
- Tibshirani R (1996) Regression shrinkage and selection via the LASSO. *J R Stat Soc Ser B* 58:267–288
- Tipping M (2001) Sparse Bayesian learning and the relevance vector machine. *J Mach Learn Res* 1:211–244
- Wang H, Li G, Tsai C (2007) Regression coefficient and autoregressive order shrinkage and selection via the LASSO. *J R Stat Soc Ser B* 69:63–78
- Wipf D, Rao B, Nagarajan S (2011) Latent variable Bayesian models for promoting sparsity. *IEEE Trans Inf Theory* 57:6236–6255
- Yuan M, Lin Y (2006) Model selection and estimation in regression with grouped variables. *J R Stat Soc Ser B* 68:49–67
- Zorzi M, Chiuso A (2017) Sparse plus low rank network identification: a nonparametric approach. *Automatica* 76:355–366
- Zorzi M, Chiuso A (2018) The harmonic analysis of kernel functions. *Automatica* 94:125–137
- Zou H (2006) The adaptive Lasso and its oracle properties. *J Am Stat Assoc* 101:1418–1429

System Identification: An Overview

Lennart Ljung

Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping, Sweden

Abstract

This entry gives an overview of system identification. It outlines the basic concepts in the area and also serves as an umbrella contribution for the related 12 articles on system identifications in this encyclopedia. The basis is the classical statistical approach of para-

metric methods using maximum likelihood and prediction error methods. The paper also describes the properties of the estimated models for large data sets.

Keywords

Asymptotic model properties · Dynamical systems · Estimation · Mathematical models · Maximum likelihood · Parameter estimates · Prediction error method · Regularization

An Introductory Example

System identification is the theory and art of estimating models of dynamical systems, based on observed inputs and outputs. Consider as a concrete example the Swedish aircraft fighter Gripen; see Fig. 1. From one of the earlier test flights, some data were recorded as depicted in Fig. 2.

To design the simulation software and the autopilot, the aircraft manufacturer, the SAAB company, needed a mathematical model for the dynamics of the system. It is a question to describe how, in this case, the pitch rate is affected by the three inputs. A fair amount of knowledge exists about aircraft dynamics, and in industrial practice, “gray-box” models based on Newton’s laws of motion and unknown parameters like aerodynamical derivatives are employed to estimate the flight dynamics. Here, for the purpose of illustrating basic principles, let us just try a simple “black-box” difference equation relation. Denote the output, the pitch rate, at sample number t by $y(t)$, and three control inputs at the same time by $u_k(t)$, $k = 1, 2, 3$. Then assume that we can write

$$\begin{aligned} y(t) = & -a_1 y(t-1) - a_2 y(t-2) - a_3 y(t-3) \\ & + b_{1,1} u_1(t-1) + b_{1,2} u_1(t-2) \\ & + b_{2,1} u_2(t-1) + b_{2,2} u_2(t-2) \\ & + b_{3,1} u_3(t-1) + b_{3,2} u_3(t-2) \end{aligned} \quad (1)$$

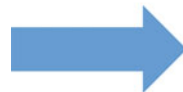
In this simple relationship, we can adjust the parameters to fit the observed data as well as possible by a common least squares fit. We use only the 90 first data points of the observed data. That gives certain numerical values of the 9 parameters below:

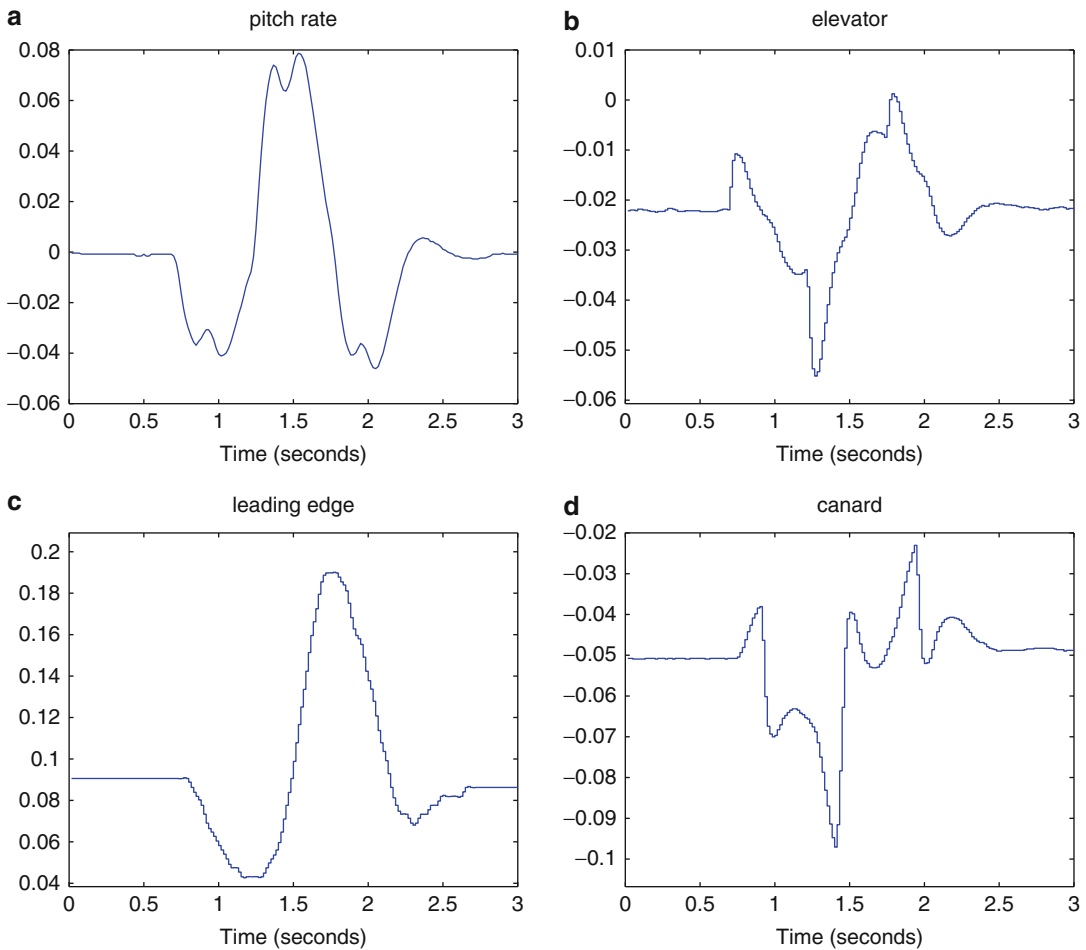
$$\begin{aligned} a_1 = -1.15, \quad a_2 = 0.50, \quad a_3 = -0.35, \\ b_{1,1} = -0.54 \quad b_{1,2} = 0.4, \quad b_{2,1} = 0.15, \\ b_{2,2} = 0.16, \quad b_{3,1} = 0.16, \quad b_{3,2} = 0.07 \end{aligned} \quad (2)$$

We may note that this model is unstable – it has a pole at 1.0026 – but that is in order, because the pitch channel of the real aircraft is unstable at the velocity and altitude in question.

How can we test if this model is reasonable? Since we used only half of the observed data for the estimation, we can test the model on the whole data record. Since the model is unstable it is natural to test it by letting it predict future outputs, say five samples ahead, and compare with the measured outputs. That is done in Fig. 3. We see that the simple model (2) provides quite reasonable predictions over data it has not seen before. This could conceivably be improved if more elaborate model structures than (1) were tried out. Also, in practice more advanced techniques would be required to validate that the estimated model is sufficiently reliable.

System Identification:
An Overview, Fig. 1 The
Swedish aircraft Gripen





System Identification: An Overview, Fig. 2 Data from an early test flight of Gripen. These data cover 3 s of flight and are sampled at 60 Hz. (a) The output: pitch rate.

(b) Control input 1: elevator angle. (c) Control input 2: leading edge flap. (d) Control input 3: canard angle

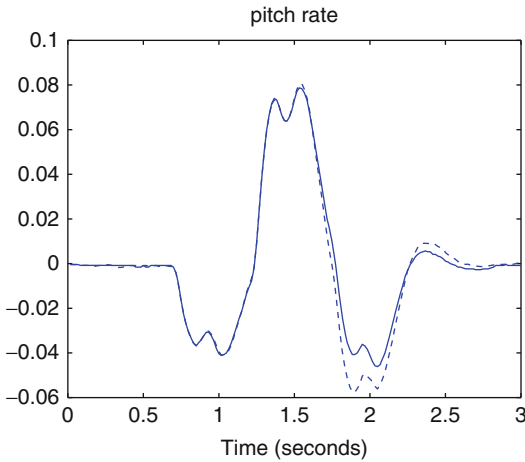
This simple introductory example points to the basic flow of system identification, and it also points to the pertinent issues, which will be listed in the section “[The State-of-the-Art Identification Setup](#).”

Models and System Identification

The Omnipresent Model

It is clear to everyone in science and engineering that *mathematical models* are playing increasingly important roles. Today, model-based design and optimization is the dominant engineering paradigm to systematic design and maintenance

of engineering systems. It has proven very successful and is widely used in basically all engineering disciplines. Concerning control applications, the aerospace industry is the earliest example on a grand scale of this paradigm. This industry was very quick to adopt the theory for model-based optimal control that emerged in the 1960s and is spending great efforts and resources on developing models. In the process industry, model predictive control (MPC) has during the last 25 years become the dominant method to optimize production on an intermediate level. MPC uses dynamical models to predict future process behavior and to optimize the manipulated variables subject to process constraints.



System Identification: An Overview, Fig. 3 The measured output (*solid line*) compared to the five-step-ahead prediction one (*dashed line*)

Increasing demands on performance, efficiency, safety, and environmental aspects are pushing engineering systems to become increasingly complex. Advances in (wireless) communications systems and microelectronics are key enablers for this rapid development, allowing systems to be efficiently interconnected in networks, reducing costs and size, and paving the way for new sensors and actuators.

Model-based techniques are also gaining importance outside engineering applications. Let us just mention systems biology and health care. In the latter case, it is expected that personalized health systems will become more and more important in the future.

Common to the examples given above are the requirements of permeating sensing, actuation, communication, and computation abilities of the engineering systems, in many cases in distributed architectures. It is also clear that these systems should be able to operate in a reliable way in an uncertain and temporally and spatially changing environment. In many applications, cognitive abilities and abilities to adapt will be important. With systems being decentralized and typically containing many actuators, sensors, states, and nonlinearities, but with limited access to sensor information, model building that delivers models of sufficient fidelity becomes very challenging.

System Identification: Data-Driven Modeling

Construction of models requires access to observed data. It could be that the model is developed entirely from information in signals from the system (“black-box models”), or it could be that physical/engineering insights are combined with such information (“gray-box models”). In any case, verification (validation) of a model must be done in the light of measured data. Theories and methodologies for such model construction have been developed in many different research communities (to some extent independently). *System identification* is the term used in the control community for the area of constructing mathematical models of dynamical systems from measured input-output signals. Other communities use other terms for often very similar techniques. The term *machine learning* has become very common in recent years, e.g., Rasmussen and Williams (2006).

System identification has a history of more than 50 years, since the term was coined by Lotfi Zadeh (1956). It is a mature research field with numerous publications, textbooks, conference series, and software packages. It is often used as an example in the control field of an area with good interaction between theory and industrial practice. The backbone of the theory relies upon statistical grounds, with maximum likelihood methods and asymptotic analysis (in the number of observed data). The goal of the system identification field is to find a model of the plant in question as well as of its disturbances and also to find a characterization of the uncertainty bounds of the description.

The State-of-the-Art Identification Setup

To approach a system identification problem, like in section “[An Introductory Example](#),” a number of questions need to be answered, such as

- What model type, e.g., (1) should be used?
- How should the parameters in the model be adjusted?
- What inputs should be applied to obtain a good model?
- How do we assess the quality of the model?
- How do we gain confidence in an estimated model?

There is a very extensive literature on the subject, with many textbooks, like Ljung (1999), Söderström and Stoica (1989), and Pintelon and Schoukens (2012).

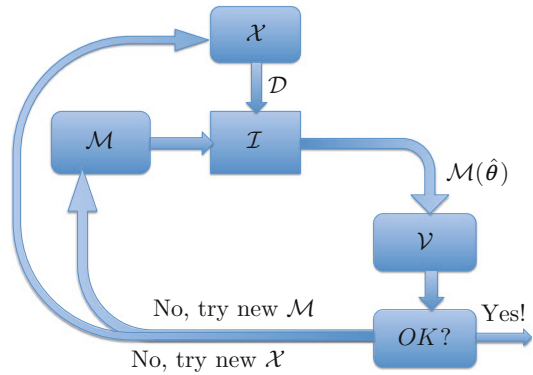
System identification is characterized by five basic concepts:

- $\mathcal{X}$: The experimental conditions under which the data is generated
- $\mathcal{D}$: The data
- $\mathcal{M}$: The model structure and its parameters θ
- $\mathcal{I}$: The identification method by which a parameter value $\hat{\theta}$ in the model structure $\mathcal{M}(\theta)$ is determined based on the data $\mathcal{D}$
- $\mathcal{V}$: The validation process that scrutinizes the identified model

See Fig. 4. For off-line (batch) application the steps are typically an iterative process to navigate to a model that passes through the validation test (“is not falsified”), involving revisions of the necessary choices. For several of the steps in this loop, helpful support tools have been developed. It is however not quite possible or desirable to fully automate the choices, since subjective perspectives related to the intended use of the model are very important. For online (recursive) identification (see, e.g., Young (2011)), the same considerations apply, even though it might be more cumbersome to implement revised choices in real time.

$\mathcal{M}$: Model Structures

A model structure $\mathcal{M}$ is a parameterized collection of models that describe the relations between the inputs u and outputs y of the system. The



System Identification: An Overview, Fig. 4 The identification work loop

parameters are denoted by θ , so $\mathcal{M}(\theta)$ is a particular model. The model set then is

$$\mathcal{M} = \{\mathcal{M}(\theta) | \theta \in D_{\mathcal{M}}\} \quad (3)$$

Many ways exist to collect mathematical expressions that encompass a model; see, e.g., ▶ “Modeling of Dynamic Systems from First Principles”, ▶ “Nonlinear System Identification: An Overview of Common Approaches”, and ▶ “Nonlinear System Identification Using Particle Filters”. The models may be both linear and nonlinear as well as time invariant and time varying, and it is useful to have as a common ground that a model gives a rule to predict (one-step-ahead) the output at time t , i.e., $y(t)$ (a p -dimensional column vector), based on observations of previous input-output data up to time $t - 1$ (denoted by Z^{t-1}).

$$\hat{y}(t|\theta) = g(t, \theta, Z^{t-1}) \quad (4)$$

This covers a broad variety of model descriptions, in a somewhat abstract way. The descriptions become much more explicit when we specialize to linear models.

A note on “inputs” It is important to include all measurable disturbances that affect y among the inputs u to the system, even if they cannot be manipulated as control inputs. In some cases the system may entirely lack measurable inputs,

so the model (4) then just describes how future outputs can be predicted from past ones. Such models are called *time series* and correspond to systems that are driven by unobservable disturbances. Most of the techniques described in this entry apply also to such models.

A note on disturbances A complete model involves both a description of the input-output relations and a description of how various noise sources affect the measurements. The noise description is essential to understand both the quality of the model predictions and the model uncertainty. Proper control design also requires a picture of the disturbances in the system.

Linear Models

For linear time invariant systems, a general model structure is given by the transfer function G from input u to output y and the transfer function H from a white noise source e to output additive disturbances (for notational convenience, we specialize to single-input-single-output systems, but all expressions are valid in the multivariable case with simple notational changes):

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \quad (5a)$$

$$Ee^2(t) = \sigma^2; \quad Ee(t)e^T(k) = 0 \text{ if } k \neq t \quad (5b)$$

(E denotes mathematical expectation.) This model is in discrete time, and q denotes the shift operator $qy(t) = y(t+1)$. We assume for simplicity that the sampling interval is a one-time unit. The expansion of $G(q, \theta)$ in the inverse (backwards) shift operator gives the *impulse response* of the system:

$$\begin{aligned} G(q, \theta)u(t) &= \sum_{k=1}^{\infty} g_k(\theta)q^{-k}u(t) \\ &= \sum_{k=1}^{\infty} g_k(\theta)u(t-k) \end{aligned} \quad (6)$$

The discrete time Fourier transform (or the z -transform of the impulse response, evaluated in $z = e^{i\omega}$) gives the *frequency response* of the system:

$$G(e^{i\omega}, \theta) = \sum_{k=1}^{\infty} g_k(\theta)e^{-ik\omega} \quad (7)$$

The function G describes how an input sinusoid shifts phase and amplitude when it passes through the system.

The additive noise term $v = He$ is quite versatile, and with a suitable choice of H (which is *monic*, i.e., its expansion in q^{-1} starts with a “1”), it can describe a disturbance with arbitrary spectrum. To link with the predictor as a unifying model concept, it is useful to compute the predictor for (5a) (the conditional mean of $y(t)$ given past data), which is

$$\begin{aligned} \hat{y}(t|\theta) &= G(q, \theta)u(t) + [1 - H^{-1}(q, \theta)] \\ &\quad [y(t) - G(q, \theta)u(t)] \end{aligned} \quad (8)$$

Note that the expansion of $[1 - H^{-1}]$ starts with a delay term $\tilde{h}_1 q^{-1}$, since H is monic, so there is a delay in y in the predictor. It is natural to interpret the expression as a simulation using the input u , adjusted with a prediction of the additive disturbance $v(t)$ at time t , based on past values of v . The predictor is thus an easy reformulation of the basic transfer functions G and H . The question now is how to parameterize these.

Black-Box Models

A *black-box* model uses no physical insight or interpretation but is just a general and flexible parameterization. It is natural to let G and H be rational in the shift operator:

$$G(q, \theta) = \frac{B(q)}{F(q)}; \quad H(q, \theta) = \frac{C(q)}{D(q)} \quad (9a)$$

$$B(q) = b_1 q^{-1} + b_2 q^{-2} + \dots + b_{nb} q^{-nb} \quad (9b)$$

$$F(q) = 1 + f_1 q^{-1} + \dots + f_{nf} q^{-nf} \quad (9c)$$

$$\theta = [b_1, \dots, c_1, \dots, d_1, \dots, f_{nf}] \quad (9d)$$

C and D are like F *monic*, i.e., start with a “1.”

A very common case is that $F = D = A$ and $C = 1$ which gives the *ARX model* (autoregressive with exogenous input):

$$y(t) = A^{-1}(q)B(q)u(t) + A^{-1}(q)e(t) \text{ or} \quad (10a)$$

$$A(q)y(t) = B(q)u(t) + e(t) \text{ or} \quad (10b)$$

$$y(t) + a_1y(t-1) + \dots + a_nay(t-na) \quad (10c)$$

$$= b_1u(t-1) + \dots + b_nbu(t-nb) + e(t) \quad (10d)$$

This is the model structure we used in (1) in the introductory example, but for several inputs.

Other common black-box structures of this kind are FIR (finite impulse response model, $F = C = D = 1$), ARMAX (autoregressive moving average with exogenous input, $F = D = A$), and BJ (Box-Jenkins, all four polynomials are different).

Gray-Box Models

If some physical facts are known about the system, it is possible to build that into a *gray-box model*. It could, for example, be that for the airplane in the introduction, the motion equations are known from Newton's laws, but certain parameters are unknown, like the aerodynamical derivatives. Then it is natural to build a continuous-time state-space model from physical equations:

$$\begin{aligned} \dot{x}(t) &= A(\theta)x(t) + B(\theta)u(t) \\ y(t) &= C(\theta)x(t) + D(\theta)u(t) + v(t) \end{aligned} \quad (11)$$

Here θ are simply some entries of the matrices A, B, C, D , corresponding to unknown physical parameters, while the other matrix entries signify known physical behavior. This model can be sampled with well-known sampling formulas (obeying the input inter-sample properties, zero-order hold, or first-order hold) to give

$$\begin{aligned} x(t+1) &= \mathcal{F}(\theta)x(t) + \mathcal{G}(\theta)u(t) \\ y(t) &= C(\theta)x(t) + D(\theta)u(t) + w(t) \end{aligned} \quad (12)$$

The model (12) has the transfer function from u to y

$$G(q, \theta) = C(\theta)[qI - \mathcal{F}(\theta)]^{-1}\mathcal{G}(\theta) + D(\theta) \quad (13)$$

so we have achieved a particular parameterization of the general linear model (5a).

Continuous-Time Models

The general model description (4) describes how the predictions evolve in discrete time. But in many cases, we are interested in continuous-time (CT) models, like models for physical interpretation and simulation (e.g., electrical circuit simulators like ADS, Spice, Spectre, and Microwave Office use continuous-time models). But CT model estimation is contained in the described framework, as the linear state-space model (11) illustrates. More comments on direct estimation of CT models are given in section “[Estimating Continuous Time Models](#).”

Nonlinear Models

A nonlinear model is a relation (4), where the function g is nonlinear in the input-output data Z . There is a rich variation in how to specify the function g more explicitly. A quite general way is the nonlinear state-space equation, which is a counterpart to (12):

$$\begin{aligned} x(t+1) &= f(x(t), u(t), v(t), \theta) \\ y(t) &= h(x(t), e(t), \theta) \end{aligned} \quad (14)$$

where v and e are white noises. This is further discussed in ► “[Nonlinear System Identification using Particle filters](#)”, where x is described as a Markov process with v defining the transitions, and in ► [Nonlinear System Identification: An Overview of Common Approaches](#)”, where (14) (with $v \equiv 0$) is related to a continuous-time gray-box model. The latter article also discusses several other nonlinear model structures that can be seen as extensions and modifications of linear models: nonlinear mappings of past input-output data corresponding to (10), mixing static nonlinearities with linear dynamical models, etc.

7: Identification Methods: Criteria

The goal of identification is to match the model to the data. Here the basic techniques for such matching will be discussed.

Time Domain Data

Suppose now we have collected a data record in the time domain

$$Z^N = \{u(1), y(1), \dots, u(N), y(N)\} \quad (15)$$

Since the model is in essence a predictor, it is quite natural to evaluate it by how well it predicts the measured output. So, form the prediction errors for (4):

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta) \quad (16)$$

The “size” of this error can be measured by some scalar norm:

$$\ell(\varepsilon(t, \theta)) \quad (17)$$

and the performance of the predictor over the whole data record Z^N becomes

$$V_N(\theta) = \sum_{t=1}^N \ell(\varepsilon(t, \theta)) \quad (18)$$

A natural parameter estimate is then

$$\hat{\theta}_N = \arg \min_{\theta \in D_{\mathcal{M}}} V_N(\theta) \quad (19)$$

This is the *prediction error method (PEM)* and is applicable to general model structures. See, e.g., Ljung (1999) or (2002) for more details. See also ► “Nonlinear System Identification: An Overview of Common Approaches”.

The PEM approach can be embedded in a statistical setting to guarantee optimal statistical properties. The ML methodology below offers a systematic framework to do so.

A Maximum Likelihood View

If the system innovations e have a probability density function (pdf) $f(x)$, then the criterion function (18) with $\ell(x) = -\log f(x)$ will be the logarithm of the *likelihood function*. See Lemma 5.1 in Ljung (1999). More specifically, assume that the system has p outputs and that the innovations are Gaussian with zero mean and covariance matrix Λ , so that

$$y(t) = \hat{y}(t|\theta) + e(t), \quad e(t) \in N(0, \Lambda) \quad (20)$$

for the θ that generated the data. Then it follows that the negative logarithm of the likelihood function for estimating θ from y is

$$L_N(\theta) = \frac{1}{2} [V_N(\theta) + N \log \det \Lambda + Np \log 2\pi] \quad (21)$$

where $V_N(\theta)$ is defined by (18), with

$$\ell(\varepsilon(t, \theta)) = \varepsilon^T(t, \theta) \Lambda^{-1} \varepsilon(t, \theta) \quad (22)$$

So the maximum likelihood model estimate (MLE) for known Λ is obtained by minimizing $V_N(\theta)$. If Λ is not known, it can be included among the parameters and estimated (Ljung 1999, page 218), which results in a criterion

$$D_N(\theta) = \det \sum_{t=1}^N \varepsilon(t, \theta) \varepsilon^T(t, \theta) \quad (23)$$

to be minimized.

The EM Algorithm

The EM algorithm (Dempster et al. 1977) is closely related to the ML technique. It is a method that is especially useful when the ML criterion is difficult to evaluate from the observed data but would be easier to find if certain unobserved *latent* variables were known. The algorithm alternates between an expectation step estimating the log likelihood and a maximization step bringing the parameter estimate closer in each step to the MLE. Its application to the nonlinear state-space model (14) is described in ► “Nonlinear System Identification Using Particle Filters”.

Regularization

Solving for the estimate in (19) is a so-called inverse problem, which means that the solution may be ill conditioned. To deal with that in (18), we could add a quadratic norm:

$$W_N(\theta) = V_N(\theta) + \lambda(\theta - \theta^\dagger)^T R(\theta - \theta^\dagger) \quad (24)$$

(λ is a scaling, R is a positive semidefinite (psd) matrix, and $\theta^\dagger$ is a nominal value of the parameters). The estimate is then found by minimizing $W_N(\theta)$. The criterion (24) makes sense in a classical estimation framework as an ad hoc modification of the MLE to deal with possible ill-conditioned minimization problems. The added quadratic term then serves as proper (*Tikhonov*) *regularization* of an ill-conditioned inverse problem; see, for example, Tikhonov and Arsenin (1977). This criterion is a clear-cut balance between model fit and a penalty on the model parameter size. The amount of penalty is governed by λ and R .

Other useful regularization penalties could be to add an ℓ_1 norm of the parameter. Such techniques are further discussed in ► “[System Identification Techniques: Convexification, Regularization, Relaxation](#)”.

Bayesian View

For a broader perspective, it is useful to invoke a Bayesian view. Then the sought parameter vector θ is itself a random vector with a certain pdf. This random vector will of course be correlated with the observations y . If we assume that the *prior distribution* of θ (before y has been observed) is Gaussian with mean θ^* and covariance matrix Π ,

$$\theta \in N(\theta^*, \Pi) \quad (25)$$

its prior pdf is

$$P(\theta) = \frac{1}{\sqrt{(2\pi)^p \det(\Pi)}} e^{-(\theta - \theta^*)^T \Pi^{-1} (\theta - \theta^*)/2} \quad (26)$$

The posterior (after y has been measured) pdf then is by Bayes rule (Y denoting all measured

y signals):

$$P(\theta|Y) = \frac{P(\theta, Y)}{P(Y)} = \frac{P(Y|\theta)P(\theta)}{P(Y)} \quad (27)$$

In the last step, $P(Y|\theta)$ is the likelihood function (cf. the negative log likelihood function $L_N(\theta)$ in (21)), $P(\theta)$ is the prior pdf (26), and $P(Y)$ is a θ -independent normalization. Apart from this normalization, and other θ -independent terms, twice the negative logarithm of (27) equals $W_N(\theta)$ in (24) with

$$\lambda R = \Pi^{-1} \quad (28)$$

That means that with (28), the regularized estimate from (24) is the *maximum a posteriori* (MAP) estimate. As more and more data become available, the role of the prior will tend to zero, so as $N \rightarrow \infty$ the MAP Estimate $\rightarrow$ MLE.

This Bayesian interpretation of the regularized estimate also gives a clue to select the regularization quantities λ , R , and θ^* .

For black-box models, a reasonable prior (Π, θ^*) may not be available. Then it is possible to parameterize them with *hyperparameters* α and then estimate these through the marginal likelihood:

$$\hat{\alpha} = \arg \max P(Y|\alpha) \quad (29)$$

A survey of how such techniques may improve system identification techniques is given in Pilonetti et al. (2014).

More aspects of the Bayesian view of system identification are given in ► “[System Identification Techniques: Convexification, Regularization, Relaxation](#)” and in ► “[Nonlinear System Identification Using Particle Filters](#)”.

Frequency Domain Data

Frequency domain data are obtained either from frequency analyzers or by applying the Fourier transform to measured time domain data. The data could be in the input-output form

$$Y_N(e^{i\omega_k}), U_N(e^{i\omega_k}), k = 1, 2, \dots, M \quad (30)$$

$$Y_N(z) = \frac{1}{\sqrt{N}} \sum_{k=1}^N y(k)z^{-k} \quad (31)$$

or being observed samples from the frequency function

$$\hat{G}_N(e^{i\omega_k}), \quad k = 1, 2, \dots, M \quad (32)$$

$$\text{e.g., } \hat{G}_N(e^{i\omega}) = \frac{Y_N(e^{i\omega})}{U_N(e^{i\omega})} \text{ (ETFE)} \quad (33)$$

((33) is the *empirical transfer function estimate*, ETFE).

Linear Parametric Models

By taking the Fourier transform of (5a), we see that

$$Y(e^{i\omega}) = G(e^{i\omega}, \theta)U(e^{i\omega}) \quad (34)$$

plus a noise term that has variance

$$\sigma^2 |H(e^{i\omega}, \theta)|^2 \quad (35)$$

Simple least squares (LS) curve fitting of (34) says that we should fit observations with weights that are inversely proportional to the measurement variance. That gives the weighted LS criterion

$$V_N(\theta) = \sum_{k=1}^M |Y(e^{i\omega_k}) - G(e^{i\omega_k}, \theta)U_N(e^{i\omega_k})|^2 / |H(e^{i\omega_k}, \theta)|^2 \quad (36)$$

(the constant σ^2 does not affect the minimization of V_N).

It can readily be verified that (36) coincides with (18), $(\ell(\varepsilon) = |\varepsilon|^2)$ by Parseval's identity in case $M = N$ and the frequencies ω_k are selected as the DFT grid.

Notice that (36) can be written as

$$V_N(\theta) = \sum_{k=1}^M \left| \frac{Y_N(e^{i\omega_k})}{U_N(e^{i\omega_k})} - G(e^{i\omega_k}, \theta) \right|^2$$

$$\left| \frac{U_N(e^{i\omega_k})}{H(e^{i\omega_k}, \theta)} \right|^2 \quad (37)$$

We can see that as a properly weighted curve fitting of the frequency function to the ETFE (33).

See ► [“Frequency Domain System Identification”](#) for more details of using frequency domain data for estimating dynamical systems.

Nonparametric Methods

From frequency domain data, the frequency response functions $G(e^{i\omega})$, $H(e^{i\omega})$ can also be estimated directly as functions without any parametric model. See ► [“Nonparametric Techniques in System Identification”](#) for a detailed account of this.

IV and Subspace Methods

Instrumental Variables

The family of identification methods that can be described as minimizing a specific criterion function, like (19), covers many theoretically and practically important techniques. Still, several methods do not belong to this family. A useful technique is to characterize a good model, as one that gives prediction errors that are uncorrelated with available information:

$$\hat{\theta} = \text{sol}_{\theta \in D} \mathcal{M} \sum_{t=1}^N \varepsilon(t, \theta) \zeta(t, \theta) = 0 \quad (38)$$

Here, $\varepsilon(t, \theta)$ is the prediction error (16), and sol means “solution to.” The sequence $\{\zeta(t), t = 1, \dots, N\}$ is constructed from the observed data, possibly also dependent on some design variables that are included in θ . Typically $\zeta(t)$ is constructed from past inputs, so a good model should not have prediction errors that are correlated with past observations. The variables ζ are called *instrumental variables*, and there is an extensive literature about how to select these. See, e.g., Ljung (1999), Section 7.5, Söderström and Stoica (1983), and Young (2011).

Subspace Methods

A related technique is to estimate black-box state-space models like (12) (without any internal parametric structure) by realizing the states from data and then estimating the matrices by least squares method. This gives a powerful family of methods for state-space model estimation. They are described in detail in ► “Subspace Techniques in System Identification”. The major advantage of subspace methods is that they easily apply to multiple-input-multiple-output systems and are non-iterative. A drawback is that the model properties and their dependence on certain design variables are not fully known.

Errors-in-Variables (EIV) Techniques

The estimation techniques described so far assume that the input has been measured without errors. In some cases, it is natural to assume that both inputs and outputs have measurement errors. The estimation problem then becomes more difficult, and some kind of knowledge about the measurement errors is typically required. In Pintelon and Schoukens (2012), Section 8.2, it is described how criteria of the type (36) are modified in the presence of input noise, and Söderström (2018) can be consulted for a summarizing treatise on EIV techniques. See also the section “Errors-in-Variables Framework” in ► “Frequency Domain System Identification”.

Asymptotic Properties of the Estimated Models

An estimated model is useless, unless something is known about its reliability and error bounds. Therefore, it is important to analyze the model properties.

Bias and Variance

The observations, certainly of the output from the system, are affected by noise and disturbances, which of course also will influence the estimated model parameters (19). The disturbances are typically described as stochastic processes, which makes the estimate $\hat{\theta}_N$ a *random variable*.

This has a certain pdf, and often the analysis is restricted to its mean and variance only. The difference between the mean and a true description of the system measures the *bias* of the model. If the mean coincides with the true system, the estimate is said to be *unbiased*. The total error in a model thus has two contributions: the bias and the variance.

Properties of the PEM Estimate (19)

as $N \rightarrow \infty$

Except in simple special cases, it is quite difficult to compute the pdf of the estimate $\hat{\theta}_N$. However, its *asymptotic properties* as $N \rightarrow \infty$ are easier to establish. The basic results can be summarized as follows (E denotes mathematical expectation; see Ljung (1999), Chapters 8 and 9, for a more complete treatment):

- **Limit Model:**

$$\begin{aligned} \hat{\theta}_N &\rightarrow \theta^* \\ &= \arg \min \left[\lim_{N \rightarrow \infty} \frac{1}{N} V_N(\theta) \approx \text{El}(\varepsilon(t, \theta)) \right] \end{aligned} \quad (39)$$

So the estimate will converge to the best possible model, in the sense that it gives the smallest average prediction error.

- **Asymptotic Covariance Matrix for Scalar Output Models:**

In case the prediction errors $e(t) = \varepsilon(t, \theta^*)$ for the limit model are approximately white, the covariance matrix of the parameters is asymptotically given by

$$\text{Cov} \hat{\theta}_N \sim \frac{\kappa(\ell)}{N} \left[\text{Cov} \frac{d}{d\theta} \hat{y}(t|\theta) \right]^{-1} \quad (40)$$

So the covariance matrix of the parameter estimate is given by the inverse covariance matrix of the gradient of the predictor wrt the parameters. Here (prime denoting derivatives)

$$\kappa(\ell) = \frac{E[\ell'(e(t))]^2}{E[\ell''(e(t))]^2} \quad (41)$$

Note that

$$\kappa(\ell) = \sigma^2 = E e^2(t) \quad \text{if } \ell(e) = e^2/2$$

If the model structure contains the true system, it can be shown that its covariance matrix is the smallest that can be achieved by any unbiased estimate, in case the norm ℓ is chosen as the logarithm of the pdf of e . That is, it fulfills the *Cramér-Rao inequality* (Cramér 1946).

These results are valid for quite general model structures. Now, specialize to linear models (5a) and assume that the true system is described by

$$y(t) = G_0(q)u(t) + H_0(q)e(t) \quad (42)$$

which could be general transfer functions, possibly much more complicated than the model. Then

$$\theta^* = \arg \min_{\theta} \int_{-\pi}^{\pi} |G(e^{i\omega}, \theta) - G_0(e^{i\omega})|^2 \frac{\Phi_u(\omega)}{|H(e^{i\omega}, \theta)|^2} d\omega \quad (43)$$

That is, the frequency function of the limiting model will approximate the true frequency function as well as possible in a frequency norm given by the input spectrum Φ_u and the noise model.

For a linear black-box model

$$\text{Cov}G(e^{i\omega}, \hat{\theta}_N) \sim \frac{n}{N} \frac{\Phi_v(\omega)}{\Phi_u(\omega)} \quad \text{as } n, N \rightarrow \infty \quad (44)$$

where n is the model order and Φ_v is the noise spectrum $\sigma^2 |H_0(e^{i\omega})|^2$. The variance of the estimated frequency function at a given frequency is, thus, for a high-order model, proportional to the noise-to-signal ratio at that frequency. That is a natural and intuitive result.

Trade-Off Between Bias and Variance

Generally speaking the quality of the model depends on the quality of the measured data and the flexibility of the chosen model structure (3). A more flexible model structure typically has smaller bias, since it is easier to come

closer to the true system. At the same time, it will have a higher variance: with higher flexibility it is easier to be fooled by disturbances. So the trade-off between bias and variance to reach a small total error is a choice of balanced flexibility of the model structure.

As the model gets more flexible, the fit to the estimation data in (19), $V_N(\hat{\theta}_N)$, will always improve. To account for the variance contribution, it is thus necessary to modify this fit to assess the total quality of the model. A much used technique for this is Akaike's criterion (AIC) (Akaike 1974):

$$\hat{\theta}_N = \arg \min_{\mathcal{M}, \theta \in D_{\mathcal{M}}} 2L_N(\theta) + 2\dim\theta \quad (45)$$

where L_N is the negative log likelihood function. The minimization also takes place over a family of model structures with different number of parameters ($\dim \theta$).

For Gaussian innovations e with unknown and estimated variance, AIC takes the form

$$\hat{\theta}_N = \arg \min_{\mathcal{M}, \theta \in D_{\mathcal{M}}} \left[\log \det \left[\frac{1}{N} \sum_{t=1}^N \varepsilon(t, \theta) \varepsilon^T(t, \theta) \right] + 2 \frac{\dim\theta}{N} \right] \quad (46)$$

after normalization and omission of model-independent quantities.

A variant of AIC is to put a higher penalty on the model complexity:

$$\hat{\theta}_N = \arg \min [2L_N(\theta) + \dim\theta \log N] \quad (47)$$

This is known as Bayesian information criterion (BIC) or Rissanen's minimum description length (MDL) criterion (Rissanen 1978).

Section “[3: Model Validation](#)” contains further aspects on the choice of model structure.

⌘: Experiment Design

Experiment design is the question of choosing which signal to measure, the sampling rate, and designing the input.

The theory of experiment design primarily relies upon analysis of how the asymptotic parameter covariance matrix (40) depends on the design variables: so the essence of experiment design can be symbolized as

$$\min_{\mathcal{X}} \text{trace}\{C[E\psi(t)\psi^T(t)]^{-1}\}$$

where ψ is the gradient of the prediction wrt the parameters and the matrix C is used to weight variables reflecting the intended use of the model.

For linear systems the input design is often expressed as selecting the spectrum (frequency contents) of u .

This leads to the following recipe: Let the input's power be concentrated to frequency regions where a good model fit is essential and where disturbances are dominating.

Issues of experiment design are treated in much more detail in ► “Experiment Design and Identification for Control”.

The measurement setup, like if band-limited inputs are used to estimate continuous-time models and how the experiment equipment is instrumented with band pass filters (see, e.g., Pintelon and Schoukens 2012, Sections 13.2–3), also belongs to the important experiment design questions.

⌘: Model Validation

Model validation is about examining and scrutinizing an estimated model to check if it can be used for its purpose. These methods unavoidably are problem dependent and contain several subjective elements, and no conclusive procedure for validation can be given. A few useful techniques will be listed in this

section. Basically it is a matter of trying to falsify a model under the conditions it will be used for and also to gain confidence in its ability to reproduce new data from the system.

Falsifying Models: Residual Analysis

An estimated model is never a correct description of a true system. In that sense, a model cannot be “validated.” Instead it is instructive to try and *falsify* it, i.e., confront it with facts that may contradict its correctness. A good principle is to look for the *simplest unfalsified model*; see, e.g., Popper (1934).

Residual analysis is the leading technique for falsifying models: The residuals, or one-step-ahead prediction errors $\hat{\varepsilon}(t) = \varepsilon(t, \hat{\theta}_N) = y(t) - \hat{y}(t|\hat{\theta}_N)$, should ideally not contain any traces of past inputs or past residuals. If they did, it means that the predictions are not ideal. So, it is natural to test the correlation functions

$$\hat{r}_{\hat{\varepsilon},u}(k) = \frac{1}{N} \sum_{t=1}^N \hat{\varepsilon}(t+k)u(t) \quad (48)$$

$$\hat{r}_{\hat{\varepsilon}}(k) = \frac{1}{N} \sum_{t=1}^N \hat{\varepsilon}(t+k)\hat{\varepsilon}(t) \quad (49)$$

and check that they are not larger than certain thresholds. Here N is the length of the data record, and k typically ranges over a fraction of the interval $[-N \ N]$. See, e.g., Ljung (1999), Section 16.6 for more details.

Comparing Different Models

When several models have been estimated, it is a question to choose the “best one.” Then, models that employ more parameters naturally show a better fit to the data, and it is necessary to outweigh that. The model selection criteria AIC (46) and BIC (47) are examples of how such decisions can be taken. They can be extended to regular hypothesis tests where

more complex models are accepted or rejected at various test levels (Ljung 1999, Sect. 16.4).

Making comparisons in the frequency domain is a very useful complement for domain experts who are used to thinking in terms of natural frequencies, natural damping, etc.

Cross Validation

Cross validation is an important statistical concept that loosely means that the model performance is tested on a data set (*validation data*) other than the estimation data. There is an extensive literature on cross validation, e.g., Stone (1977), and many ways to split up available data into estimation and validation parts have been suggested. A simple way, often used in system identification, is to use one-half of the data to estimate the model and the other half to evaluate simulation or prediction fit. Trying out different model structures (or other decision variables, like regularization parameters), one then picks the choice that gives the best performance on validation data.

Other Topics

Connections to Machine Learning

In today's estimation and decision world, *machine learning* has become a central term for constructing models from observed data. It has thus clear links to system identification. The core of all machine learning can be said to be *function estimation*: Measure noisy observations of a function between two spaces $\mathcal{X}$ to $\mathcal{Y}$

$$y(t) = F(x(t)) + \text{noise} \quad (50)$$

and infer the function F . From a statistical perspective, this “curve-fitting problem” is a standard *non-linear regression*. Most applications of machine learning have a *black-box* view of F , i.e., no physical knowledge about the function is employed, but only flexible mappings are used when estimating it.

In machine learning a basic technique to avoid a perfect (and useless) fit to the observed data $x(t_i)$, $y(t_i)$ is to employ *Regularization* as in (24). It turns out that this can also be interpreted as the *Gaussian Processes approach*, described in ▶ “[Use of Gaussian Processes in System Identification](#)” and Rasmussen and Williams (2006). See also ▶ “[System Identification Techniques: Convexification, Regularization, Relaxation](#)”.

Another basic machine learning technique is to use flexible general parameterizations of the function F , such as neural networks; see ▶ “[Nonlinear System Identification: An Overview of Common Approaches](#)”. *Deep learning* is a very popular term used when the model is parametrized as a deep network; see ▶ “[Deep Learning in a System Identification Perspective](#)”.

Numerical Algorithms and Software Support

The central numerical task to estimate the model lies in the innocent-looking “arg min” in (38). Since the criterion often is non-convex, this global minimization can be nontrivial. Typically some iterative numerical optimization method, like Gauss-Newton, Levenberg-Marquardt, or trust regions, e.g., Nocedal and Wright (2012), is employed. The iterations are initiated at a carefully selected point, for black-box linear systems often based on ARX or subspace estimates.

The practical use of system identification relies upon efficient software support. Many such packages exist. They are further treated along with numerical and computational aspects in ▶ “[System Identification Software](#)”.

Estimating Continuous-Time Models

Most of the techniques described here formally seem to deal with estimating discrete time models. However continuous-time (CT) models are to be preferred in many contexts, and most of the modeling of physical systems really concerns CT models. A natural approach is to do physical modeling in continuous time as in (11) and then do esti-

mation of the CT matrices via the sampled model (12). All the described algorithms and results apply to this approach to CT model estimation. Another approach is to use band-limited inputs and compute the CT Fourier transforms of data (that coincide with the discrete time transforms for band-limited data) and apply ► [“Frequency Domain System Identification”](#).

Yet another approach is to directly fit CT model parameters to discrete time data, using specially designed filters; see, e.g., Garnier and Wang (2008).

Recursive Estimation

For certain adaptive and in-line applications, it may be necessary to continuously compute the models by recursively updating the estimates. The techniques for that resemble state estimation algorithms and are dealt with in a general setting in ► [“Nonlinear System Identification Using Particle Filters”](#). See also Ch 11 in Ljung (1999).

Data Management

The collected data often requires particular attention before it can be used for estimation. Issues like missing observations, obviously erroneous values (outliers), slowly varying disturbances, trends, etc., need attention. In industrial applications, a practical question is often to select portions of the data records that contain relevant information for the model building. Such questions are application dependent and related to experiment design and also to database management. Some techniques for preparing data for identification are mentioned in Ch 14 of Ljung (1999).

Summary and Future Directions

System identification is a mature and well-established area in automatic control. The methods are successfully and routinely applied in industrial practice, and the understanding of theoretical issues is mostly

excellent. The standard theory relies very much on basic statistical concepts and methods.

What is exciting about future development is what increased computation power may mean for the area: Can nonlinear models be efficiently estimated by massive computational efforts? Will tools inspired by machine learning turn out to be superior to the conventional approaches? Can reliable uncertainty regions be computed for arbitrary noises and without the asymptotic formulas?

Several of these questions are illuminated in the articles listed under Cross-References.

Cross-References

- [Deep Learning in a System Identification Perspective](#)
- [Experiment Design and Identification for Control](#)
- [Frequency Domain System Identification](#)
- [Modeling of Dynamic Systems from First Principles](#)
- [Nonlinear System Identification: An Overview of Common Approaches](#)
- [Nonlinear System Identification Using Particle Filters](#)
- [Nonparametric Techniques in System Identification](#)
- [Nonparametric Techniques in System Identification: The Time-Varying and Missing Data Cases](#)
- [Subspace Techniques in System Identification](#)
- [System Identification Software](#)
- [System Identification Techniques: Convexification, Regularization, Relaxation](#)
- [Use of Gaussian Processes in System Identification](#)

Recommended Reading

A textbook that covers and extends the material in this contribution is Ljung (1999). Another textbook in the same spirit is Söderström and Stoica (1989), while Pintelon

and Schoukens (2012) give a comprehensive treatment of frequency domain methods. Recursive methods are treated in Young (2011).

Acknowledgments Support from the European Research Council under the advanced grant LEARN, contract 267381, is gratefully acknowledged.

Bibliography

- Akaike H (1974) A new look at the statistical model identification. *IEEE Trans Autom Control* AC-19:716–723
- Cramér H (1946) *Mathematical methods of statistics*. Princeton University Press, Princeton
- Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the EM algorithms. *J R Stat Soc Ser B* 39(1):1–38
- Garnier H, Wang L (eds) (2008) *Identification of continuous-time models from sampled data*. Springer, London
- Ljung L (1999) *System identification – theory for the user*, 2nd edn. Prentice-Hall, Upper Saddle River
- Ljung L (2002) Prediction error estimation methods. *Circuits Syst Signal Process* 21(1):11–21
- Nocedal J, Wright J (2012) *Numerical optimization*, 2nd edn. Springer, Berlin
- Pillonetti G, Dinuzzo F, Chen T, De Nicolao G, Ljung L (2014) Kernel methods in system identification, machine learning and function estimation: a survey. *Automatica* 50(3):657–683
- Pintelon R, Schoukens J (2012) *System identification – a frequency domain approach*, 2nd edn. IEEE, New York
- Popper KR (1934) *The logic of scientific discovery*. Basic Books, New York
- Rasmussen CE, Williams CKI (2006) *Gaussian processes for machine learning*. MIT, Cambridge
- Rissanen J (1978) Modelling by shortest data description. *Automatica* 14:465–471
- Söderström T (2018) *Errors-in-variables methods identification in system identification*. Springer, New York
- Söderström T, Stoica P (1983) *Instrumental variable methods for system identification*. Springer, New York
- Söderström T, Stoica P (1989) *System identification*. Prentice-Hall, London
- Stone M (1977) Asymptotics for and against cross-validation. *Biometrika* 64(1):29–35
- Tikhonov AN, Arsenin VY (1977) *Solutions of Ill-posed problems*. Winston/Wiley, Washington, DC
- Young PC (2011) *Recursive estimation and time-series analysis*, 2nd edn. Springer, Berlin
- Zadeh LA (1956) On the identification problem. *IRE Trans Circuit Theory* 3:277–281

Tactical Missile Autopilots

Curtis P. Mracek

Raytheon Missile Systems, Waltham, MA, USA

Keywords

Classical control · Control surfaces · Pitch · Proportional and integral control · Roll channel · Tactical missile · Yaw channels

Abstract

Tactical missile autopilots are part of the wider guidance navigation and control missile system whose goal is to achieve a successful intercept. The missile autopilot task is to turn guidance commands into fin deflection and is generally divided into two lateral direction (pitch and yaw) controllers and the roll orientation or roll rate controller. These three “channel control” outputs are then mixed to produce fin commands. The controllers can be composed of different architectures but most lateral autopilots use a three loop structure with acceleration and angular rate feedback. The roll controller is usually either a proportional integral (PI) or proportional integral derivative (PID) controller. The controllers are designed using gain scheduling for large flight envelope applications and have nonlinear elements to shape the time response. Integrator reset logic, to deal with control surface saturation, is also an integral part of tactical missile autopilots.

Introduction

The purpose of a tactical missile is to intercept targets, and since tactical missile autopilots are part of the larger tactical missile system, they must contribute to that goal. The process by which a missile executes an intercept is by first sensing the target. The target information is then used to generate guidance commands. The guidance commands are determined such that if followed with precision the missile will intercept the target. The problem is to follow with precision. This is where the autopilot comes in. The missile autopilot receives guidance commands and produces control deflections to move the missile in a manner consistent with completing the intercept. There are many control challenges unique to tactical missiles, namely closing velocities can be very high and targets very small and very maneuverable. Usually the guidance commands are acceleration commands though other quantities are sometimes used. For this discussion, acceleration commands will be the autopilot commands. Once the acceleration

commands or demands (as some in the guidance community call the autopilot inputs) are presented to the autopilot, the autopilot's only concern is to produce the desired command as fast as possible with some level of robustness. The key performance metric is the time of response. The response time is a key factor that drives the miss distance, and thus the probability of a successful intercept. Another metric, though less important than the response time, is the available maneuverability. As mentioned, the autopilot achieves the desired acceleration through moving the control surfaces. Usually, the control surfaces are aerodynamic and either positioned in front of (canard control) or in back of (tail control) the center of gravity. Both tail and canard control surfaces will be called fins for the purposes of this analysis. Some recent missile designs have significantly altered the autopilot design problem by using both canards and tails or other effectors like reaction jets.

The tactical missile autopilot control problem is therefore to produce accelerations by moving the fins in a controlled manner such that the response is as fast as possible while remaining under control under various flight conditions and in the presence of uncertainties (being robust). Tactical missiles autopilots are a classic control challenge in that there is a direct trade between performance and robustness. The tactical autopilot tends to lean toward the performance instead of the robustness because, as the continuing argument goes, "what good is the missile being stable if you miss the target" versus "if the missile is unstable it may never get to the target." So far, relevant analysis has mentioned the controller but mostly ignored robustness. Achieving robustness is done through the use of feedback, and in tactical missiles, inertial sensing devices are used to provide this critical information. These tactical sensors are currently packaged as a complete inertial sensor suite. This suite usually consists of three orthogonal linear accelerometers and three angular rate gyros. One reason guidance commands are the linear acceleration is that the sensing device directly measures this desired quantity.

There are two noticeable differences between tactical missile control and other aerodynamic

control applications. The first is that the dynamics and controls are divided into three distinct channels with each channel nearly independent of the other two. These are the lateral (pitch and yaw) channels and the axial (roll) channel. The pitch and yaw designs are usually very similar, if not identical, and the roll channel is separate. The second is that the controllers (fins) are intertwined. That is, there are no predominately pitch, yaw, or roll controllers, such as there are on airplanes. At least two and sometimes four fins are used in a single channel. This mixing of controls is through what is called a fin mix. This fin mixing occurs in the software (used to be hardware in analog controllers) after the autopilot and prior to the signals being sent to the individual fins.

Historically tactical missile autopilot development has consisted of both a design phase and an analysis phase. This distinction is due to the controller being designed on a subset of the operating envelope. That design is then evaluated at many more conditions to determine if the design works well enough everywhere to be deployed. In both the design and analysis phases, models are used to establish performance and robustness. Linear planar, linear coupled, and nonlinear models are used. The linear models are usually restricted to the early design phases and the frequency response determination of the system. The nonlinear models are used for time domain analysis.

The remainder of the chapter is organized by examining the linear planar pitch and yaw autopilots, followed by roll control. The concept of combining controllers is then presented. This is followed by a short section on other considerations, such as coupled designs and nonlinear elements.

Pitch and Yaw Control

For tactical missile autopilot development, the equations of motion are usually derived in a body-fixed system with the two lateral velocities replaced by the local angle of attack (α) and sideslip angle (β). It should be noted that the

sideslip is not defined as the aircraft sideslip but instead as the equivalent of the angle of attack in the horizontal plane. This is because of the symmetry that is found in missiles that does not exist in aircraft. The nonlinear equations of motion can be found in Blakelock (1991). For tactical missiles there is usually no axial acceleration control, and thus the total velocity equation is uncontrollable and removed from both the design and analysis. For the coupled equations of motion of the system, there are five equations of motion and three control inputs. For a planar view of the problem, the pitch and yaw channels in a tactical missile autopilot are usually separated, and with the appropriate sign changes in the feedback signals can use the same gains. These channels use an inertial measuring device for feedback. Usually these sensors come in a package with three accelerometers and the gyros. The outputs of these devices used in the pitch are the linear acceleration perpendicular to the axial direction and the angular rate of the missile about the other perpendicular axis. That is, the z linear acceleration (A_{zm}) and the y angular rate (q_m). Using the other four sensors would cause coupling between pitch and yaw and roll, and thus this sensor information is usually ignored in the pitch channel. They are available and used in select cases where there is strong aerodynamic coupling, in which case these cross channels can be used to decouple the system. Without getting into the actual definitions of all the variables and the numerical values (see Mracek and Ridgely 2005a for full details), the state space linearized equations of motion for the pitch plane are:

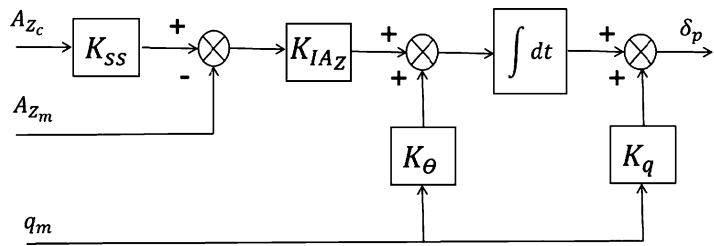
$$\begin{aligned} A &= \begin{bmatrix} 1/V_{mo} \left[\frac{Z_{\alpha o}}{m} - A_{Xo} \right] & 1 \\ M_{\alpha o}/I_{YY} & 0 \end{bmatrix} \\ B &= \begin{bmatrix} Z_{\delta po}/mV_{mo} \\ M_{\delta po}/I_{YY} \end{bmatrix} \\ C &= \begin{bmatrix} Z_{\alpha o}/mg - M_{\alpha o}\bar{x}/gI_{YY} & 0 \\ 0 & 1 \end{bmatrix} \\ D &= \begin{bmatrix} Z_{\delta po}/mg - M_{\delta po}\bar{x}/gI_{YY} \\ 0 \end{bmatrix} \end{aligned}$$

where

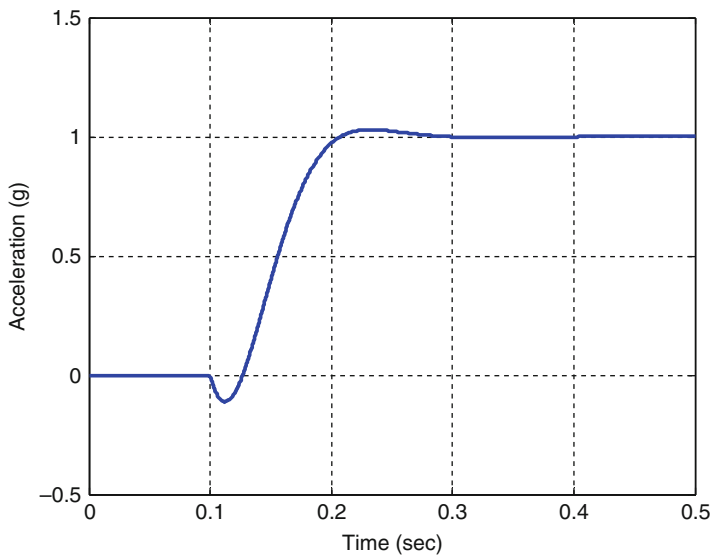
$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad x = \begin{bmatrix} \alpha \\ q \end{bmatrix} \quad u = \delta_p \quad y = \begin{bmatrix} A_{zm} \\ q_m \end{bmatrix}$$

Thus in the most reduced form, the tactical missile autopilot equations of motion reduce to two equations with two variables and one control. This is a very simple control problem. Since full state feedback can be used to provide an “optimal” control solution, only two feedback signals are needed for the above state space problem. For a tail controlled missile, the two state control leads ultimately to increasing missile acceleration in the wrong direction, as faster and faster designs are realized. This is because the system is, in controls language, “non-minimum phase.” That is, tail controlled missiles move in the wrong direction before they move in the commanded direction. Canard controlled missiles do not suffer this problem. See Mracek (2005) and Gutman (2003) on the relative merits of canards and tails. If the control rate is used as the input instead of the control position, there would be three states in the basic plant used in the analysis, and three signals would need to be included. Now if we consider the fin position as a variable for feedback with the accelerometer and gyro feedbacks, there are a number of different combinations of sensor feedback signals that can be used to solve the three state problem. There are, in fact, nine possible topologies, two of which are consistently robust, with one topology showing excellent robustness characteristic. For a complete comparison see Mracek and Ridgely (2005b). This topology is shown in Fig. 1. Notice that there is an integral in the formulation. This limits the actual command rate from being infinite when a step command is input to the system. Without the command going through the integrator the controller would see the step, and, since the force instantly produces an acceleration, the feedback would jump (given no actuation delay). A typical acceleration response to a step acceleration command and control deflection rate needed to produce the response is presented in Figs. 2 and 3, respectively.

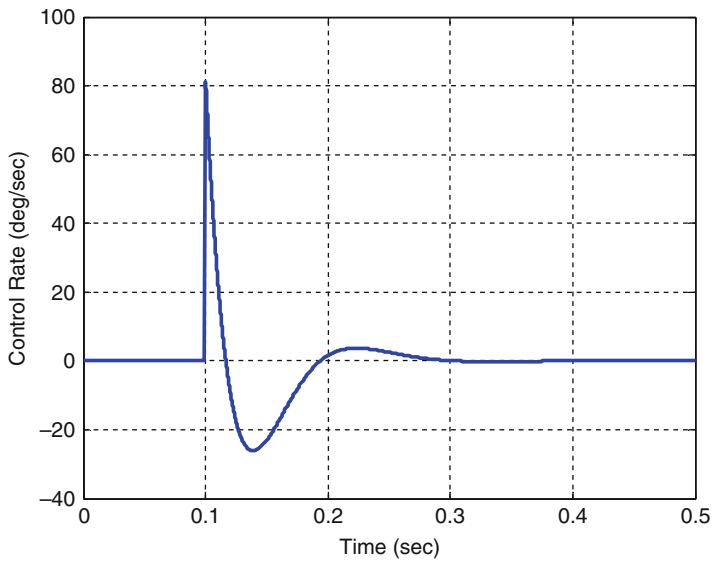
Tactical Missile Autopilots, Fig. 1 Three loop pitch topology



Tactical Missile Autopilots, Fig. 2 Acceleration response to a step input



Tactical Missile Autopilots, Fig. 3 Control rate usage



The feedback control law is:

$$\delta_p = K_{IA_z} K_{ss} \int A_{Z_c} dt - K_{IA_z} \int A_{Z_m} dt + K_\theta \int q_m dt + K_q q_m$$

Clearly, other components within the autopilot loop have to be considered. The control actuation system (CAS) and inertial measurement device characteristics need to be included in the design and synthesis of the autopilot. To this end, the gains are usually selected to provide the best performance (in the time domain) based on constraints. The above optimal control solution provides guaranteed margins, but when the additional components are included in the analysis the margins are an important constraint in the ultimate performance that can be achieved. Like most control problems, the constraints are both time and frequency dependent. Because of the emphasis on performance, some of the margin constraints must be examined closely. For a more detailed treatment of the three loop autopilot see Zarchan (2002).

Roll Control

Thus far we have discussed the two lateral channels of the missile. That is because those two are the channels that directly affect the miss distance. The third channel does not directly influence the miss but it still is usually controlled. The roll channel is usually the fastest of the channels for a tactical missile. Historically, the three channels were decoupled by moving the roll “out of the way” of the other channels by designing to a

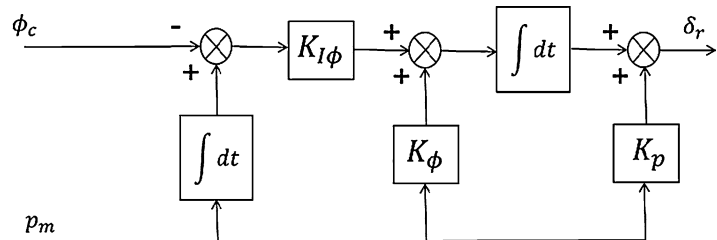
higher bandwidth than the pitch and yaw channels. Because of the need for squeezing performance this practice is not always employed. The cost of not increasing the bandwidth of the roll beyond the pitch and yaw is that the interdependence of the channels needs more scrutiny. The roll channel has only one sensor element, the roll rate sensor. This measures the angular rate of the body about its central axis relative to the inertial frame. The objective, and thus the autopilot, can differ depending on the missile application. Mostly the objective would be one of the following: maintain zero roll rate, zero integral of roll rate, or some preferred Euler angle orientation. The last two can be accomplished with the same autopilot architecture, with exception handling for the Euler roll control based on the singularity in the Euler roll angle at $\pm 90^\circ$ pitch orientations.

Since there is only one sensor and one control, this channel is a classical SISO system and can be controlled with a proportional derivative (PD) or proportional integral derivative (PID) controller. The three loop topologies with integral roll rate reference are presented in Fig. 4.

Gain Scheduling

Early generation missiles had analog autopilots and some were marvels of ingenuity. Now digital control is used almost exclusively. As can be readily seen from the above discussion, the autopilots performance is largely dictated by gains within a given topology. Unlike with early autopilots, with digital control the gains can be set precisely and can vary greatly as needed over a wide range of flight conditions. Rarely can a single set of gains be found

Tactical Missile Autopilots, Fig. 4 Three loop roll topology



that provides adequate performance under all conditions. Thus, the autopilot design process is to design for a large number of flight conditions and then join the individual designs into a coherent whole. Typically, the conditions for the individual designs would be something like Mach number, altitude, and center of mass location. Once the individual gains are designed, they are joined together through an algorithm. Most likely they are “looked up” as a continuous function of the independent variables through some sort of interpolation. This gain changing philosophy is called gain scheduling. There have been some successful attempts for full envelope autopilot design. Dynamic inversion or model-based approaches have also been developed, most notably JDAM (Wise et al. 2005) where the autopilot was borrowed and adjusted on the fly from a sister design. The argument for the validity of this approach is that the flight conditions are not changing rapidly so they can be ignored. Of course the synthesis of the design needs to include examination of “off break point” conditions (flight conditions within the flight envelope that were not considered in the design process) to ensure compliance with stability requirements. History has shown that tactical missile autopilot gains tend to be somewhat power functions of dynamic pressure based on the design constraints.

Other Considerations

The selection of gains using planar linear models and then scheduling them is not the complete autopilot design exercise. There are other challenges that must be considered. First, the plant equations are coupled through both the kinematic equations and the aerodynamics of the problem. There are two predominant ways to attack this problem in autopilot design. The first is as discussed earlier in which the system is made to be as decoupled as possible, create gains for the decoupled system and analyze them in the

coupled system. The second is to use feedback to create a more integrated design through cross coupling terms.

Besides the coupling there can be other problems. The design problem is hard enough as described above, but we have learned over the years that the models developed earlier have neglected certain aspects of the missile that can lead to problems. One aspect is that missiles can be very flexible, and since an inertial sensor is being used for feedback, the flexible characteristics can drive the missile unstable. The flexible characteristics were examined by Nesline and Nesline (1985). In that paper, the flexible model is presented and a technique for ignoring the first mode is discussed. (It should be noted that the model presented in the appendix has some “typos” and should be used with caution.)

Another aspect is the consideration of nonlinear elements of the autopilot. These could include integrator reset logic, command error limits, and acceleration limits. The three loop autopilot has an integrator and the fact that integrators “wind up” when the output is saturated. For tactical missiles this saturation could be caused by position or rate limits. The integrator should be reset to account for these conditions so that the missile responds quicker when the system is no longer in a saturated condition. The command error limits can be used to modify the response characteristics to achieve a more consistent response. Finally, acceleration limits are used to limit the input into the system such that the guidance commands do not put the missile into a position from which it cannot maintain controlled flight. Generating acceleration limits is a complex topic itself.

Summary and Future Directions

Tactical missile autopilots are generally designed by separating the problem into two independent lateral controls (pitch and yaw), with a third control governing the roll attitude. A good autopi-

lot design produces a balance between performance and robustness and incorporates nonlinear elements and integrator resets. The design process take into account robustness throughout the flight envelope and structural elements.

From a controls standpoint, the future direction of tactical missile autopilot development is in nonlinear, adaptive, and fault tolerant control. Adaptive control is useful not only because it provides a more predictable flight response but also because of the potential in reducing or maybe even eliminating development time.

Cross-References

- [Aircraft Flight Control](#)
- [PID Control](#)

Bibliography

- Blakelock JH (1991) Automatic control of aircraft and missiles, 2nd edn. Wiley, New York
- Gutman S (2003) Superiority of canards in homing missiles. *IEEE Trans Aerosp Electron Syst* 39(3): 740–746
- Mracek CP (2005) A miss distance study for homing missiles: tail vs canard control. In: Proceedings of the AIAA guidance navigation and control conference, Minneapolis, Aug 2006
- Mracek CP, Ridgely DB (2005a) Missile longitudinal autopilots: connections between optimal control and classical topologies. In: Proceedings of the AIAA GNC conference, San Francisco, Aug 2005
- Mracek CP, Ridgely DB (2005b) Missile longitudinal autopilots: comparison of multiple three loop topologies. In: Proceedings of the AIAA guidance navigation and control conference, San Francisco, Aug 2005
- Nesline FW, Nesline ML (1985) Phase vs gain stabilization of structural feedback oscillations in homing missiles. In: Proceedings of the American control conference, 1985
- Wise KA, Lavretsky E, Zimmerman J, Francis JH Jr, Dixon D, Whitehead B (2005) Adaptive flight control of a sensor guided munition. In: Proceedings of the AIAA guidance navigation and control conference, San Francisco, Aug 2005
- Zarchan P (2002) Tactical and strategic missile guidance. *Progress in Astronautics and Aeronautics*, vol 199, 4th edn. AIAA, Reston

Time-Scale Separation in Power System Swing Dynamics: Singular Perturbations and Coherency

Joe H. Chow

Department of Electrical and Computer Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract

Large power systems often exhibit slow and fast electromechanical oscillations between interconnected synchronous machines. The slow interarea oscillations involve coherent groups of machines swinging together. This coherency phenomenon can be attributed to the coherent areas of machines being weakly coupled, either because of higher impedance transmission lines, heavily loaded transmission lines, or fewer connections between the coherent areas compared to the connections within a coherent area. Singular perturbations can be used to display the time-scale separation of the slow interarea modes and the faster local modes.

Keywords

Model reduction · Power system oscillations · Singular perturbations · Two-time-scale systems

Interarea Mode Oscillation in a Power System

A large power system consists of interconnected synchronous machines supplying power to loads via transmission lines. As a dynamical system, it can be considered as the rotating inertias of the synchronous machines interacting electrically through the impedances of the transmission system. During a disturbance, such as a lightning strike on a transmission line, the

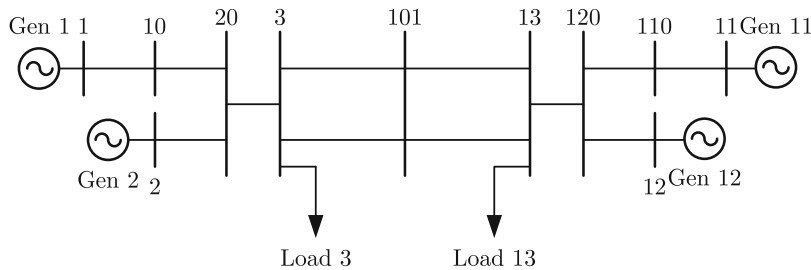
rotating inertias will oscillate against each other. The frequency and extent of these oscillations may vary: the local modes of frequencies 1–2.5 Hz originate from the interactions of a few close-by machines, and the interarea modes of frequencies 0.2–0.8 Hz involve groups of machines swinging against other groups. Coherency is this phenomenon of groups of machines swinging together against other groups of machines during disturbances.

Coherency can be illustrated in the simple power system shown in Fig. 1 (Rogers 2000). The system consists of two areas: Generators 1 and 2 in Area 1 and Generators 11 and 12 in Area

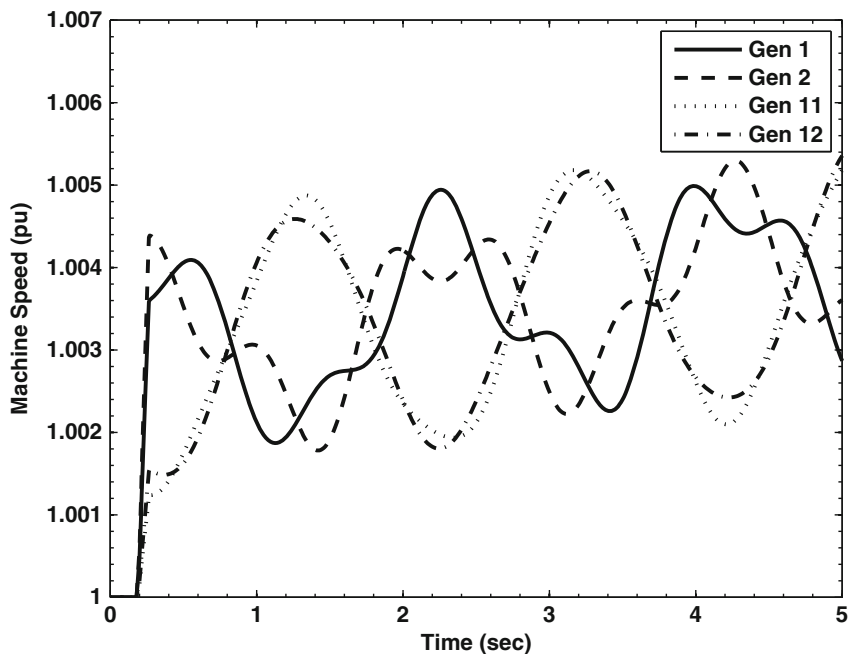
2. For a disturbance in Area 1, Fig. 2 shows the response of the machine speeds. The interarea mode consists of Generators 1 and 2 swinging coherently against Generators 11 and 12. The difference between the responses of Generators 1 and 2 is due to the local mode in Area 1, which is excited by the disturbance.

Coherency Analysis

Coherency with respect to the slow interarea modes, also known as slow coherency, is an inherent property of many power systems. Tradi-



Time-Scale Separation in Power System Swing Dynamics: Singular Perturbations and Coherency, Fig. 1 Two-area, four-machine system example



Time-Scale Separation in Power System Swing Dynamics: Singular Perturbations and Coherency, Fig. 2 Machine speed response of the two-area, four-machine system

tional power systems consist of operating regions dictated by physical or administrative constraints with relatively strong connections within an operating region. These control regions are also interconnected with tielines to share base-load and seasonal power resources as well as to rely on each other for reserves. Thus a practical interconnected power system will, by design, necessarily have strong connections within each operating region and weaker connections between the regions. Due to the time-scale separation of the slow interarea modes and the faster local modes, the coherency phenomenon can be analyzed using singular perturbations method provided a suitable small parameter can be identified.

For a simplified coherency analysis, the linearized second-order model of an N -machine power system

$$M \frac{d^2 \Delta \delta}{dt^2} = K \Delta \delta \quad (1)$$

can be used. In (1), δ is the N -dimensional vector of individual machine rotor angles δ_i , $i = 1, \dots, N$, Δ denotes small perturbations, M is the diagonal matrix of machine rotational inertias m_i , $i = 1, \dots, N$, and the *connection matrix* K consists of the linearized synchronizing coefficients K_{ij} between machines i and j , denoting the restoring force between the two machines.

An important property of K is

$$K_{ii} = - \sum_{j=1, j \neq i}^N K_{ij} \quad (2)$$

that is, the sum of each row of K is zero. Thus K has a zero eigenvalue, which is known as the system mode. This mode arises due to the lack of a reference, as only the relative angles between the machines are important. It can be eliminated when one of the machines is chosen as the reference.

Suppose the N -machine system has r areas of coherent machines, whose internal connections within the areas are stronger than the external connections between the areas. The weak connection strength is denoted by a small parameter ε ,

which can be the ratio of the relative stiffness of the internal transmission lines versus the external transmission lines, or the ratio of the smaller number of external connections versus the larger number of internal connections, or both. Thus the connection matrix of linearized synchronizing coefficients can be rewritten as

$$K = K^I + \varepsilon K^E \quad (3)$$

where K^I is the matrix of internal connections and K^E is the matrix of external connections scaled by ε . If the machine angles in each coherent area are arranged in consecutive order in the vector δ , then K^I is block diagonal with r zero eigenvalues, that is, one system mode per area.

Singular Perturbation Analysis

To exhibit the time scales in (1) and (3), a transformation to obtain the slow variables and the fast variables is introduced. The slow motion is obtained by defining for each area, an inertia-weighted *aggregate variable*

$$y^\alpha = \sum_{i=1}^{n_\alpha} m_i^\alpha \Delta \delta_i^\alpha / m^\alpha, \quad m^\alpha = \sum_{i=1}^{n_\alpha} m_i^\alpha, \quad \alpha = 1, 2, \dots, r \quad (4)$$

where n_α is the number of machines in area α , m_i^α is the inertia of machine i in area α , and m^α is the aggregate inertia of area α . For the fast dynamics, we select in each area a reference machine, say the first machine, and define the motions of the other machines in the same area relative to this reference machine by the *local variables*

$$z_{i-1}^\alpha = \Delta \delta_i^\alpha - \Delta \delta_1^\alpha, \quad i = 2, 3, \dots, n_\alpha, \quad \alpha = 1, 2, \dots, r \quad (5)$$

The transformations (4) and (5) can be combined to form

$$\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} M_a^{-1} U^T M \\ G \end{bmatrix} \Delta \delta \quad (6)$$

where

$$U = \text{blockdiag}(u_1, u_2, \dots, u_r) \quad (7)$$

is the grouping matrix with $n_\alpha \times 1$ column vectors

$$u_\alpha = [1 \ 1 \ \dots \ 1]^T, \quad \alpha = 1, 2, \dots, r \quad (8)$$

$$M_a = \text{diag}(m^1, m^2, \dots, m^r) = U^T M U \quad (9)$$

and

$$G = \text{blockdiag}(G_1, G_2, \dots, G_r) \quad (10)$$

with G_α being the $(n_\alpha - 1) \times n_\alpha$ matrix

$$G_\alpha = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & \dots & 1 \end{bmatrix} \quad (11)$$

The inverse of this transformation is explicitly known

$$\Delta\delta = [U \ G^T(GG^T)^{-1}] \begin{bmatrix} y \\ z \end{bmatrix} \quad (12)$$

Applying the transformation (6) to the model (1) and (3), the electromechanical model becomes

$$\begin{aligned} M_a \ddot{y} &= \varepsilon K_a y + \varepsilon K_{ad} z \\ M_d \ddot{z} &= \varepsilon K_{da} y + (K_d + \varepsilon K_{dd}) z \end{aligned} \quad (13)$$

where

$$\begin{aligned} M_d &= (GM^{-1}G^T)^{-1}, \quad K_a = U^T K^E U \\ K_{da} &= U^T K^E M^{-1} G^T M_d, \\ K_{da} &= M_d G M^{-1} K^E U \\ K_d &= M_d G M^{-1} K^I M^{-1} G^T M_d, \\ K_{dd} &= M_d G M^{-1} K^E M^{-1} G^T M_d \end{aligned} \quad (14)$$

Note that K_a , K_{ad} , and K_{da} are independent of the internal connection matrix K^I because $K^I U = 0$. Furthermore, K_a is negative semi-definite and K_{dd} is negative definite. System

(13) is in the *standard singularly perturbed form* (Kokotović et al. 1986) showing that y is the slow variable and z is the fast variable. Thus ε is both the weak connection parameter and the singular perturbation parameter, giving rise to slow coherency.

The dynamics of the singularly perturbed system (13) are approximated by the interarea modes $\pm j\sqrt{-\varepsilon\lambda(M^{-1}K_a)}$ and the local modes $\pm j\sqrt{-\lambda(M_d^{-1}K_{dd})}$, where λ denotes eigenvalues.

Identifying Coherent Areas

Several methods can be used to identify coherent areas, including the following:

1. Time simulation method (Podmore 1978): This method simulates the dynamic responses to a selected set of disturbances and groups the machines having similar time responses as coherent areas. For a faster simulation, a linearized power system model can be used.
2. Eigenvector method (Chow et al. 1982): This method computes the slow eigenvalues of the matrix $M^{-1}K$ and identifies machines with similar row vectors of the slow eigenvector matrix as coherent machines.
3. Weak link methods (Zaborszky et al. 1982; Nath et al. 1985): These methods search through the transmission line impedances to find the weak links between the areas.

Applications

The applications of the coherency concept include:

1. Dynamic model reduction (deMello et al. 1975): The synchronous machines in a coherent area can be aggregated into a single equivalent machine, thus reducing the system size. Model reduction programs capable of handling upwards of 30,000 buses are available (Morison and Wang 2013).
2. Interarea mode analysis and damping control design (Larsen et al. 1995): Damping

of interarea modes is an operational concern for systems with heavily loaded long-distance transmission lines. The slow coherency concept contributes to the development of damping controller design.

3. Islanding as a defense mechanism (You et al. 2004): During system disturbances causing severe power flow interruption, the last resort may be to separate the systems into viable islands, avoiding a total system blackout. Coherent areas tend to be natural choices of islands.

In addition to power system analysis, the coherency concept and methods can potentially be applied to dynamic systems with a system mode (eigenvalue equal to 0 for a continuous-time model and eigenvalue equal to 1 for a discrete-time model). An example is the PageRank computation in (Ishii et al. 2012).

Cross-References

- [Consensus of Complex Multi-agent Systems](#)
- [Lyapunov Methods in Power System Stability](#)
- [Markov Chains and Ranking Problems in Web Search](#)
- [Model Order Reduction: Techniques and Tools](#)
- [Small Signal Stability in Electric Power Systems](#)

Recommended Reading

An early investigation of coherency was reported in Podmore and Germond (1977). A recent compilation of power system coherency, model reduction, and interarea oscillation results can be found in Chow (2013).

Bibliography

- Chow JH (ed) (2013) Power system coherency and model reduction. Springer, New York
- Chow JH, Peponides G, Kokotović PV, Avramović B, Winkelman JR (1982) Time-scale modeling of

- dynamic networks with applications to power systems. Springer, New York
- deMello RW, Podmore R, Stanton KN (1975) Coherency-based dynamic equivalents: applications in transient stability studies. 1975 PICA conference proceedings, pp 23–31
- Ishii H, Tempo R, Bai E-W (2012) A web aggregation approach for distributed randomized PageRank algorithms. *IEEE Trans Autom Control* 57:2703–2717
- Kokotović PV, Khalil H, O'Reilly J (1986) Singular perturbation methods in control: analysis and design. Academic, London
- Larsen EV, Sanchez-Gasca JJ, Chow JH (1995) Concepts for design of FACTS controllers to damp power swings. *IEEE Trans Power Syst* 10:948–956
- Morison K, Wang L (2013) Reduction of large power system models: a case study. In: Chow JH (ed) Power system coherency and model reduction. Springer, New York, Chapter 7
- Nath R, Lamba SS, Rao KSP (1985) Coherency based system decomposition into study and external areas using weak coupling. *IEEE Trans Power Appar Syst PAS-104:1443–1449*
- Podmore R, Germond A (1977) Dynamic equivalents for transient stability studies. EPRI Report RP-765
- Podmore R (1978) Identification of coherent generators for dynamic equivalents. *IEEE Trans Power Appar Syst PAS-97(4):1344–1354*
- Rogers G (2000) Power system oscillations. Kluwer Academic, Dordrecht
- You H, Vittal V, Wang X (2004) Slow coherency-based islanding. *IEEE Trans Power Syst* 19: 483–491
- Zaborszky J, Whang K-W, Huang GM, Chiang L-J, Lin S-Y (1982) A clustered dynamical model for a class of linear autonomous systems using simple enumerative sorting. *IEEE Trans Circuit Syst CAS-29:747–758*

Tracking and Regulation in Linear Systems

Alessandro Astolfi

Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma Tor Vergata, Rome, Italy

Abstract

Tracking and regulation refers to the ability of a control system to track/reject a given family of reference/disturbance signals modelled as

solutions of a differential/difference equation. The problem can be posed as a stabilization problem with a constraint on the steady-state response of the system. For linear, time-invariant, systems the problem can be solved provided a system of linear matrix equations admits a solution. Properties of this system of equations are discussed, together with a general property of all controllers achieving tracking and regulation: the so-called internal model principle.

Keywords

Linear systems · Tracking · Regulation · Internal model principle

Introduction

Consider a linear system affected by disturbances and such that its output is required to asymptotically track a certain, prespecified, reference signal. In what follows, we discuss and solve this control problem: known as the *tracking and regulation problem*.

Consider a linear control system described by equations of the form

$$\begin{aligned}\sigma x &= Ax + Bu + Pd, \\ e &= Cx + Qd,\end{aligned}\tag{1}$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $e(t) \in \mathbb{R}^p$, $d(t) \in \mathbb{R}^r$, $t \in T \subset \mathbb{R}$, and A , B , P , C , and Q constant matrices. In Eq. (1) $\sigma x = \sigma x(t)$ stands for $\dot{x}(t)$, if the system is continuous-time, and for $x(t+1)$, if the system is discrete-time. Since the system is time-invariant, it is assumed, without loss of generality, that all signals are defined for $t \geq 0$, that is, if the system is continuous-time, then $T = \mathbb{R}^+$, i.e., the set of nonnegative real numbers, whereas if the system is discrete-time, then $T = \mathbb{Z}^+$, i.e., the set of nonnegative integers. For ease of notation the argument “ t ” is dropped whenever this does not cause confusion.

The signal d , denoted exogenous signal, is in general composed of two components: the former models a set of disturbances acting on the system

to be controlled and the latter a set of reference signals. We assume that the exogenous signal is generated by a linear system, denoted *exosystem*, described by the equation

$$\sigma d = Sd,\tag{2}$$

with S a matrix with constant entries. Under this assumption it is possible to generate, for example, constant or polynomial references/disturbances and sinusoidal references/disturbances with any given frequency.

The variable e , denoted tracking error, is a measure of the error between the ideal behavior of the system and the actual behavior. Ideally, the variable e should be regulated to zero, i.e., should converge asymptotically to zero, despite the presence of the disturbances. If this happens, the disturbances are not affecting the asymptotic behavior of the system, and the output Cx is asymptotically tracking the reference signal $-Qd$. In general the tracking error does not naturally converge to zero, hence it is necessary to determine an input signal u which *drives* it to zero. The simplest possible way to construct such an input signal is to assume that it is generated via static feedback of the state x of the system to be controlled and of the state d of the exosystem, i.e.,

$$u = Kx + Ld.\tag{3}$$

In practice it is unrealistic to assume that both x and d be available for feedback, hence it may be more natural to assume that the input signal u be generated via dynamic feedback of the error signal only, i.e., be generated by the system

$$\begin{aligned}\sigma \chi &= F\chi + Ge, \\ u &= H\chi,\end{aligned}\tag{4}$$

with $\chi(t) \in \mathbb{R}^v$, for some $v > 0$, and F , G , and H matrices with constant entries.

Using the above definitions, it is possible to formally pose the regulator problem as follows.

Definition 1 (Full information regulator problem) Consider the system (1), driven by the exosystem (2) and interconnected with the con-

troller (3). The *full information regulator problem* is the problem of determining the matrices K and L of the controller such that ((S) stands for stability and (R) for regulation.)

- (S) the system $\sigma x = (A + BK)x$ is asymptotically stable;
 (R) all trajectories of the system

$$\begin{aligned}\sigma d &= Sd, \\ \sigma x &= (A + BK)x + (BL + P)d, \\ e &= Cx + Qd,\end{aligned}\quad (5)$$

are such that $\lim_{t \rightarrow \infty} e(t) = 0$.

Definition 2 (Error feedback regulator problem) Consider the system (1), driven by the exosystem (2) and interconnected with the controller (4). The *error feedback regulator problem* is the problem of determining the matrices F , G , and H of the controller such that

- (S) the system

$$\begin{aligned}\sigma x &= Ax + BH\chi, \\ \sigma \chi &= F\chi + GCx,\end{aligned}$$

is asymptotically stable;

- (R) all trajectories of the system

$$\begin{aligned}\sigma d &= Sd, \\ \sigma x &= Ax + BH\chi + Pd, \\ \sigma \chi &= F\chi + GCx + GQd, \\ e &= Cx + Qd,\end{aligned}\quad (6)$$

are such that $\lim_{t \rightarrow \infty} e(t) = 0$.

The Full Information Regulator Problem

Consider the full information regulator problem and assume the following.

Assumption 1 The matrix S of the exosystem has all eigenvalues with nonnegative real part, in the case of continuous-time systems, or with modulo not smaller than one, in the case of discrete-time systems.

Assumption 2 The system (1) with $d = 0$ is reachable.

Assumption 1 implies that there are no initial conditions $d(0)$ such that the signal d converges (asymptotically) to zero. This assumption is not restrictive. In fact, disturbances converging to zero do not have any effect on the asymptotic behavior of the system, and references which converge to zero can be tracked simply by driving the state of the system to zero, i.e., by stabilizing the system. Assumption 2 implies that it is possible to arbitrarily assign the eigenvalues of the matrix $A + BK$ by a proper selection of K . Note that, in practice, this assumption can be replaced by the weaker assumption that the system (1) with $d = 0$ is stabilizable.

We now present a preliminary result which is instrumental to derive a solution to the full information regulator problem.

Lemma 1 Consider the full information regulator problem. Suppose Assumption 1 holds. Suppose, in addition, that there exists a matrix K such that condition (S) holds.

Then condition (R) holds if and only if there exists a matrix $\Pi \in \mathbb{R}^{n \times r}$ such that the equations

$$\begin{aligned}\Pi S &= (A + BK)\Pi + (P + BL), \\ 0 &= C\Pi + Q,\end{aligned}\quad (7)$$

hold.

Proof. Consider the system (5) and the coordinates transformation

$$\begin{aligned}\hat{d} &= d, \\ \hat{x} &= x - \Pi d,\end{aligned}$$

where Π is the solution of the equation

$$\Pi S = (A + BK)\Pi + (P + BL). \quad (8)$$

Note that, by condition (S) and Assumption 1, there is a unique matrix Π which solves this equation. In the new coordinates $\hat{x}$ and $\hat{d}$ the system is described by the equations

$$\begin{aligned}\sigma \hat{d} &= S\hat{d}, \\ \sigma \hat{x} &= (A + BK)\hat{x}, \\ e &= C\hat{x} + (C\Pi + Q)\hat{d}.\end{aligned}$$

By condition (S) $\lim_{t \rightarrow \infty} \hat{x}(t) = 0$, hence condition (R) holds, by Assumption 1, if and only if $C\Pi + Q = 0$. In summary, under the state assumptions, condition (R) holds if and only if there exists a matrix Π such that Eqs. (7) hold. $\square$

The Eq. (8) is a so-called Sylvester equation. The Sylvester equation is a (matrix) equation of the form $A_1X = XA_2 + A_3$, in the unknown X . This equation has a unique solution, for any A_3 , if and only if the matrices A_1 and A_2 do not have common eigenvalues.

We are now ready to state and prove the result which provides conditions for the solvability of the full information regulator problem.

Theorem 1 *Consider the full information regulator problem. Suppose Assumptions 1 and 2 hold. There exists a full information control law described by the Eq. (3) which solves the full information regulator problem if and only if there exist two matrices Π and Γ such that the equations*

$$\begin{aligned}\Pi S &= A\Pi + B\Gamma + P, \\ 0 &= C\Pi + Q,\end{aligned}\tag{9}$$

hold.

Proof. (Necessity) Suppose there exist two matrices K and L such that conditions (S) and (R) of the full information regulator problem hold. Then, by Lemma 1, there exists a matrix Π such that Eqs. (7) hold. As a result, the matrices Π and $\Gamma = K\Pi + L$ are such that Eqs. (9) hold.

(Sufficiency) The proof of the sufficiency is constructive. Suppose there are two matrices Π and Γ such that Eqs. (9) hold. The full information regulator problem is solved selecting K and

L as follows. The matrix K is any matrix such that the system $\sigma x = (A + BK)x$ is asymptotically stable. By Assumption 2, such a matrix K does exist. The matrix L is selected as $L = \Gamma - K\Pi$. This selection is such that condition (S) of the full information regulator problem holds, hence to complete the proof we have only to show that, with K and L as selected above, the Eqs. (7) hold. This is trivially the case. In fact, replacing L in (7) yields the Eqs. (9), which hold by assumption. As a result, also condition (R) of the full information regulator problem holds, and this completes the proof. $\square$

The proof of Theorem 1 implies that a controller which solves the full information regulator problem is described by the equation

$$u = Kx + (\Gamma - K\Pi)d,$$

with K such that a stability condition holds, and Π and Γ such that Eqs. (9) hold. By Assumption 2, the stability condition can always be satisfied. As a result, the solution of the full information regulator problem relies upon the existence of a solution of Eqs. (9).

The Francis Equations

The Eqs. (9), known as the Francis equations, are linear equations in the unknowns Π and Γ , for which the following statement holds.

Lemma 2 *The Eqs. (9), in the unknowns Π and Γ , are solvable for any P and Q if and only if*

$$\text{rank} \begin{bmatrix} sI - A & B \\ C & 0 \end{bmatrix} = n + p, \tag{10}$$

for all s which are eigenvalues of the matrix S .

For single-input, single-output, systems (i.e., $m = p = 1$), the condition expressed by Lemma 2 has a very simple interpretation. In fact, the complex numbers s such that

$$\text{rank} \begin{bmatrix} sI - A & B \\ C & 0 \end{bmatrix} < n + 1$$

are the zeros of the system

$$\begin{aligned} \sigma x &= Ax + Bu, \\ y &= Cx, \end{aligned}$$

that is the roots of the numerator polynomial of the transfer function $W(s) = C(sI - A)^{-1}B$, i.e., the zeros of $W(s)$. This implies that, for single-input, single-output systems the full information regulator problem is solvable for any P and Q if and only if the eigenvalues of the exosystem are not zeros of the transfer function of the system (1) with input u , output e and $d = 0$.

The Error Feedback Regulator Problem

To provide a solution to the error feedback regulator problem we need to introduce a new assumption.

Assumption 3 The system

$$\begin{aligned} \begin{bmatrix} \sigma x \\ \sigma d \end{bmatrix} &= \begin{bmatrix} A & P \\ 0 & S \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix}, \\ e &= [C \ Q] \begin{bmatrix} x \\ d \end{bmatrix} \end{aligned} \quad (11)$$

is observable.

Note that Assumption 3 implies observability of the system

$$\begin{aligned} \sigma x &= Ax, \\ y &= Cx. \end{aligned} \quad (12)$$

To show this property, note that observability of the system (11) implies that

$$\text{rank} \begin{bmatrix} C & Q \\ CA & \vdots \\ \vdots & \vdots \\ CA^{n+r-1} & \vdots \end{bmatrix} = n + r.$$

This, in turn, implies

$$\text{rank} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n+r-1} \end{bmatrix} = n$$

and, by Cayley-Hamilton Theorem,

$$\text{rank} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} = n,$$

which implies observability of system (12). Similarly to what discussed in the case of Assumption 2, Assumption 3 can be replaced by the weaker assumption that the system (11) is detectable. We are now ready to state and prove the result which provides conditions for the solvability of the error feedback regulator problem.

Theorem 2 Consider the error feedback regulator problem. Suppose Assumptions 1, 2, and 3 hold. There exists an error feedback control law described by the Eq. (4) which solves the error feedback regulator problem if and only if there exist two matrices Π and Γ such that the equations

$$\begin{aligned} \Pi S &= A\Pi + B\Gamma + P, \\ 0 &= C\Pi + Q, \end{aligned} \quad (13)$$

hold.

Theorem 2 can be alternatively stated as follows. Consider the error feedback regulator problem. Suppose Assumptions 1, 2, and 3 hold. Then the error feedback regulator problem is solvable if and only if the full information regulator problem is solvable.

Proof. (Necessity) The proof of the necessity is similar to the proof of the necessity of Theorem 1, hence omitted.

(Sufficiency) The proof of the sufficiency is constructive. Suppose there are two matrices Π and Γ such that Eqs. (13) hold. Then, by Theorem 1 the full information control law $u = Kx + (\Gamma - K\Pi)d$,

with K such that the system $\sigma x = (A + BK)x$ is asymptotically stable, solves the full information regulator problem. This control law is not implementable, because we only measure e . However, by Assumption 3, it is possible to build asymptotically converging estimates ξ and δ of x and d , hence to implement the control law

$$u = K\xi + (\Gamma - K\Pi)\delta. \quad (14)$$

To this end, consider an observer described by the equation

$$\begin{aligned} \begin{bmatrix} \sigma \xi \\ \sigma \delta \end{bmatrix} &= \begin{bmatrix} A & P \\ 0 & S \end{bmatrix} \begin{bmatrix} \xi \\ \delta \end{bmatrix} \\ &+ \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \left([C \ Q] \begin{bmatrix} \xi \\ \delta \end{bmatrix} - e \right) \\ &+ \begin{bmatrix} B \\ 0 \end{bmatrix} [K \ \Gamma - K\Pi] \begin{bmatrix} \xi \\ \delta \end{bmatrix}. \end{aligned}$$

The estimation errors $e_x = x - \xi$ and $e_d = d - \delta$ are such that

$$\begin{bmatrix} \sigma e_x \\ \sigma e_d \end{bmatrix} = \left(\begin{bmatrix} A & P \\ 0 & S \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} [C \ Q] \right) \begin{bmatrix} e_x \\ e_d \end{bmatrix}, \quad (15)$$

hence, by Assumption 3, there exist G_1 and G_2 that assign the eigenvalues of this error system. Note now that the control law (14)

can be rewritten as $u = Kx + (\Gamma - K\Pi)d - (Ke_x + (\Gamma - K\Pi)e_d)$; hence the control law is composed of the full information control law, which solves the regulator problem, and of an additive disturbance which decays exponentially to zero and does not affect the regulation requirement, provided the closed-loop system is asymptotically stable. Therefore, to complete the proof, we need to show that condition (S) holds. In the coordinates x , e_x , and e_d the closed-loop system, with $d = 0$, is described by the equations

$$\begin{bmatrix} \sigma x \\ \sigma e_x \\ \sigma e_d \end{bmatrix} = \begin{bmatrix} A + BK & -BK & -B(\Gamma - K\Pi) \\ 0 & A + G_1C & P + G_1Q \\ 0 & G_2C & S + G_2Q \end{bmatrix} \begin{bmatrix} x \\ e_x \\ e_d \end{bmatrix}. \quad (16)$$

Recall that the matrices G_1 and G_2 have been selected to render system (15) asymptotically stable and that K is such that the system $\sigma x = (A + BK)x$ is asymptotically stable. As a result, system (16) is asymptotically stable. $\square$

The Internal Model Principle

The proof of Theorem 2 implies that a controller which solves the error feedback regulator problem is described by equations of the form (4) with

$$\begin{aligned} \chi &= \begin{bmatrix} \xi \\ \delta \end{bmatrix}, & F &= \begin{bmatrix} A + G_1C + BK & P + G_1Q + B(\Gamma - K\Pi) \\ G_2C & S + G_2Q \end{bmatrix}, \\ G &= \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}, & H &= [K \ \Gamma - K\Pi], \end{aligned} \quad (17)$$

K , G_1 , and G_2 such that a stability condition holds, and Π and Γ such that Eqs. (13) hold. This controller, and in particular the matrix F , possesses a very interesting property.

Proposition 1 (Internal model property) *The matrix F in Eq. (17) is such that*

$$F\Sigma = \Sigma S,$$

for some matrix Σ of rank r . In particular, any eigenvalue of S is also an eigenvalue of F .

Proof. Let

$$\Sigma = \begin{bmatrix} \Pi \\ I \end{bmatrix}$$

and note that $\text{rank } \Sigma = r$, by construction, and that

$$\begin{aligned} F\Sigma &= \begin{bmatrix} A\Pi + G_1C\Pi + BK\Pi + P + G_1Q + B(\Gamma - K\Pi) \\ -G_2C\Pi + S - G_2Q \end{bmatrix} \\ &= \begin{bmatrix} (A\Pi + B\Gamma + P) + G_1(C\Pi + Q) \\ S - G_2(C\Pi + Q) \end{bmatrix} = \begin{bmatrix} \Pi S \\ S \end{bmatrix} = \Sigma S, \end{aligned}$$

hence the first claim. To prove the second claim, let λ be an eigenvalue of S and v the corresponding eigenvector. Then $Sv = \lambda v$, hence

$$F\Sigma v = \Sigma Sv = \lambda \Sigma v,$$

which shows that λ is an eigenvalue of F with eigenvector Σv , and this proves the second claim. $\square$

It is possible to prove that the property highlighted in Proposition 1 is shared by all error feedback control laws which solve the considered regulation problem. This property, which is often referred to as the *internal model principle*, can be interpreted as follows. The control law solving the regulator problem has to *contain* a copy of the exosystem, i.e., it has to be able to generate, when $e = 0$, a copy of the exogenous signal.

Summary and Future Directions

The problem of tracking and regulation for linear systems in the presence of references and/or disturbances generated by a linear exosystem has been solved. It has been shown that the problem is solvable provided a system of linear matrix equations admits a solution. The tracking and regulation problem can be studied and solved for more general classes of systems, including nonlinear systems, distributed parameter systems, and hybrid systems, exploiting the same ideas presented in this entry.

Cross-References

- [Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions](#)
- [Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions](#)
- [Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions](#)
- [Linear Systems: Discrete-Time, Time-Varying, State Variable Descriptions](#)
- [Output Regulation Problems in Hybrid Systems](#)
- [Regulation and Tracking of Nonlinear Systems](#)
- [Tracking Model Predictive Control](#)

Recommended Reading and References

Classical references on the tracking and regulation problem for linear systems are given below.

Bibliography

- Davison EJ (1976) The robust control of a servomechanism problem for linear time invariant multivariable systems. *IEEE Trans Autom Control* 21: 25–34
- Francis BA, Wonham WM (1975) The internal model principle for linear multivariable regulators. *Appl Math Optim* 2:170–194
- Wonham WM (1985) *Linear multivariable control: a geometric approach*, 3rd edn. Springer, New York

Tracking Model Predictive Control

Daniel Limon and Teodoro Alamo
Departamento de Ingeniería de Sistemas y
Automática, Escuela Técnica Superior de
Ingeniería, Universidad de Sevilla, Sevilla, Spain

Keywords

Reference tracking · Loss of recursive feasibility · MPC for tracking · Offset-free control

Introduction

Abstract

The main objective of tracking model predictive control is to stabilize the plant satisfying the constraints and steering the tracking error, that is, the difference between the reference and the output, to zero. In order to predict the expected evolution of the tracking error, some assumptions on the future values of the reference must be considered. Since the reference may differ from expected, the tracking problem is inherently uncertain.

The most common case is to assume that the reference will remain constant along the prediction horizon. These predictive control schemes are typically based on a two-layer control structure in which, provided the value of the reference, first an appropriate target equilibrium point is computed, and then an MPC is designed to regulate the system to this target. Under certain assumptions, closed-loop stability can be guaranteed if the initial state is inside the feasibility region of the MPC. However, if the value of the reference is changed, then there is no guarantee that feasibility and stability properties of the resulting control law hold. Specialized predictive controllers have been designed to deal with this problem. Particularly interesting is the so-called MPC for tracking, which ensures recursive feasibility and asymptotic stability of the setpoint when the value of the reference is changed.

The presence of exogenous disturbances or model mismatches may lead to the controlled system to exhibit offset error. Offset-free control can be achieved by using disturbance models and disturbance estimators together with the tracking predictive controller.

The problem of designing stabilizing model predictive controllers to regulate a system to the origin has been widely studied, and there are well-known solutions for varied cases including linear, nonlinear, and uncertain systems among others (Rawlings et al. 2017).

The objective of tracking MPC is to ensure that the tracking error, which is the difference between a reference or desired output r and the actual output y , tends to zero.

The most common tracking problem is when the reference r is constant. In this case, the controller is required to steer the state of the plant x and the control input applied to the plant u to the equilibrium point (x_r, u_r) where the tracking error $y - r$ is zero. It is also necessary to ensure that x_r is asymptotically stable for the controlled system, i.e., that the state x converges to x_r and that, near x_r , small changes in x cause small changes in the subsequent trajectory. A relatively straightforward solution for this problem exists (Pannocchia 2004).

Setpoint or constant reference tracking is a relevant control problem in the process industry in which the plant is typically designed to operate at an equilibrium point that maximizes the profit of the plant. In this case, the optimal reference is calculated online by a real-time optimizer (RTO) according to an economic criteria. The reference remains constant for a long period of time, until the RTO, which is executed at a very low frequency, calculates a different one. The equilibrium point associated to the given reference must be calculated and provided to the MPC to steer the plant to this target (Muske 1997).

The tracking problem is considerably more difficult when the reference r varies in a way not known a-priori because MPC is naturally suited to deterministic control problems. As such, in

order to deal with varying references, specialized techniques must be developed, a variety of which have been proposed in Pannocchia and Rawlings (2003), Maeder and Morari (2010), Bemporad et al. (1997), Chisci and Zappa (2003), Limon et al. (2008), or Rossiter et al. (1996).

Another tracking problem arises when there exists a mismatch between the model used for prediction in the optimal control problem and the real plant. If the reference is constant and the model mismatch is sufficiently small not to cause loss of asymptotic stability, the state and control will converge to values at which the predicted tracking error, but not the actual tracking error, is zero. The difference between the predicted and actual values of the output y is known as the offset; offset free tracking when the reference is constant may be achieved by the incorporation of a suitable observer to estimate the offset (Pannocchia and Rawlings 2003).

Notation

The set $\mathbb{I}_M$ denotes the set of integers $\{0, 1, \dots, M\}$. I_n denotes the identity matrix in $\mathbb{R}^{n \times n}$. $\mathbf{z}$ denotes a signal (or time sequence) $\mathbf{z} = \{z(0), z(1), \dots\}$, whose cardinality is inferred from the context. A signal that depends on a parameter θ is denoted as $\mathbf{z}(\theta)$, and $z(i; \theta)$ denotes its i -th element. A closed polyhedron $\mathcal{X} \subset \mathbb{R}^n$ is a set that results of the intersection of a finite number of hyperplanes as follows: $\mathcal{X} = \bigcap_i \{x : F_i x \leq f_i\}$, where $F_i \in \mathbb{R}^{1 \times n}$ and $f_i \in \mathbb{R}$.

Problem Statement

In this article, for the sake of simplicity, we consider that the system to be controlled can be modeled as a linear time-invariant system described by a discrete-time state-space linear model

$$x(k+1) = Ax(k) + Bu(k) \quad (1a)$$

$$y(k) = Cx(k), \quad (1b)$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, and $y(k) \in \mathbb{R}^p$ are the state, the manipulable inputs, and the

outputs of the system at time step k , respectively. This model will be used to calculate the predictions in the predictive controller.

The evolution of the plant must be such that the constraint

$$(x(k), u(k)) \in \mathcal{Z} \quad (2)$$

is satisfied for all $k \geq 0$. It is assumed that set $\mathcal{Z}$ is a closed polyhedron with a nonempty interior. Without loss of generality, it is assumed that $(0, 0) \in \mathcal{Z}$.

The main objective of tracking model predictive control is to stabilize the plant fulfilling the constraints and to steer the system output to the reference, that is, steer the tracking error $y - r$ to zero. In order to predict the expected evolution of the tracking error, some assumptions on the future values of the reference $\mathbf{r}$ must be considered. Since the reference may differ from expected, the tracking problem is inherently uncertain.

Thus, assuming that the reference signal is known a priori, $\mathbf{r} = \{r(0), r(1), \dots\}$, the tracking model predictive control law $u(k) = \kappa(x(k), \mathbf{r})$ must be designed to ensure that the resulting controlled system

$$x(k+1) = Ax(k) + B\kappa(x(k), \mathbf{r})$$

$$y(k) = Cx(k)$$

satisfies the constraints, i.e., $(x(k), \kappa(x(k), \mathbf{r})) \in \mathcal{Z}$ for all $k \geq 0$, is stable, and, if it is possible, the controlled output converges to the reference, that is,

$$\lim_{k \rightarrow \infty} \|y(k) - r(k)\| = 0.$$

It is assumed that the system is stabilizable and that the outputs are linearly independent. It is also assumed that the state is measured and available at each sample time.

Tracking MPC for a Constant Reference

The most simple tracking problem is to consider that the reference signal is a constant signal in the

future equal to the actual value of the reference, i.e., $r(k) = r$. This control problem is known as constant reference or setpoint tracking, and it is very common in the process industry, for instance, where processes are typically designed to operate at certain equilibrium point.

Determining the MPC Setpoint

Corresponding to each value of the reference, r , a MPC setpoint (x_r, u_r) must be calculated. This is an equilibrium point of the prediction model

$$x_r = Ax_r + Bu_r, \quad (3)$$

such that it guarantees a null tracking error

$$y_r = Cx_r = r. \quad (4)$$

Conditions for the existence of a setpoint possessing the above properties are given in Rawlings et al. (2017, Lemma 1.14).

The MPC setpoint must also fulfill the constraints (2)

$$(x_r, u_r) \in \mathcal{Z}.$$

In practice, the condition $(x_r, u_r) \in \mathcal{Z}$ is replaced by $(x_r, u_r) \in \mathcal{Z}_s$, where $\mathcal{Z}_s$ is a set contained in the interior of $\mathcal{Z}$. This allows us to ensure that the constraints $(x_r, u_r) \in \mathcal{Z}$ are not active once the system reaches the setpoint, which might lead to a loss of controllability.

For a given reference r , the MPC setpoint can be calculated by solving the following optimization problem:

$$(x_r, u_r) = \arg \min_{(x_s, u_s) \in \mathcal{Z}_s} \ell_t(x_s, u_s, r) \quad (5a)$$

$$s.t. \ x_s = Ax_s + Bu_s, \quad (5b)$$

where ℓ_t is a strictly convex function that penalizes the tracking error. It is typically chosen as the following quadratic function:

$$\ell_t(x_s, u_s, r) = \|Cx_s - r\|_{Q_s}^2 + \|u_s\|_{R_s}^2.$$

This problem is referred to as steady-state target optimization problem (Rao and Rawlings 1999).

Model Predictive Controller Design

If the reference to be tracked is a constant, i.e., $r(k) = r$ for all k , then the control objective is to stabilize the system and steer the initial state $x(0)$ to the MPC setpoint (x_r, u_r) .

For a given reference r , first the MPC setpoint is calculated, and then the MPC is designed to regulate the system to this setpoint. Therefore, the MPC optimization problem depends on the current state x and the constant reference r (notice that the MPC setpoint is a function of the reference r), i.e., $P_N(x, r)$. Its solution gives an optimal control sequence $\mathbf{u}^o(x, r) = \{u^o(0; x, r), u^o(1; x, r), \dots, u^o(N-1; x, r)\}$ and an optimal state trajectory $\mathbf{x}^o(x, r) = \{x^o(0; x, r) = x, x^o(1; x, r), \dots, x^o(N; x, r)\}$, where N is the prediction horizon. The first element of this sequence, namely, $u^o(0; x, r)$, is applied to the system.

In this chapter, a stabilizing design of the predictive controller based on a terminal cost and a terminal constraint is adopted. This is basically a regulation MPC penalizing the deviation w.r.t. the MPC setpoint. The optimization problem $P_N(x, r)$ is defined as follows:

$$\min_{\mathbf{u}} \sum_{j=0}^{N-1} \ell(x(j), u(j), r) + V_f(x(N), r) \quad (6a)$$

$$s.t. \ x(0) = x, \quad (6b)$$

$$x(j+1) = Ax(j) + Bu(j), \quad j \in \mathbb{I}_{N-1} \quad (6b)$$

$$(x(j), u(j)) \in \mathcal{Z}, \quad j \in \mathbb{I}_{N-1} \quad (6c)$$

$$x(N) \in X_f(r). \quad (6d)$$

The stage cost function $\ell(\cdot)$ is a measure of the predicted tracking error, that is, $\ell(x_r, u_r, r) = 0$ and $\ell(x, u, r) \geq \alpha_1(\|x - x_r\|)$. The terminal cost function $V_f(\cdot)$ is such that

$$\alpha_2(\|x - x_r\|) \leq V_f(x_r, r) \leq \alpha_3(\|x - x_r\|).$$

The functions α_1 , α_2 , and α_3 are $\mathcal{K}_\infty$ functions. (A function $\alpha : [0, \infty) \rightarrow [0, \infty)$ is $\mathcal{K}_\infty$ if it

is continuous, strictly increasing, $\alpha(0) = 0$ and $\lim_{s \rightarrow \infty} \alpha(s) = \infty$.)

The set of states where this optimization problem is feasible depends on the reference r , and it is denoted as $X_N(r)$. The solution of the optimal control problem $P_N(x, r)$ yields the receding horizon control law

$$\kappa_N(x, r) = u^o(0; x, r),$$

and the system controlled by this model predictive control is given by

$$x(k+1) = Ax(k) + B\kappa_N(x(k), r). \quad (7)$$

Because the horizon N is finite, x_r is not necessarily asymptotically stable for this system, but asymptotic stability can be ensured if the terminal cost function $V_f(\cdot)$ and the terminal region $X_f(r)$ are chosen appropriately. To this aim, the functions $\ell(\cdot)$, $V_f(\cdot)$ and the set $X_f(r)$ must satisfy the following condition:

Stability conditions for tracking MPC: For all $x \in X_f(r)$, there exists a control input u such that $(x, u) \in \mathcal{Z}$, the successor state $x^+ = Ax + Bu$ is contained in $X_f(r)$ and

$$V_f(x^+, r) - V_f(x, r) \leq -\ell(x, u, r).$$

Under these stability conditions, the optimization problem is recursively feasible, i.e., if $P_N(x(0), r)$ is feasible, then all subsequent problems $P_N(x(i), r)$ are also feasible. Besides, the optimal cost function is a Lyapunov function of the system (7). Then, the MPC setpoint (x_r, u_r) is an asymptotically stable equilibrium point of the system (7), and the domain of attraction is $X_N(r)$ (Rawlings et al. 2017). A particular case when the stability conditions are trivially satisfied is taking $X_f(r) = x_r$ and $V_f(x, r) = 0$.

Tracking MPC for a Changing Reference

The previous predictive controller is inherently deterministic, since it is assumed that the reference is known, and this will remain constant in

the future. However, in a realistic scenario, the reference may be changed without a predefined deterministic law. As it will be shown next, the changing references may make the stabilizing design of (6) fail. In this section, a tracking predictive controller for the case when the reference is piecewise constant (or varying, but ultimately converging to a constant value) is presented.

Feasibility and Stability Issues

If the reference r is constant, then tracking MPC (6) ensures asymptotic stability of the target state x_r and convergence to zero of the tracking error $(y - r)$. However, if the reference r varies, recursive feasibility (i.e., feasibility of $P_N(x(k), r(k))$ at each time instant k) and asymptotic stability may be compromised.

Consider that the tracking MPC has been designed to steer the system to the reference $r = r_1$ and the state x is in the domain of attraction $X_N(r_1)$. If the value of r changes from r_1 to r_2 , the feasibility region, i.e., the domain of attraction of the controller, changes from $X_N(r_1)$ to $X_N(r_2)$. The current state x , which lies in $X_N(r_1)$, does not necessarily lie in $X_N(r_2)$ so that $\kappa_N(x, r_2)$ would be undefined and the model predictive controller would fail. Furthermore, the terminal constraint set $X_f(r_1)$ and the terminal cost function $V_f(\cdot, r_1)$ might not satisfy the stabilizing conditions of the tracking MPC (6) for the reference r_2 which would require their recomputation for the new reference.

This phenomenon is illustrated for the double integrator system, for which the matrices of model (1) are given by,

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0.5 \\ 1 & 0.5 \end{bmatrix}, \quad C = [1 \ 0],$$

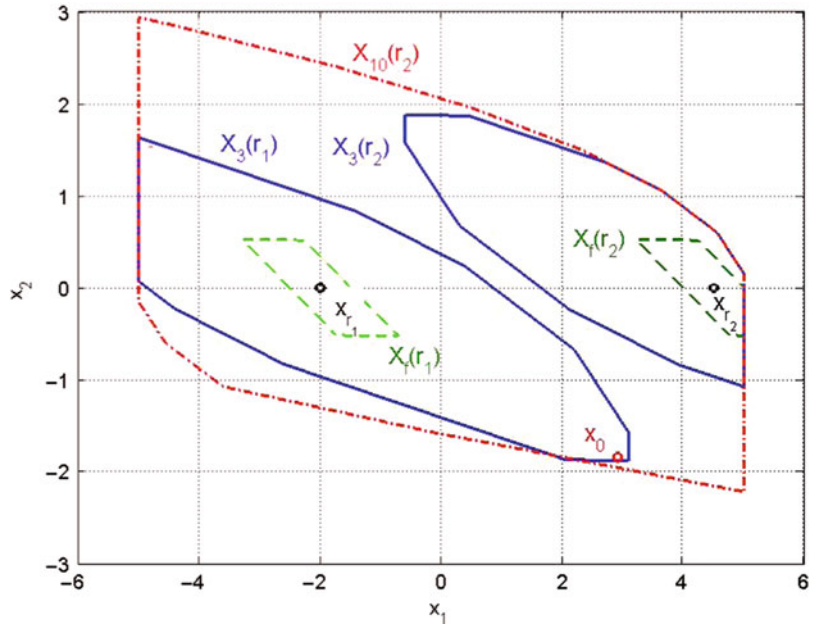
and the set of constraints is given by

$$\mathcal{Z} = \{(x, u) : \|x\|_\infty \leq 5, \|u\|_\infty \leq 0.3\}.$$

The initial state is $x(0) = (2.91, -1.83)$ and the initial value of the reference is $r_1 = -2$. The corresponding MPC setpoint is (x_{r_1}, u_{r_1}) where $x_{r_1} = (-2, 0)$ and $u_{r_1} = (0, 0)$. If the reference

Tracking Model Predictive Control, Fig. 1

Example of the double integrator: terminal regions ($X_f(r_1)$ and $X_f(r_2)$) and domains of attraction of MPC ($X_3(r_1)$, $X_3(r_2)$, and $X_{10}(r_2)$)



changes from $r_1 = -2$ to $r_2 = 4.5$, then the new MPC setpoint is (x_{r_2}, u_{r_2}) where $x_{r_2} = (4.5, 0)$ and $u_{r_2} = (0, 0)$. The horizon is chosen to be $N = 3$, and the domains of attraction for the two values of r are, respectively, $X_3(r_1)$ and $X_3(r_2)$. As it is shown in Fig. 1, these two domains, $X_3(r_1)$ and $X_3(r_2)$, are disjoint. While $r = r_1$, the state trajectory commencing at $x(0) \in X_3(r_1)$ remains in $X_3(r_1)$. If r subsequently changes its value to r_2 at the sample k_1 , the model predictive controller fails since $x(k_1)$ does not lie in $X_3(r_2)$.

These feasibility and stability issues can be overcome if the predictive controller is redesigned for the new setpoint. This would require the calculation of a new terminal set and a prediction horizon each time the setpoint changes. For instance, in the example of Fig. 1, if the terminal constraint is recalculated for r_2 and the prediction horizon is chosen as $N = 10$, then when the reference r changes from r_1 to r_2 , the MPC controller to be used is $\kappa_{10}(\cdot, r_2)$, which steers the system to the reference r_2 since $x(0) \in X_3(r_1) \subset X_{10}(r_2)$. This recalculation can be done off-line if the MPC setpoint changes are known a priori (Findeisen et al. 2000; Wan and Kothare 2003). Other methods to avoid this issues are: design a predictive controller to provide a certain degree of robustness to setpoint variations

(Pannocchia and Kerrigan 2005; Pannocchia 2004), a predictive control law with a mode to recover recursive feasibility (Rossiter et al. 1996; Chisci and Zappa 2003), or using specialized predictive control laws (Magni et al. 2001; Magni and Scattolini 2005). Another solution is to use a reference governor and a predictive controller (Bemporad et al. 1997; Oлару and Dumur 2005).

Stabilizing MPC for Tracking

The idea behind the reference governor is to introduce an artificial reference r^a that is manipulated to ensure that the current state is in the domain of attraction $X_N(r^a)$ and which tends to the actual reference r if r remains constant or tends to a constant. In Limon et al. (2008), this idea is used to formulate the so-called MPC for tracking. The artificial reference r^a is an extra decision variable in the optimal control problem that avoids the loss of feasibility issue. In order to enforce the convergence to the actual reference r , a term that penalizes the deviation between the artificial reference r^a and the actual reference r , $\ell_o(r^a, r)$ is added. This function is assumed to be convex in r^a . A suitable choice of this term is the cost function of the steady-state target calculator (5), i.e., $\ell_o(r^a, r) = \ell_t(x_{r^a}, u_{r^a}, r)$, where

(x_{r^a}, u_{r^a}) is the artificial setpoint associated to the artificial reference r^a .

The optimal control problem $P_N^t(x, r)$ of the MPC for tracking is given by

$$\begin{aligned} \min_{\mathbf{u}, r^a} \quad & \sum_{j=0}^{N-1} \ell(x(j), u(j), r^a) \\ & + V_f(x(N), r^a) + \ell_o(r^a, r) \\ \text{s.t. } & x(0) = x, \end{aligned} \quad (8a)$$

$$x(j+1) = Ax(j) + Bu(j), \quad j \in \mathbb{I}_{N-1} \quad (8b)$$

$$(x(j), u(j)) \in \mathcal{Z}, \quad j \in \mathbb{I}_{N-1} \quad (8c)$$

$$r^a \in \mathcal{R} \quad (8d)$$

$$(x(N), r^a) \in \Gamma, \quad (8e)$$

where

$$\begin{aligned} \mathcal{R} &= \{r : (x_r, u_r) \in \mathcal{Z}_s, \\ & Ax_r + Bu_r = x_r, Cx_r = r\}. \end{aligned} \quad (9)$$

Condition (8e) is an extended terminal constraint of both, the terminal state $x(N)$ and the artificial reference r^a . The feasibility region of this optimization problem X_N^t is the set of states that can be steered to any reference of the set $\mathcal{R}$ in N steps, that is,

$$X_N^t = \bigcup_{r^a \in \mathcal{R}} X_N(r^a).$$

The terminal cost function $V_f(\cdot)$ and the terminal constraint set, Γ , must satisfy appropriately modified stability conditions in order to ensure recursive feasibility and asymptotic stability of (x_r, u_r) . The stability conditions are:

Stability conditions of MPC for tracking: For all $(x, r^a) \in \Gamma$: $r^a \in \mathcal{R}$, and there exist a u satisfying: (i) $(x, u) \in \mathcal{Z}$, (ii) the successor state $x^+ = Ax + Bu$ is such that $(x^+, r^a) \in \Gamma$ and $V_f(x^+, r^a) - V_f(x, r^a) \leq -\ell(x, u, r^a)$.

As shown in Limon et al. (2008), if the control action u of the former stability conditions is given by the following so-called terminal control law $u = K(x - x_{r^a}) + u_{r^a}$ with K such that the

eigenvalues of $A + BK$ are in the unitary disk, then the terminal set Γ is a polyhedron that can be computed using standard algorithms to compute positively invariant sets for constrained linear systems. A simple choice of the terminal cost and constraint satisfying these stability conditions is $V_f(\cdot) = 0$ and $\Gamma = \{(x, r^a) : x = x_{r^a}\}$.

Theorem 1 *If the stability conditions of the MPC for tracking hold, then the predictive control law derived from the optimal control problem $P_N^t(x, r)$ is such that:*

- (1) *For every feasible initial state, i.e., $x(0) \in X_N^t$ and for all $r \in \mathbb{R}^p$, the optimization problem is recursively feasible, that is, if $P_N^t(x(0), r)$ is feasible, then all the subsequent problems $P_N^t(x(i), r)$ will also be feasible.*
- (2) *If r is admissible, i.e., $r \in \mathcal{R}$, then the setpoint (x_r, u_r) is an asymptotically stable equilibrium point of the closed-loop system, and the domain of attraction is X_N^t .*
- (3) *If r is not admissible, that is, $r \notin \mathcal{R}$, then the setpoint (x_{r^*}, u_{r^*}) such that*

$$r^* = \arg \min_{r^a \in \mathcal{R}} \ell_o(r^a, r)$$

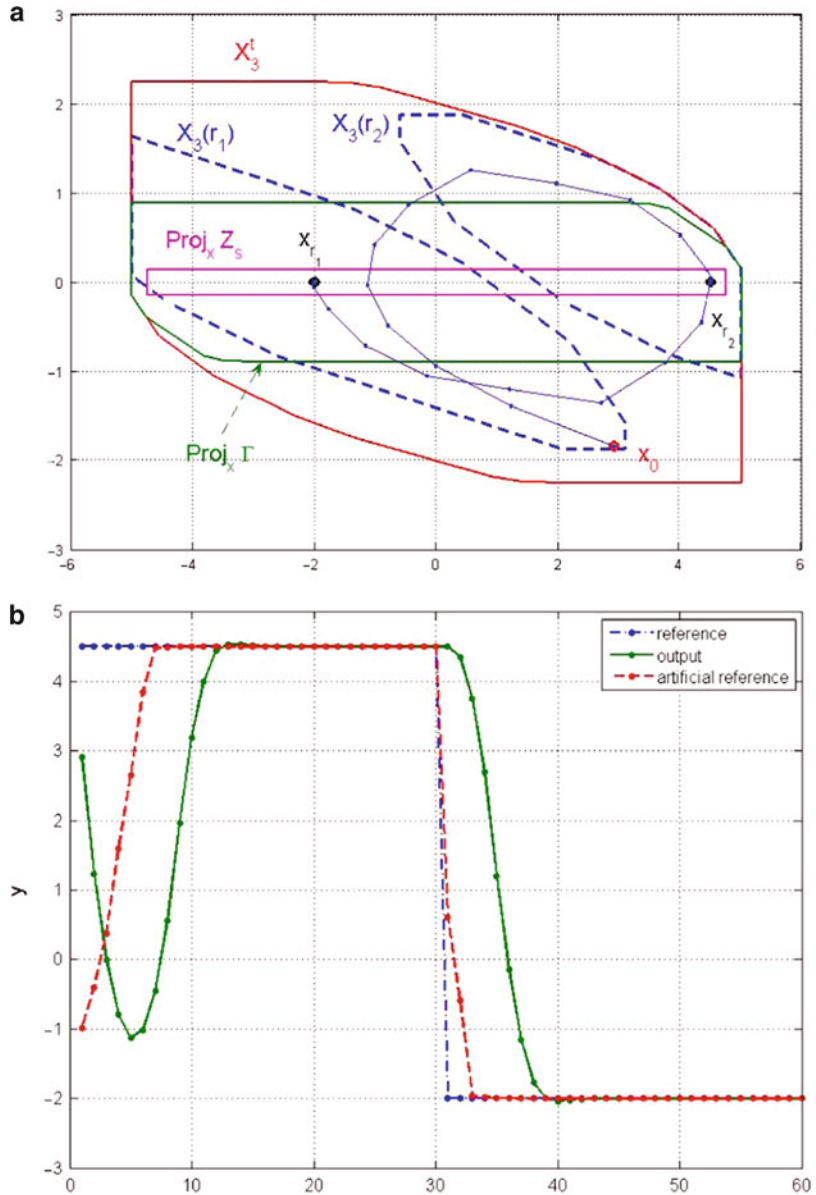
is asymptotically stable, and the domain of attraction is X_N^t .

- (4) *The domain of attraction X_N^t is larger than or equal to the domain of the tracking MPC for any reference $r \in \mathcal{R}$, that is, $X_N(r) \subseteq X_N^t$, and contains all the equilibrium points that lie in $\mathcal{Z}_s$.*
- (5) *If the reference $r(k)$ is not constant and converges to an steady value r , the optimization problem is recursively feasible, and the setpoint (x_{r^*}, u_{r^*}) is an asymptotically stable equilibrium point for all $x(0) \in X_N^t$.*

In Fig. 2 a, the aforementioned properties are illustrated for the example of the double integrator. The MPC for tracking has been designed with the same prediction horizon $N = 3$ and the same terminal control law and terminal cost function as in the previous tracking MPC case. The initial state is also the same, and the reference signal is $r(k) = r_2$ for $k \leq 30$ and $r(k) = r_1$ for

Tracking Model Predictive Control,

Fig. 2 The double integrator controlled by the MPC for tracking. **(a)** Comparison of the domains of attraction of the tracking MPC $X_3(r_1)$ and $X_3(r_2)$ vs. the domain of attraction of the MPC for tracking X_3^t . **(b)** Trajectories of the reference, the controlled output, and the artificial reference r^a



$k > 30$. Notice that the tracking MPC cannot be used to do this without redesign. In Fig. 2 a, it can be seen that the domain of attraction of the MPC for tracking X_3^t is larger than the domain provided by the standard tracking MPC, $X_3(r_1)$, or $X_3(r_2)$. This figure also shows the state portrait of the closed-loop trajectory. In Fig. 2 b, the trajectories of the reference signal $\mathbf{r}$, the controlled output y , and the artificial reference r^a are depicted.

Notice the role of the artificial reference: r^a differs from the reference in order to guarantee recursive feasibility and finally converges to the reference r to enforce asymptotic stability.

The stabilizing MPC for tracking has been extended to the case of nonlinear prediction models in Limon et al. (2018). This controller inherits the properties of its linear prediction model counterpart presented in this section.

Offset-Free Tracking

In practice, there may exist mismatches between the prediction model and the dynamics of the real plant to be controlled, due, for instance, to un-modeled nonlinearities or unmeasured disturbances. This would require to design the predictive controller to be robust to these uncertain effects. Even in the case that the predictive controller based on nominal predictions makes that the uncertain controlled system is stable and converges to a steady state, there may exist an steady error between reference and the output.

This offset can be canceled taking into account a prediction model corrected by a disturbance model (Pannocchia and Rawlings 2003; Maeder and Morari 2010). To achieve offset-free control, the disturbance is assumed to be constant, leading to the following augmented model:

$$x(k+1) = Ax(k) + Bu(k) + B_d d(k), \quad (10a)$$

$$d(k+1) = d(k), \quad (10b)$$

$$y(k) = Cx(k) + D_d d(k). \quad (10c)$$

Matrices B_d and D_d define the disturbance model and are typically chosen as $B_d = 0$ and $D_d = I_p$ (Pannocchia and Rawlings 2003).

The state $\hat{x}$ and the disturbance signal $\hat{d}$ are estimated using an observer based on the disturbance model (10). The disturbance model and the estimator gains can be calculated separately, but this may lead to a poor closed-loop performance. A joint design procedure has been proposed in Pannocchia and Bemporad (2007).

Once the estimated state and disturbance, $(\hat{x}, \hat{d})$ are available, the corrected prediction model

$$x(k+1) = Ax(k) + Bu(k) + B_d \hat{d}, \quad (11a)$$

$$y(k) = Cx(k) + D_d \hat{d}, \quad (11b)$$

must be used to replace the nominal prediction model in the steady-state target optimizer (5b) and in the tracking MPC (6b). In the MPC for

tracking, the prediction model in (8b) and in (9) must be replaced by (11).

Future Directions

Tracking model predictive control is an inherently uncertain control problem due to the unexpected changes in the reference. Constant reference tracking has been widely studied, and there exist a number of nice solutions.

The case of trajectory tracking is not as mature as the setpoint tracking case. If the reference signal is known a priori, this can be used to design a suitable tracking predictive controller. This control problem can be solved by using a two-layer structure: a trajectory planning, aimed to calculate the optimal reachable trajectory, on top of a predictive control law that steers the system to the optimal reference. Asymptotic stability to the target trajectory can be achieved resorting to a regulation problem, using a terminal equality constraint, for instance. Another interesting line is to assume that the reference is the output of a certain dynamic system. For different families of trajectories, such as ramps or sinusoidal signals, Maeder (Maeder and Morari 2010) has proposed a reference tracking MPC based on extended disturbance models.

The problem of tracking MPC in case of unknown (or changing) reference signals is challenging and can be considered an open problem that deserves more research efforts. The MPC for tracking has been recently extended to deal with periodic references (Limon et al. 2016). This controller can asymptotically stabilize the system to the best reachable periodic trajectory even when the reference is changed. The only required assumption on the reference is that its period is known. The extension to cope with a more general class of references is an open problem.

Another interesting control problem is the tracking of unreachable (equilibrium point as well as trajectory) targets. Recently this problem has been posed as an economic model predictive control problem (Rawlings et al. 2017). Therefore, the stabilizing design of economic MPC (Angeli et al. 2012) can be extended to the case of tracking unreachable targets.

Cross-References

- [Economic Model Predictive Control](#)
- [Nominal Model-Predictive Control](#)
- [Regulation and Tracking of Nonlinear Systems](#)
- [Tracking and Regulation in Linear Systems](#)

Recommended Reading

The book (Camacho and Bordons 2004) covers the classic approach to the tracking MPC. In Rawlings et al. (2017), the authors deal with the tracking MPC in a very general and clear way and survey existing results on stability, target calculation, and offset-free control for linear and nonlinear models. In Muske (1997), the reachability of setpoints is studied and in Rao and Rawlings (1999) the target calculation problem. Disturbance models are widely analyzed in Pannocchia and Rawlings (2003), Pannocchia and Bemporad (2007), Maeder et al. (2009), and Maeder and Morari (2010). Another offset-free MPC based on the internal model principle can be found in Magni and Scattolini (2007). Further results on MPC for tracking are addressed in Ferramosca et al. (2009) and Limon et al. (2016, 2018). A survey on the MPC for tracking can be found in Limon et al. (2012). The authors have also extended the MPC for tracking to the robust case both in constant reference and periodic reference framework (Pereira et al. (2017)).

Acknowledgments The authors would like to thank the support received by the MINECO-Spain and FEDER Funds under project DPI2016-76493-C3-1-R.

Bibliography

- Angeli D, Amrit R, Rawlings JB (2012) On average performance and stability of economic model predictive control. *IEEE Trans Autom Control* 57:1615–1626
- Bemporad A, Casavola A, Mosca E (1997) Nonlinear control of constrained linear systems via predictive reference management. *IEEE Trans Autom Control* 42:340–349
- Camacho EF, Bordons C (2004) *Model predictive control*, 2nd edn. Springer, New York
- Chisci L, Zappa G (2003) Dual mode predictive tracking of piecewise constant references for constrained linear systems. *Int J Control* 76:61–72
- Ferramosca A, Limon D, Alvarado I, Alamo T, Camacho EF (2009) MPC for tracking with optimal closed-loop performance. *Automatica* 45:1975–1978
- Findeisen R, Chen H, Allgöwer F (2000) Nonlinear predictive control for setpoint families. In: *Proceedings of the American control conference*
- Limon D, Alvarado I, Alamo T, Camacho EF (2008) MPC for tracking of piece-wise constant references for constrained linear systems. *Automatica* 44:2382–2387
- Limon D, Ferramosca A, Alamo T, Gonzalez AH (2012) Model predictive control for changing economic targets. Paper presented at the IFAC conference on nonlinear model predictive control 2012 (NMPC'12). Noordwijkerhout, 23–27 Aug 2012
- Limon D, Pereira M, Muñoz de la Peña D, Alamo T, Jones CN, Zeillinger MN (2016) MPC for tracking periodic references. *IEEE Trans Autom Control* 61:1123–1128
- Limon D, Ferramosca A, Alvarado I, Alamo T (2018) Nonlinear MPC for tracking piece-wise constant reference signals. *IEEE Trans Autom Control* 63:3735–3750
- Maeder U, Morari M (2010) Offset-free reference tracking with model predictive control. *Automatica* 46(9):1469–1476
- Maeder U, Borrelli F, Morari M (2009) Linear offset-free model predictive control. *Automatica* 45:2214–2222
- Magni L, Scattolini R (2005) On the solution of the tracking problem for non-linear systems with MPC. *Int J Syst Sci* 36(8):477–484
- Magni L, Scattolini R (2007) Tracking on non-square nonlinear continuous time systems with piecewise constant model predictive control. *J Process Control* 17: 631–640
- Magni L, De Nicolao G, Scattolini R (2001) Output feedback and tracking of nonlinear systems with model predictive control. *Automatica* 37:1601–1607
- Muske K (1997) Steady-state target optimization in linear model predictive control. Paper presented at the 16th American control conference, Albuquerque, 4–6 June 1997
- Olaru S, Dumur D (2005) Compact explicit MPC with guarantee of feasibility for tracking. In: *Conference decision and control and European control conference 2005*, pp 969–974
- Pannocchia G (2004) Robust model predictive control with guaranteed setpoint tracking. *J Process Control* 14:927–937
- Pannocchia G, Bemporad A (2007) Combined design of disturbance model and observer for offset-free model predictive control. *IEEE Trans Autom Control* 52:1048–1053
- Pannocchia G, Kerrigan E (2005) Offset-free receding horizon control of constrained linear systems. *AIChE J* 51:3134–3146
- Pannocchia G, Rawlings JB (2003) Disturbance models for offset-free model-predictive control. *AIChE J* 49:426–437

- Pereira M, Muñoz de la Peña D, Limon D, Alvarado I, Alamo T (2017) Robust model predictive controller for tracking changing periodic signals. *IEEE Trans Autom Control* 62:5343–5350
- Rao CV, Rawlings JB (1999) Steady states and constraints in model predictive control. *AIChE J* 45:1266–1278
- Rawlings JB, Mayne DQ, Diehl MM (2017) *Model predictive control: theory, computation and design*, 2nd edn. Nob-Hill Publishing, Madison
- Rossiter JA, Kouvaritakis B, Gossner JR (1996) Guaranteeing feasibility in constrained stable generalized predictive control. *IEEE Proc Control Theory Appl* 143:463–469
- Wan Z, Kothare MV (2003) An efficient off-line formulation of robust model predictive control using linear matrix inequalities. *Automatica* 39:837–846

Trajectory Generation for Aerial Multicopters

Mark W. Mueller¹ and Raffaello D'Andrea²

¹University of California, Berkeley, CA, USA

²ETH Zurich, Zurich, Switzerland

Abstract

The ability to generate feasible and safe trajectories is crucial for autonomous multicopter systems. These trajectories can ideally be generated on low-cost, embedded computational hardware and exploit the system's full dynamic capabilities while satisfying constraints. As operations increasingly focus on operation at high speeds, or in dynamically changing environments, strategies are required that can rapidly plan and replan trajectories. This entry reviews typical approaches for trajectory generation of aerial robots, with a focus on multicopters, and discusses various approximations that may be used to make the problem more tractable. The strategy of planning in higher derivatives of the vehicle position (such as acceleration, jerk, and snap) is discussed in depth. We also discuss the related issue of expressing system limitations and constraints in these derivatives. Finally, possible future directions are discussed.

Keywords

Multicopters · UAV · Planning · Differential flatness

Introduction

Aerial robots are increasingly routinely used in everyday operations, accomplishing tasks such as remote sensing, surveillance, and delivery of goods and people. Continuing technological development, especially of energy storage, and development of the regulatory environment mean that such systems may soon be commonplace. A crucial requirement for their autonomous operations is the ability to plan motions to achieve goals, especially trajectories moving them through space. Due to their mechanical simplicity, and the ability to hover in place when required, the most common aerial robots are multicopters, especially quadcopters.

A typical trajectory generation problem consists of moving a single multicopter from an initial state (typically described as a position, velocity, orientation, and angular velocity) to a final state. The final state may be as detailed as the initial state, or it may only specify some components (such as an emergency stopping trajectory that requires simply zero final velocity). The resulting trajectory must respect the system dynamics and potentially avoid additional constraints, such as collisions. The satisfaction of system dynamics and constraints is to be understood as satisfying some sufficiently accurate approximation of the true system capabilities (i.e., sufficiently accurate for the problem at hand). Moreover, the available resources (e.g., computational power and computation time) are typically important considerations, as is the ability to accurately model the system and disturbances.

This entry focuses on model-based approaches, that is, generation methods that rely on first-principle models of the multicopter systems to generate trajectories. An additional focus is on low computational complexity, so that trajectories may be computed on constrained hardware.

Dynamics and Low-Level Control

We briefly review the equations of motions governing the motion of a multicopter, using the example of a quadcopter with propellers located symmetrically about the center of mass, with fixed, parallel axes of rotation. The model may be easily extended to other classes of multicopter; such extensions are briefly discussed.

The vehicle body's six degrees of freedom are captured in the position vector $\mathbf{x} = (x_1, x_2, x_3)$, expressed in an inertial frame, and the rotation matrix $\mathbf{R}$ that relates the body-fixed and inertial frames. The compact notation $\mathbf{x} = (x_1, x_2, x_3)$ is used to express the elements of a column vector. These evolve over time as functions of the translation velocity $\mathbf{v}$ and angular velocity $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$ as

$$\dot{\mathbf{x}} = \mathbf{v}, \quad \dot{\mathbf{R}} = \mathbf{R} \mathbf{S}(\boldsymbol{\omega}) \quad (1)$$

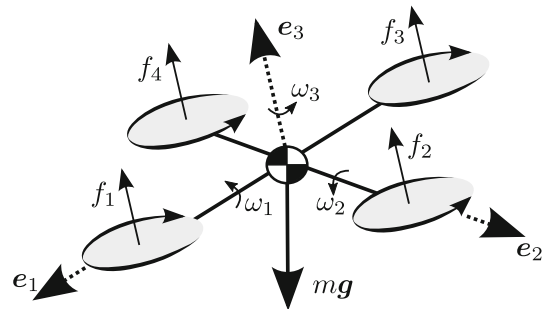
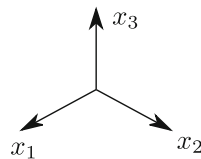
where $\mathbf{S}(\mathbf{a})$ is the skew-symmetric matrix version of the cross product, so that $\mathbf{a} \times \mathbf{b} = \mathbf{S}(\mathbf{a}) \mathbf{b}$ for any vectors $\mathbf{a}$ and $\mathbf{b}$.

Considering for simplicity a quadcopter, as illustrated in Fig. 1, the vehicle is actuated by four propeller forces f_i acting at displacements $\mathbf{r}_i$ from the center of mass. The propellers point along the body-fixed $\mathbf{e}_3$ direction and produce a pure moment τ_i about their axes of rotation in addition to the force. This moment is typically assumed to be proportional to the force produced. The governing differential equations are then given as below

$$m\mathbf{a} = \mathbf{R} \mathbf{e}_3 f_\Sigma + m\mathbf{g} + \mathbf{f}_{\text{aero}} \quad (2)$$

Trajectory Generation for Aerial Multicopters,

Fig. 1 Dynamic model of a quadcopter, acted upon by gravity $\mathbf{g}$, a thrust force f pointing along the (body-fixed) axis $\mathbf{e}_3$; and rotating with angular rate $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$, with its position in inertial space given as (x_1, x_2, x_3)



$$\mathbf{J} \dot{\boldsymbol{\omega}} = -\mathbf{S}(\boldsymbol{\omega}) \mathbf{J} \boldsymbol{\omega}$$

$$+ \sum_i (\mathbf{S}(\mathbf{r}_i) \mathbf{e}_3 f_i + \mathbf{e}_3 \tau_i) + \boldsymbol{\tau}_{\text{aero}} \quad (3)$$

with $\mathbf{a}$ the acceleration of the vehicle, $\mathbf{g}$ the acceleration due to gravity as expressed in the inertial coordinate frame, and $\mathbf{f}_{\text{aero}}$ aerodynamic forces in addition to the simple propeller model (such as propeller in-plane forces). Furthermore, $\mathbf{J}$ is the mass moment of inertia, and $\boldsymbol{\tau}_{\text{aero}}$ any aerodynamic torques acting on the system in addition to those captured by the propeller model. We use the shorthand $f_\Sigma = \sum_i f_i$ for the total thrust force.

Multicopters with more than four propellers with parallel thrust directions will have the same equations as above, except that the summation will be over more than four propellers. Vehicles with thrust directions that are not parallel also exist, and such designs may have various benefits. For quadcopters, this may improve control authority and energy efficiency (Holda et al. 2018). When using more than four propellers, this reduces (or completely eliminates) the system's underactuation; see, e.g., the hexacopter of Jiang and Voyles (2014) with improved disturbance rejection or the omnidirectional vehicle of Brescianini and D'Andrea (2018). Vehicles with fewer than four propellers can also be constructed, but require special considerations especially with regard to attitude control; for a discussion, see, e.g., Mueller and D'Andrea (2016).

The motor forces vary as a function primarily of the respective propeller angular velocity and may be accurately modelled as varying

proportionally to the angular velocity squared, though more detailed models exist considering, e.g., relative airspeed. The forces are typically assumed to vary instantaneously, though the dependence on angular velocity of the propellers implies that an instantaneous change is impossible for actual systems with finite motor torques.

The individual motor forces are typically constrained as $f_i \in [f_{\min}, f_{\max}]$, with the upper limit $f_{\max}$ following from the maximum torque characteristics of the motors, and the lower limit $f_{\min}$ following, e.g., from the capacity of the speed controllers driving the motors to operate at low rotational speeds, so that usually $f_{\min} > 0$. For brushed motors, as is common on low-cost and inexpensive systems, typically $f_{\min} = 0$, while for systems with variable pitch or reversible propellers, we may have $f_{\min} < 0$.

Closed-loop control is commonly achieved with nested control, especially by considering the angular velocity of the body as a high-bandwidth subsystem and treating a target angular velocity as instantaneously achievable. This simplification, similar to the assumption of instantaneously achievable propeller forces, is here justified by noting that the angular velocity is fully actuated, so that the nonlinear terms may be compensated by feedback linearization, and that the achievable angular accelerations are very large due to the vehicle's low mass moment of inertia (as mass is typically concentrated at the center) and high torques (due to large moment arms).

Trajectory Generation

For a general dynamic system, the goal of trajectory generation is a control input sequence, which would move it from an initial state in order to achieve a goal. This is typically posed as an optimization problem, minimizing some cost function. As the system dynamics are expressed as differential equations, the problem is most directly stated in continuous-time searching over the space of functions. Alternatively, the system dynamics may be approximated in discrete time, so that the optimization is now over a finite dimensional space. In both cases closed-form

solutions are difficult to solve for, and general problems typically can only be solved through numerical approaches. As these generic problems tend to be non-convex and nonlinear, brute-force numerical solutions tend to struggle, especially when computational power is limited. However, by exploiting the dynamic characteristics of multicopters, various transformations are possible that make the problem more easily solvable.

A key factor in the difficulty of finding a solution is the level of abstraction applied to the system dynamics. This is often implicitly done through low-level control loops, where certain states are treated as responding instantaneously to track desired values. The lowest level inputs typically considered are the electric signals to the motors, e.g., as a voltage level applied over a motor. When the aerial vehicle is equipped with high-performance motor speed controllers, the motor speeds (equivalently, the propeller forces) may be used as input; the next level up assumes sufficiently fast angular dynamics to allow the angular velocity to be used as input. Coarser models still consider the system as a double integrator, with acceleration as input (assuming, effectively, instantaneous changes in attitude), or most crudely as a single integrator with translational velocity as input. Increasingly coarse models have multiple advantages: they reduce the dimension of the planning system's state space (leading to smaller problems that are easier and faster to solve), they may hide nonlinearities (e.g., treating angular acceleration as input rather than torque avoids the nonlinearities in the angular velocity dynamics), and they may hide constraints (e.g., low-level input constraints are not visible if planning is done at higher levels).

A very successful approach for multicopters is to plan in some derivative of the position and consider each translational axis independently, thus solving three trajectories, each substantially less complicated. An early example of this kind of approach is given in Kalmár-Nagy et al. (2004), applied to a surface robot. The differential flatness properties of multicopters mean that the actual inputs and states can be recovered straightforwardly from these position derivatives, though for uniqueness a rotation about the thrust axis

is also required. A discussion on the differential flatness of multicopters is given in Mellinger and Kumar (2011), while general discussion on flatness may be found in Fliess et al. (1995) or Murray et al. (1995). Due to the superlinear complexity scaling of most optimization problems in the problem size, this spatial decomposition typically leads to a dramatic reduction in the required computational time: an example of quadcopter trajectory generation using this approach coupled with time discretization is given in Mueller and D'Andrea (2013). An obvious concern with spatial decoupling is the difficulty of decoupling constraints acting on the system – the decoupling of input constraints is discussed below, and obstacles are unlikely to present as neatly decoupled along user-specified directions.

Decoupled Planning and Constraints

When a decoupling approach is used, it is necessary to relate the planning states to the allowable inputs. A first requirement may be that the total thrust along the trajectory does not exceed the maximum available value. From (2), we have

$$|f_{\Sigma}|^2 = \|m\mathbf{a} - m\mathbf{g} - \mathbf{f}_{\text{aero}}\|^2 \quad (4)$$

Assuming, for simplicity, that the aerodynamic disturbances are negligible, the thrust bound, through (4), may be encoded per axis by specifying conservative box constraints. Let, e.g., $\bar{a}_i$ be such that $(\sum_i (\bar{a}_i^2))^{\frac{1}{2}}$ is less than the maximum allowed thrust, and then the following decoupled bounds guarantee total thrust feasibility:

$$-\bar{a}_i \leq \mathbf{e}_i^T \mathbf{a} \leq \bar{a}_i \quad (5)$$

Ensuring that the trajectory does not violate lower bounds on the thrust (relevant when $f_{\min}$ is strictly positive) is more difficult. This may be achieved by specifying an additional lower bound on the vehicle's acceleration along the gravitational direction:

$$\mathbf{e}_3^T \mathbf{a} \geq -\|\mathbf{g}\| + \frac{1}{m} \sum_i f_{\min} \quad (6)$$

This constraint is very conservative, as it effectively limits the tilt of the vehicle's thrust axis and does not, for instance, allow the vehicle to rotate its thrust axis to point downward.

Limiting the trajectory jerk $\mathbf{j} = \dot{\mathbf{a}}$ allows for applying bounds on the angular velocity and change in total thrust along the motion, but this relationship is substantially more complex than the acceleration. Taking the time derivative of the translational dynamics (2) and neglecting for simplicity changes in the aerodynamic forces yield

$$m\mathbf{R}^T \mathbf{j} = -\mathbf{S}(\mathbf{e}_3) \boldsymbol{\omega} f_{\Sigma} + \mathbf{e}_3 \dot{f}_{\Sigma} = \begin{bmatrix} -\omega_2 f_{\Sigma} \\ \omega_1 f_{\Sigma} \\ \dot{f}_{\Sigma} \end{bmatrix} \quad (7)$$

From this equation it is clear that a simple bound on the jerk (either per-axis, or total) is insufficient to bound the angular velocity along the trajectory unless additionally a minimum bound on the thrust is specified.

To make statements regarding individual motor forces, an additional derivative of position is required, the snap $\mathbf{s}$. Taking the derivative of (7) gives

$$\begin{aligned} m\mathbf{R}^T \mathbf{s} &= \mathbf{S}(\boldsymbol{\omega})^2 \mathbf{e}_3 f_{\Sigma} + \mathbf{S}(\boldsymbol{\omega}) \mathbf{e}_3 \dot{f}_{\Sigma} \\ &\quad - \mathbf{S}(\mathbf{e}_3) \dot{\boldsymbol{\omega}} f_{\Sigma} + \mathbf{S}(\boldsymbol{\omega}) \mathbf{e}_3 \dot{f} + \mathbf{e}_3 \ddot{f}_{\Sigma} \end{aligned} \quad (8)$$

where the angular dynamics of (3) can be substituted for $\dot{\boldsymbol{\omega}}$ to relate motor forces to snap. As before, a bounded (per-axis or total) snap does not imply that the system's states or inputs remain bounded – one example is that the angular acceleration component parallel along the body's $\mathbf{e}_3$ axis may become unbounded while the snap remains bounded. A more subtle example is of a vehicle starting at rest at time $t = 0$, with acceleration trajectory $\mathbf{a}(t) = \mathbf{g}t$. In such a motion, the vehicle starts at rest with $f_{\Sigma} = m\|\mathbf{g}\|$, at $t = 1$ is in free-fall, and at $t = 2$ is thrusting downward with $f_{\Sigma} = m\|\mathbf{g}\|$. For this motion the total thrust requirements are benign (ranging between zero

and the hover thrust), the jerk is constant at $\mathbf{j} = \mathbf{g}$, and the snap is identically zero. However, this is a physically impossible trajectory, as the thrust direction at $t = 1$ must rotate instantaneously from pointing vertically upward to pointing downward. Thus, naively planning trajectories with bounded position derivatives may result in motions requiring unbounded motor forces or unbounded angular velocities. Ensuring that the forces remain bounded would require simultaneously considering all axes, thereby negating the main benefit of the original decoupling. However, the above issues may not be common in many applications, and planning separately in position derivatives has been very successful with wide application in the literature.

Closed-Form Solutions

Under certain circumstances, closed-form solutions can be found for trajectories. Such solutions are typically computationally inexpensive to generate, but are restricted to work in specific circumstances. These kinds of trajectories naturally lend themselves to being used as motion primitives, which can be concatenated together to achieve complex goals. Moreover, if they are sufficiently inexpensive to compute, they may be used in sampling-based approaches where their computational complexity is traded off against their specialized nature.

An example of exploiting computationally cheap primitives is given in Mueller et al. (2015): here, a quadcopter trajectory primitive is proposed for state-interception trajectories (i.e., with fixed final times) that minimize the average jerk along the motion. From (7) it can be seen that minimizing jerk along a motion can be interpreted as reducing the product of angular velocity and total thrust, intuitively leading to qualitatively “nice” trajectories. The closed-form solution of the trajectories reduces to a simple affine transformation of the initial/final states; and feasibility (interpreted here as collisions with planar objects in space; minimum/maximum total thrust values; and maximum angular velocity) is evaluated using a recursive scheme. In total, a motion primitive can be generated and evaluated for feasibility in approximately 1μ s on a laptop

computer or 100μ s on a simple micro-controller. This computational speed then immediately lends itself to searching over complex spaces and operation in very dynamic situations.

Numerical Solutions

Other approaches to trajectory generation rely on numerical optimization tools. Such numerical solutions can be posed directly on the nonlinear aerial robot dynamics and thus avoid the complexities of spatial decomposition, etc. A disadvantage of such approaches tends to be substantially higher computational loads, though tools such as the ACADO toolbox (Houska et al. 2011) allow for MPC implementations that can be run on embedded hardware.

Because the problems can be posed in a generic fashion, it is easy to include other objectives into the trajectory generation problem. One example of this is to include knowledge of the multicopter’s sensors and generate motions that not only satisfy feasibility constraints related to inputs and collisions but also ensure that motions are informative to the system’s sensors. An example that optimizes for visibility of features to assist in visual localization is given by Falanga et al. (2018), or for optimizing exploration gain in Papachristos et al. (2017).

An example using a specialized numerical solver to compute minimum time trajectories is given in Hehn et al. (2012), specifically used to argue about fundamental dynamic limits of quadcopters.

Computational Speed and Receding Horizon Control

If computing a trajectory requires a substantial time, compared to the system dynamics, the trajectory is typically used as a reference trajectory tracked by a separate feedback controller. However, if the computation is sufficiently fast, trajectories may be continuously replanned as the system moves. Such receding horizon implementations have multiple performance advantages and allow systems to naturally react to disturbances or to respond in changes in objective (e.g., the sudden appearance of an obstacle) – see, e.g., Chen et al. (2016) and Nægeli et al. (2017).

Summary and Future Directions

Trajectory generation for multicopters benefits from the same technological advancements that enable autonomy for an ever-increasing set of robots, primarily increases in computational power (and a reduction in mass, and power requirements, for computation), and improvements in generic numerical optimization tools. When sufficient computational power is available, this allows solving relatively straightforward encodings of problems, including constraints on inputs and states. Where computational power is constrained, or time is of extreme importance, it is often useful to create methods that rely on specific characteristics of multicopters, such as the spatial decoupling discussed earlier.

A relatively open area of future work is to more tightly couple the trajectory planning component of autonomy to the sensing systems. Typical state-of-the-art approaches decouple the planning problem from the estimation problem: for example, a map of obstacles may first be created, and this map is then used for planning in a second step. This approach is very good at isolating complexity and follows typical engineering practice to “divide and conquer” complex problems. However, much like designing a state feedback controller independently of an estimator to regulate a plant, such a decomposed approach may have a substantial performance penalty compared to a unified approach when uncertainty is taken into account (consider $\mathcal{H}_\infty$ output-feedback control vs. LQG control – see, e.g., Zhou and Doyle 1998). This may take the form of planning trajectories directly based on the sensors, e.g., planning motions in a laser point cloud. If combined with low computational complexity methods, this may lead to very responsive behavior that naturally takes uncertainty in the sensors and system into account.

Another area of relevance for future work is to plan trajectories that take energy and power consumption into account. This is of particular importance for multicopters, as a primary constraint to their widespread use is limitations in range and payload due to modest energy budgets.

Examples exploring this include Ware and Roy (2016) and Tagliabue et al. (2019). Trajectories for multiple vehicles operating simultaneously are also challenging, due to increasing size of the state space and mutual collision avoidance; for large collections of vehicles, this remains an open problem. Another useful direction would be strategies that allow an operator to directly specify safety objectives (e.g., by parametrizing the risk to third parties from certain maneuvers).

Cross-References

- [Differential Geometric Methods in Nonlinear Control](#)
- [Iterative Learning Control](#)
- [Model Predictive Control for Power Networks](#)
- [Underactuated Robots](#)
- [Unmanned Aerial Vehicle \(UAV\)](#)

Recommended Reading

A good overview of convex optimization can be found in Boyd and Vandenberghe (2004), and a discussion of tools specifically for embedded applications is given by Ferreau et al. (2017).

A good overview of multicopter modeling and control is given by Mahony et al. (2012), with more specialized models used for, e.g., fault tolerance in Mueller and D’Andrea (2016).

Learning may be applied to aid in trajectory tracking; some examples of this use frequency-based methods (Hehn and D’Andrea 2014) or deep neural nets (Zhou et al. 2017).

Bibliography

- Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge, England
- Brescianini D, D’Andrea R (2018) An omni-directional multirotor vehicle. *Mechatronics* 55:76–93
- Chen J, Liu T, Shen S (2016) Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments. In: 2016 IEEE

- international conference on robotics and automation (ICRA). IEEE, pp 1476–1483
- Falanga D, FoeHN P, Lu P, Scaramuzza D (2018) PAMPC: perception-aware model predictive control for quadrotors. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 1–8
- Ferreau HJ, Almér S, Verschueren R, Diehl M, Frick D, Domahidi A, Jerez JL, Stathopoulos G, Jones C (2017) Embedded optimization methods for industrial automatic control. *IFAC-PapersOnLine* 50(1): 13194–13209
- Fliess M, Lévine J, Martin P, Rouchon P (1995) Flatness and defect of non-linear systems: introductory theory and examples. *Int J Control* 61(6):1327–1361
- Hehn M, D'Andrea R (2014) A frequency domain iterative learning algorithm for high-performance, periodic quadcopter maneuvers. *Mechatronics* 24(8):954–965
- Hehn M, Ritz R, D'Andrea R (2012) Performance benchmarking of quadrotor systems using time-optimal control. *Auton Robots* 33:69–88
- Holda C, Ghalamchi B, Mueller MW (2018) Tilting multicopter rotors for increased power efficiency and yaw authority. IEEE, pp 143–148
- Houska B, Ferreau H, Diehl M (2011) ACADO toolkit – an open source framework for automatic control and dynamic optimization. *Optimal Control Appl Methods* 32(3):298–312
- Jiang G, Voyles R (2014) A nonparallel hexrotor UAV with faster response to disturbances for precision position keeping. In: 2014 IEEE international symposium on safety, security, and rescue robotics. IEEE, pp 1–5
- Kalmár-Nagy T, D'Andrea R, Ganguly P (2004) Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle. *Robot Auton Syst* 46(1):47–64
- Mahony R, Kumar V, Corke P (2012) Aerial vehicles: modeling, estimation, and control of quadrotor. *IEEE Robot Autom Magaz* 19(3):20–32
- Mellinger D, Kumar V (2011) Minimum snap trajectory generation and control for quadrotors. In: IEEE international conference on robotics and automation (ICRA), pp 2520–2525
- Mueller MW, D'Andrea R (2013) A model predictive controller for quadcopter state interception. In: European control conference, pp 1383–1389
- Mueller MW, D'Andrea R (2016) Relaxed hover solutions for multicopters: application to algorithmic redundancy and novel vehicles. *Int J Robot Res* 35(8):873–889
- Mueller MW, Hehn M, D'Andrea R (2015) A computationally efficient motion primitive for quadcopter trajectory generation. *IEEE Trans Robot* 31(6): 1294–1310
- Murray RM, Rathinam M, Sluis W (1995) Differential flatness of mechanical control systems: a catalog of prototype systems. In: ASME international mechanical engineering congress and exposition
- Nägeli T, Alonso-Mora J, Domahidi A, Rus D, Hilliges O (2017) Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization. *IEEE Robot Autom Lett* 2(3):1696–1703
- Papachristos C, Khattak S, Alexis K (2017) Uncertainty-aware receding horizon exploration and mapping using aerial robots. In: 2017 IEEE international conference on robotics and automation (ICRA). IEEE, pp 4568–4575
- Tagliabue A, Wu X, Mueller MW (2019) Model-free online motion adaptation for optimal range and endurance of multicopters. In: IEEE international conference on robotics and automation (ICRA)
- Ware J, Roy N (2016) An analysis of wind field estimation and exploitation for quadrotor flight in the urban canopy layer. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE, pp 1507–1514
- Zhou K, Doyle JC (1998) *Essentials of robust control*, vol 104. Prentice Hall, Upper Saddle River
- Zhou S, Helwa MK, Schoellig AP (2017) Design of deep neural networks as add-on blocks for improving impromptu trajectory tracking. In: 2017 IEEE 56th annual conference on decision and control (CDC). IEEE, pp 5201–5207

Transmissions

Luigi Iannelli

Università degli Studi del Sannio, Benevento, Italy

Abstract

Automotive transmissions are fundamental components in modern vehicles. They are required to make the engine operating at the most efficient conditions for providing the necessary torque at the wheels and minimising the fuel consumptions. Moreover, transmissions should be able to smooth or to filter out power source torque oscillations that can appear in the driveline. For achieving such objectives, the automotive industry has looked at different technological solutions. The introduction of electronically controlled transmissions contributed to augment the possibilities of new solutions that would not have been implementable without the flexibility and the performance of electronic control. Thus, recent technological developments of automotive transmissions gave the opportunity to engineers and control scientists

for investigating challenging control problems with daily life practical applications.

Keywords

Electronic control · Driveline · Powertrain · Gear shifting · Torque converter · Dry clutch · Clutch engagement · Optimal control · Multivariable control · Electrohydraulic · Sensors · Actuators · Smart materials

Introduction

In motor vehicles the transmission is an important system that transfers the power generated by the *internal combustion engine* to the wheels, according to the driver's requests. The transmission, together with the engine, driveshaft, differential and driven wheels, constitutes the *powertrain* (Sometimes it is used the term driveline to denote the powertrain excluding the engine and the transmission.) (or drive train). The first fundamental objective of a transmission is to adjust the ratio between the wheels speed and the engine speed in order to achieve the optimal operating point of the engine, independently on the vehicle velocity. Indeed, typical internal combustion engines provide low torques at low engine speeds, and, thus, it is necessary to amplify that torque making the engine working at higher speeds when the vehicle is at low speeds, e.g. during a launch from standstill. From an equivalent point of view, the transmission allows to amplify the engine torque transferred to the wheels when higher accelerations are needed. For such reasons every type of transmission has some devices that allow to select the ratio between its input shaft angular speed (engine side) and its output shaft angular speed (the side towards the wheels). The transmission's input shaft is connected to the *flywheel* of the engine, while the output shaft of the transmission is connected to the *final drive* (that contains the *differentials*) through the *drive shaft* (In British English it is used also the term propeller shaft when dealing with a rear-wheel-driven vehicle.). Even though such subsystems

are differently located depending on the vehicle layout (if front-wheel or rear-wheel driven or even all-wheel driven), they all are parts of the powertrain and determine its behaviour.

Transmissions can be viewed also as systems that allow to transfer power from the engine to the wheels in a smooth and efficient way. In order to achieve such basic and fundamental objectives, as well as to improve fuel economy, performance and drivability, many technologies have been introduced into the market of automotive transmissions.

Types of Transmissions

In **manual transmissions** (MT), a set of gears provides each the different speed conversion ratios, and any gear can be selected by the driver by acting on the shift lever. For interrupting the power flow during the gear selection, a *clutch* is requested that disconnects the transmission from the flywheel of the engine and reconnects it just after the selection of the new gear. All such operations are performed manually by the driver.

Automatic transmissions (AT) are the other well-known type of automotive transmissions. Hydraulic ATs do not have the clutch for connecting the transmission to the flywheel; instead they have a *torque converter* that provides the fluid coupling between the transmission and the engine realising both a damping of the powertrain vibrations and a torque multiplication. Moreover in ATs a *planetary gear set* allows to get the different gear ratios. The driver selects only the operation mode, and the selection of the gear is implemented through the electronic control. The main limits of these transmissions are the low efficiency (particularly due to the slip of the torque converter), a larger space requirement and a higher weight. The new generations of automatic transmissions have reduced such disadvantages thanks to the use of lightweight materials, more shifting steps and, above all, the replacement of conventional hydraulic components by electronic and electrohydraulic counterparts.

Indeed the electronic transmission control allows not only to improve the fuel economy,

performances and drivability, but it also gives so much flexibility and new possibilities (e.g. diagnostics, fault detection, integration with other subsystems) that overcome the intrinsic disadvantages like complexity and development cost (Deur et al. 2006). Electronic transmission control played a fundamental role also for the introduction of new technologies that could have been exploited thanks to the automatic control. Thus in recent years continuously variable transmissions, automated manual transmissions, dual clutch transmissions and electrically variable transmissions have appeared in the market, which was traditionally dominated by ATs and MTs (Sun and Hebbale 2005).

An **automated manual transmission** (AMT) system can be viewed as an MT with some controlled actuators as add-ons: it still has a (dry or wet) clutch assembly and a multi-speed gearbox, both of which are equipped with electromechanical or electro-hydraulic actuators which are commanded by an electronic control unit. In AMTs the gear shift can be decided automatically by the transmission control unit (TCU) or even manually by the driver. In both cases after the gear shift command, the TCU manages all the shifting steps, through suitable signals sent to the engine, the clutch assembly and the gearbox. This technology has the advantages of lower weight, lower costs and higher efficiency with respect to ATs.

It is worth to highlight that one of the AMT's limitations could be the driving comfort reduction, caused by lack of traction during gear shift actuation. Indeed the torque interruption leads to perceived jerks due to vehicle acceleration discontinuity and is highly different with respect to the smoother conventional automatic transmissions with torque converter (Lucente et al. 2007).

Automated manual transmissions have become popular in Europe for their higher performances with respect to MTs and for the lower cost compared to ATs. In North America, instead, their use is limited because of the torque interruption during shifts that causes some discomfort.

An offshoot of the AMT is the **dual clutch transmission** (DCT), in which the gearbox assembly has two separate and independent

clutches, one for odd gears and one for even gears (Goetz et al. 2005). In a DCT, shifts can be achieved without sensible torque gap, by applying the engine torque to one clutch while the engine torque is being disconnected from the other clutch. The result is gentle, jerk-free gear shifts with the same comfortable driving of an automatic transmission combined with the efficiency and the performance of an economic manual transmission. In both DCT and AMT, electronic control (in particular aimed to solve the clutch engagement control problem) is the key to ensure a smooth torque transfer.

As a further transmission technology, the **continuously variable transmission** (CVT) enables the engine to operate in a wide range of speed and load conditions independently from the speed and the torque requests of the vehicle. A modern CVT system of the mechanical belt/chain type with pulley driven consists of a steel belt that runs between two variable-width pulleys. The distance between pulley cones can be varied to change the gear ratio between shafts, thus generating an infinite number of "gears". A CVT is less efficient than a standard discrete AT due to the losses in the belt-pulley system, but it can improve the fuel consumption making the engine work at better operating conditions. Somehow related to CVT are the **electrically variable transmissions** (EVT) that appeared in the market of hybrid electric vehicles recently and use electric machines, namely, motors/generators, with planetary gear sets so to enable the function of CVTs with flexibility, controllability and better performance. Such type of transmissions is usually part of hybrid electric vehicles.

By looking at the different types of automotive transmissions, it is possible to classify electronically controlled transmissions into two main groups: discrete ratio and continuously variable transmissions. The first group deals with the problems of automating the shift scheduling ("when to") or also controlling the shift execution ("how to") (Hrovat and Powers 1988). The latter group deals with control problems that live in a continuous domain (like classical "process control") and that allow to design a simpler control software. Indeed the discrete ratio

transmissions are more complex to control since they determine many large transients of short duration due to gear shifts and moreover their intrinsic mixed discrete-continuous nature gives rise to dynamic systems in which continuous time dynamics interact with discrete event dynamics. In other words such class of controlled transmissions can be considered a significant application of what control theory calls **hybrid systems** (Goebel et al. 2009). That makes the transmission control a very interesting and challenging control engineering problem.

Control Problems for Automotive Transmissions

Electronic control applied to automotive transmissions allows to improve the efficiency and the fuel economy, to obtain better shift quality and comfort, to implement flexible driving. In order to achieve such objectives, different approaches can be used for designing suitable control laws, and a hierarchical approach is often required for dealing with the complexity of the several problems. Thus, recent produced cars together with the engine control unit have also a TCU that manages all transmission operations and sends command signals to actuators in order to perform the desired behaviour. Some dedicated devices are then available for tackling challenging control problems and for trying to solve them exploiting classical feedback and/or modern model-based control.

Low-Level Control

In electronically controlled transmissions, many hydraulic functions of conventional transmissions have to be replaced by electrohydraulic systems. Thus it is fundamental being able to control actuators in a suitable way. Usually classical PID regulators are employed for this type of low-level controls whose aim consists of regulating some variables to reference values computed by a higher-level controller. For instance, in AMTs, the concentric slave cylinder is controlled to make the clutch disc follow a

position reference signal computed by the TCU. The clutch position reference can be obtained by taking into account some models of the clutch transmission characteristic (Vasca et al. 2011), thus realising a feedforward/feedback architecture.

Other examples of low-level controls in automotive transmissions are related to the clutch fill process (Song et al. 2010) in ATs, or the line pressure control, or also the CVT belt load control.

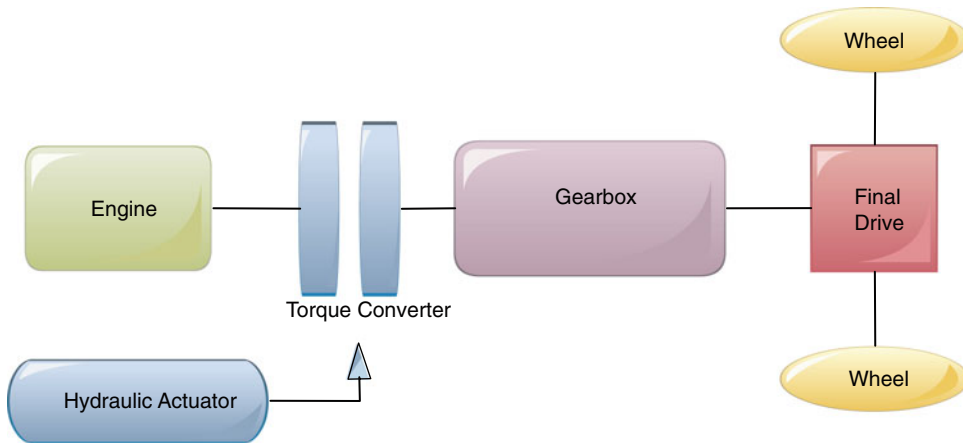
Calibration Process

Most of the industrial control strategies applied to automotive transmissions are based on feedforward/feedback architectures. Feedforward control typically relies on detailed models of the transmission that quite often consists of some look-up tables rather than specific physical models. Look-up tables are used also for implementing adaptive feedback controllers, and thus tables with calibrated variables are widely spread for automotive transmission control. With the increasing number of required functionalities, the calibration process for transmission control subsystems becomes more and more complex. Then it becomes important to investigate automated and systematic approaches for the calibration process in order to improve the reliability and performances and, above all, for diminishing the development time.

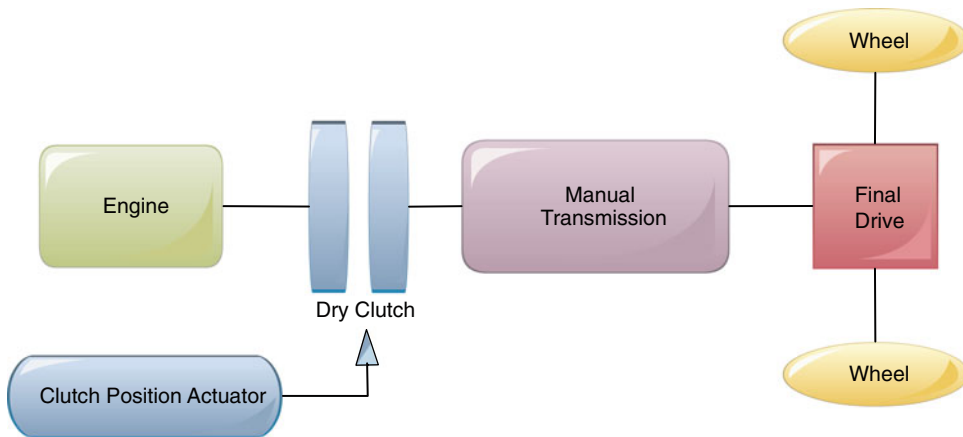
Of course, a different approach that looks at developing model-based control strategies could be the way to reduce the number of calibration variables and, thus, the calibration effort and time. The main obstacle to that is the uncertain environment that makes very challenging to find robust control solutions.

Gear Shifting

The gear shift execution is a common problem in all discrete ratio transmission. In Figs. 1 and 2, the schemes of two transmission architectures are reported. Although we are looking at completely different typologies like ATs and AMTs, the gear shift problem is almost the same from an abstract point of view: commanding the actuator



Transmissions, Fig. 1 Architecture of an automatic transmission



Transmissions, Fig. 2 Architecture of an automated manual transmission

for getting a desired torque at the primary shaft of the transmission in order to have the possibility of disengaging the old gear, engaging the neutral gear and then engaging the new gear, without shuffles and limiting the jerk experienced by driver. In ATs the basic idea is to control the hydraulic pressures of the torque converter to transfer smoothly the power from the engine to the driveline, while minimising the torque disturbance at the output shaft. Analogously in AMTs with dry clutch, the actuator is commanded for positioning the clutch disc towards the flywheel, exerting a pressure that is transformed into the transmitted torque (Della Gatta et al. 2018).

For instance, a wet clutch of an automatic transmission gives the following transmitted torque (Deur et al. 2006)

$$T = n A_p p_{app} \mu(\omega, p_{app}, \theta) r_e \operatorname{sgn}(\omega)$$

where n is the number of friction surfaces, A_p is the piston area, p_{app} is the hydraulic pressure, μ is the friction coefficient (depending also on the clutch fluid temperature θ), r_e is the equivalent radius of the clutch and ω is the clutch slip speed, i.e. the difference between the engine speed and the speed of the input shaft of the transmission.

For a dry clutch of an automated manual transmission (Vasca et al. 2011)

$$T = nF_{pp}(x_{to})\mu(\omega, \theta)r_e \operatorname{sgn}(\omega)$$

where F_{pp} is the force exerted by the cushion spring depending on the clutch actuator position x_{to} and θ is the clutch disc temperature.

In both cases the actuator allows to regulate the torque transmitted, respectively, through the torque converter or the dry clutch under a slipping condition. The main problem is that in modern transmissions there are no low-cost torque sensors, and thus, due also to model uncertainties and much different operating conditions, it is not possible to regulate the transmitted torque through a closed-loop scheme. What is usually done is to control the engine speed and/or the speed of the input shaft of the transmission. Quite often their difference (the slip speed) is the variable to be controlled.

Many different approaches have been proposed in literature for solving such control problems that can be formulated as simple as a regulation problem of the slip speed or even as a more complex multivariable control problem that considers the engine and clutch torques as control variables and the slip speed and vehicle speed as controlled variables, possibly solving the problem through robust control tools. The problem can be formulated quite naturally also as an optimal control problem that aims to minimise the engagement time, the driveline oscillations or the dissipated energy. For example, by defining the time derivative of the clutch torque as one control variable, the transmitted torque becomes a state variable, and the energy dissipated during the engagement phase can be expressed as the cross product of two state variables (Garofalo et al. 2002)

$$E_d = \int_0^{\bar{t}} \omega(s)T(s)ds,$$

and the clutch engagement can be expressed as an optimal control problem with free final time (the engagement time, $\bar{t}$) and a final state constraint (i.e. $\omega(\bar{t}) = 0$).

Some authors have also proposed a different solution for the gear shifting problem by acting through the engine control (Pettersson and Nielsen 2000). The idea is to control the gear shifting using directly the engine as the actuator that allows to modulate the transferred torque to the transmission (see Figs. 1 and 2). In particular the engine is controlled so as to get a zero transferred torque in the transmission and then the neutral gear is engaged. In this way a virtual clutch is realised.

Driveline Modelling

When model-based control is used for automotive transmissions, it is important to have a good model of the driveline that is detailed enough for capturing the main dynamics and, at the same time, sufficiently simple to deal with for designing not so complex controllers. Vehicular drivelines have many elastic parts making mechanical resonances occurring. Handling such resonances is important for driveability but also for reducing mechanical stresses. Thus driveline control is crucial not only during gear shifting but also for a more general powertrain control that could manage wheel-speed oscillations induced by sudden accelerations or following from the road roughness.

Integrated Powertrain Management

The shift scheduling is a further interesting problem of electronically controlled transmissions. The shift point is usually based on some measurements like the vehicle speed, the maximum acceleration or throttle angle. In this case the control strategy is open loop and implemented through look-up tables.

As the number of gears increases, the shift schedule gets more complicated. Thus it becomes important to take into account also the actual driving scenario. For example, entering a curve during an uphill driving is quite different with respect to a downhill driving situation, so it can be very useful getting information on the steering angle, the road grade, vehicle acceleration, etc. More information, together with new degrees

of freedom that are available to modern vehicles (e.g. vehicles with electronic throttle control), give the opportunity to realise an integrated powertrain control which coordinates the engine control and the transmission control allowing to manage the gear scheduling and the gear shifting execution in a more flexible way, trying to optimise the fuel consumption and to improve the driveability (Kim et al. 2007).

Diagnostics

In all automotive applications, the safety is a fundamental issue that becomes more and more critical when the number of subsystems and their interaction increase, as it happens when introducing electronic control. Thus diagnosing faults of control systems is a real challenging problem, in particular when there is few information richness. To this aim systems and control theory can be very useful for designing observer or fault detection algorithms that could deal with those types of problems.

Summary and Future Directions

In summary transmission control is a fertile application for looking at challenging problems of much interest both for control scientists and engineers, giving the opportunity for investigating topics like optimal control (e.g. for gear shifting and integrated powertrain control), robust control (for driveline modelling and control), estimation (diagnostics) and adaptive and predictive control.

Technological developments could affect the possibilities and the effectiveness of the transmission control. Of course new sensing devices can improve the reliability and the precision of the feedback, but they can also open the door to new control architectures. For instance the phenomenon of inverse magnetostriction that converts material strain into magnetic property changes can be exploited to measure transmitted torque and some magnetoelastic torque sensors have been investigated by a number of researchers (Pietron et al. 2013;

Klimartin 2003). In this way the gear shifting control problem can be attacked by closing the loop on the transmitted torque measurement, avoiding more or less complex torque observers or, at least, improving the control performances.

Analogous considerations can be carried out at the actuation level. For instance Kim et al. (2020), have proposed a new clutch actuator with self-energising mechanism so to reduce power consumed by engagement and improve the controllability. Smart material-based actuation devices were also developed by a number of researchers in recent years (Chaudhuri and Wereley 2012), and some specific applications to automotive transmissions are currently under investigation, like magnetorheological fluid dual clutch transmissions that were discussed in Zhang et al. (2019), in particular for electric vehicle transmissions.

At the system level, one of the most interesting research direction deals with the communication of different controlled subsystems like the engine control and the transmission control, with the final aim of integrating such subsystems for an integrated powertrain management. That becomes even more challenging when considering autonomous hybrid electric vehicles (Ghasemi and Song 2019).

Cross-References

- [Engine Control](#)
- [Powertrain Control for Hybrid-Electric and Electric Vehicles](#)

Recommended Reading

Hrovat et al. (2010) give an overview of automotive transmissions in their chapter on powertrain control of the CRC Control Handbook. Of course transmission control is discussed also in classical automotive control books: one of the first well-known books dealing with automotive control was Kiencke and Nielsen (2005). There a whole chapter on driveline control is dedicated to driveline modelling and gear shifting for

clutch-based transmissions. More recent books on automotive transmissions are Fischer et al. (2015) and Zhang and Mi (2018). Both of them consider control system design for different types of transmissions.

In the scientific literature many papers deal with transmission control: here in particular we would like to cite the optimal control approach for ATs by Haj-Fraj and Pfeiffer (2001), a deep discussion on AMT control in Glielmo et al. (2006) and papers on DCTs like Kulkarni et al. (2007) and Senatore (2009); in the latter one, the author illustrated the wide selection of patents on dual clutch showing how they replaced classical AMTs on the market.

Regarding automotive technologies, an overview of automotive sensors can be found in Fleming (2008), while a discussion on smart materials and the integration of mechanics, materials and electronics (the so-called mechatronics discipline) has been presented by Munhoz et al. (2007).

Bibliography

- Chaudhuri A, Wereley N (2012) Compact hybrid electro-hydraulic actuators using smart materials: a review. *J Intell Mater Syst Struct* 23(6):597–634
- Della Gatta A, Iannelli L, Pisaturo M, Senatore A, Vasca F (2018) A survey on modeling and engagement control for automotive dry clutch. *Mechatronics* 55:63–75
- Deur J, Petric J, Asgari J, Hrovat D (2006) Recent advances in control-oriented modeling of automotive power train dynamics. *IEEE/ASME Trans Mechatron* 11(5):513–523
- Fischer R, Küçükay F, Jürgens G, Najork R, Pollak B (2015) *The automotive transmission book*. Springer International Publishing, Cham
- Fleming WJ (2008) New automotive sensors—a review. *IEEE Sens J* 8(11):1900–1921
- Garofalo F, Glielmo L, Iannelli L, Vasca F (2002) Optimal tracking for automotive dry clutch engagement. In: 15th IFAC World Congress, pp 367–372
- Ghasemi M, Song X (2019) Powertrain energy management for autonomous hybrid electric vehicles with flexible driveline power demand. *IEEE Trans Control Syst Technol* 27(5):2229–2236
- Glielmo L, Iannelli L, Vacca V, Vasca F (2006) Gearshift control for automated manual transmissions. *IEEE/ASME Trans Mechatron* 11(1):17–26
- Goebel R, Sanfelice RG, Teel AR (2009) Hybrid dynamical systems. *IEEE Control Syst Mag* 29(2):28–93
- Goetz M, Levesley M, Crolla D (2005) Dynamics and control of gearshifts on twin-clutch transmissions. *Proc Inst Mech Eng Part D J Automob Eng* 219(8):951–963
- Haj-Fraj A, Pfeiffer F (2001) Optimal control of gear shift operations in automatic transmissions. *J Frankl Inst* 338:371–390
- Hrovat D, Powers W (1988) Computer control systems for automotive power trains. *IEEE Control Syst Mag* 8(4):3–10
- Hrovat D, Jankovic M, Kolmanovsky I, Magner S, Yanakiev D (2010) Powertrain control. In: *The control handbook*, vol 2. Control applications, 2nd edn. CRC Press, Boca Raton
- Kiencke U, Nielsen L (2005) *Automotive control systems. For engine, driveline, and vehicle*. Springer, Berlin/Heidelberg
- Kim D, Peng H, Bai S, Maguire JM (2007) Control of integrated powertrain with electronic throttle and automatic transmission. *IEEE Trans Control Syst Technol* 15(3):474–482
- Kim D, Kim J, Choi S (2020) Design and modeling of energy efficient dual clutch transmission with ball-ramp self-energizing mechanism. *IEEE Trans Veh Technol* 69(3):2525–2536
- Klimartin B (2003) Magnetoelastic torque sensor utilizing a thermal sprayed sense-element for automotive transmission applications. SAE Technical Paper 2003-01-0711
- Kulkarni M, Shim T, Zhang Y (2007) Shift dynamics and control of dual-clutch transmissions. *Mech Mach Theory* 42(2):168–182
- Lucente G, Montanari M, Rossi C (2007) Modelling of an automated manual transmission system. *Mechatronics* 17(2–3):73–91
- Munhoz D, Gregolin J, de Faria L, de Andrade T (2007) Automotive materials: current status, technology trends and challenges. SAE Technical Paper 2007-01-2671
- Pettersson M, Nielsen L (2000) Gear shifting by engine control. *IEEE Trans Control Syst Technol* 8(3):495–507
- Pietron G, Fujii Y, Kucharski J, Yanakiev D, Kapas N, Hermann S, Hogirala R, Green T (2013) Development of magneto-elastic torque sensor for automatic transmission applications. *SAE Int J Passengers Cars Mech Syst* 6(2):529–534
- Senatore A (2009) Advances in the automotive systems: an overview of dual-clutch transmissions. *Recent Patents Mech Eng* 2(2):93–101
- Song X, Zulkefli MAM, Sun Z (2010) Automotive transmission clutch fill optimal control: an experimental investigation. In: 2010 American Control Conference. IEEE, pp 2748–2753
- Sun Z, Hebbale K (2005) Challenges and opportunities in automotive transmission control. In: *Proceedings of the American Control Conference*, pp 3284–3289
- Vasca F, Iannelli L, Senatore A, Reale G (2011) Torque transmissibility assessment for automotive dry-clutch engagement. *IEEE/ASME Trans Mechatron* 16(3):564–573

- Zhang Y, Mi C (2018) Automotive power transmission systems. Wiley, Hoboken
- Zhang H, Du H, Sun S, Li W, Wang Y (2019) Design and analysis of a novel magnetorheological fluid dual clutch for electric vehicle transmission. In: Automotive Technical Papers, SAE International

Trust Models for Human-Robot Interaction and Control

Yue Wang
Department of Mechanical Engineering,
Clemson University, Clemson, SC, USA

Abstract

Trust is the key determinant of human's acceptance and hence willingness to utilize robots. Improper levels of trust may lead to performance degradation, human overload, and even catastrophic consequences. Therefore, in human-robot interaction (HRI) and robot control with humans-in-the-loop, it is crucial to quantify and analyze trust so as to develop robot motion plans, control algorithms, and autonomous decision aids to enable better human-robot collaboration (HRC). This entry provides an overview of the state-of-the-art trust models. In particular, the focus here is on computational trust models that quantify human trust in robots and capture its dynamic evolution. Four categories of computational trust models are introduced and their respective pros and cons summarized and compared. The utilization of computational trust models in robot motion planning, control, and decision-making is reviewed. Future research directions in trust modeling in HRI and control are also discussed.

Keywords

Human-robot interaction · Computational trust models · Trust-based robot control

Introduction

Trust is a major factor that influences human's acceptance and hence utilization of robots (Hancock et al. 2011; Wagner 2009; Desai 2012). Trusting a robot means that an individual believes that the robot will help to achieve his/her goal and hence is willing to delegate tasks to it in the presence of uncertainties and risks. However, if the robot does not perform the task well but the human still decides to use it, this is a misuse situation due to overtrust, which might lead to overall system performance degradation and even safety violation. For example, several Tesla crashes occurred because the drivers trusted the autopilot to do all of their driving, even though it can't. On the other hand, if the human does not use the robot even if it can perform the task well, this is a disuse situation due to undertrust, which might lead to human overload or even fatal consequences. For example, a trainee pilot and her flight instructor died in a 2010 plane crash because the instructor told the trainee to trust his own calculations over the instrument panel, causing the plane to run out of fuel. By properly understanding and analyzing human trust in robots, robot motion plans and feedback control algorithms can be developed to change robot behaviors and help humans calibrate their trust to the robot to avoid either overtrust or undertrust. This will enable full exploitation of the robot potential and ensure system safety while minimizing human workload. The need spans a wide range of applications, including teleoperated robots in remote/dangerous environment, co-robots working together with humans in factory floors, robotic wheelchairs for disabled people, (semi)autonomous vehicles with humans-in-the-loop, etc.

A significant amount of research effort has been devoted to human factors engineering and human-robot interaction (HRI) to study human trust in automation (Lee and Moray 1992) and more recently human trust in robots (Wagner 2009; Desai 2012). Despite the similarity shared by human-robot trust and human-automation trust, robots are different from automation in their capabilities in sensing and thinking. Humans

perceive robots as teammates instead of simple tools, and hence their trust-impacting factors vary. A meta-analysis conducted by Hancock et al. showed that there are three major categories of factors impacting human-robot trust: robot-related factors, environmental factors, and human-related factors (Hancock et al. 2011). Robot-related factors include robot performance (e.g., reliability, predictability) and attribute (e.g., appearance, robot type). Environmental factors include team collaboration (e.g., culture, shared mental models) and task-based factors (e.g., task type, difficulty). Human-related factors include ability-based (e.g., expertise, perceived workload) and human characteristic factors (e.g., demographics, propensity). Among these, robot-related factors have the strongest impact on trust. Furthermore, human trust in robots has a dynamic nature and is influenced by these factors throughout the entire HRI process.

In the extant literature, the most common approach for trust measurement is subjective ratings through questionnaires (Jian et al. 2000). These measures yield important data for hypothesis testing and statistical analysis. However, such approaches fail to provide quantified trust values in real time and their dynamics, which are essential for system stability and optimality analysis, as well as robot feedback control design. Although some researchers have started to use psychophysical measures such as the electroencephalography (EEG) signals to measure real-time trust, these measures are often noisy, and the result is usually binary, i.e., trust or not (Lee 2018). Even fewer works consider human trust in multi-robot systems (MRS) and swarm robotics (Nam et al. 2019). Motivated by the need to quantify, model, calibrate, and predict human trust in robots for real-time HRI, this entry will provide a brief review of computational trust models and examination of their respective pros and cons.

Computational Trust Models

The models discussed in this entry are by no means comprehensive, but the most representa-

tive ones from the literature are introduced. All model parameters are learned based on data collected from human subject tests instead of pure mathematical models; thus these models tend to capture actual human feelings and behaviors. They are put into four categories: time-series trust models, Bayesian-based trust models, Markov Decision Process (MDP)-based trust models, and mixed Bayesian and MDP trust models. Each category of models is explained, and their pros and cons are discussed.

Time-Series Trust Models

Lee and Moray developed an autoregressive moving average vector form (ARMAV) of time-series trust model to capture the dynamic evolution of human trust in machines (Lee and Moray 1992). Two factors were found to have the highest influence on human trust in machines, i.e., the occurrence of a fault and system performance. A simulated semiautomatic juice pasteurization plant was used as a test bed. The model was fitted to data collected from all human subjects. Inspired by this model and Hancock et al.'s meta-analysis (Hancock et al. 2011), Sadrfaridpour et al. proposed a personalized time-series human trust in robot model by further considering human performance (Sadrfaridpour et al. 2016):

$$\begin{aligned} T(k) = & A_0 T(k-1) + B_1 P_R(k) \\ & + B_2 P_R(k-1) + C_1 P_H(k) \\ & + C_2 P_H(k-1) + D_1 F(k) \\ & + D_2 F(k-1), \end{aligned} \quad (1)$$

where $T(k)$ and $T(k-1)$ represent normalized current and prior trust and P_R , P_H , and F are robot performance, human performance, and faults, respectively. Here, the coefficients A_0 , B_1 , B_2 , C_1 , C_2 , D_1 , and D_2 are constants, which were obtained from human subject tests and tuned for different individuals. The initial trust $T(0)$ can represent dispositional trust that reflects trust initially formed before any interaction has yet taken place (Merritt and Ilgen 2008). Note that robot and human performances are specific to the task and the human individual performing the

task. In the collaborative manufacturing assembly task considered in Sadrifaridpour et al. (2016), the robot performance is evaluated by its flexibility in accommodating human behaviors and is modeled as the difference between human and robot speeds. The human performance model accounts for the muscle fatigue and recovery dynamics that capture the fatigue level of a human body when performing repetitive kinesthetic tasks, which are typical types of human motions in manufacturing. Similar time-series trust models were adopted in Wang et al. (2015), Spencer et al. (2016) and Sadrifaridpour and Wang (2017). Saeidi et al. further extended the model to a continuous-time version (Saeidi et al. 2017).

The time-series models enable a multidimensional construct of trust and provide a causal explanation of the impacting factors (i.e., robot-related factors, environmental factors, and human-related factors) with the level of trust. This category of trust models can capture the dynamic nature of trust (i.e., memory of trust) through the first-order lag model. Hence, the time-series trust models can be used to measure and predict trust values in real time. However,

since the models are linear and deterministic, they are sensitive to noisy data and have relatively lower goodness of fit with respect to the human subject data. There is also no unified framework for modeling either robot or human performance of different scenarios in the above works.

Bayesian-Based Trust Models

Xu and Dudek developed a probabilistic trust model, called OPTIMO, based on a dynamic Bayesian network (DBN) to estimate human trust in robots in near real time (Xu and Dudek 2015). In this study, an aerial robot for vision navigation tasks (i.e., boundary tracking) is considered. The robot has an autonomous visual boundary tracker, and the human can also intervene based on the live camera feed from it. The OPTIMO model treats an individual's trust as a hidden state. It then infers Bayesian beliefs over the trust state based on robot performance, observed human interventions in the robot tracking controller, as well as human subjects' responses to survey questions during the interaction. The personalized trust model is computed iteratively as follows:

$$\begin{aligned}
 bel_f(t_k) &= Prob(t_k | p_{1:k}, i_{1:k}, e_{1:k}, c_{1:k}, f_{1:k}, t_0) = \frac{\int \overline{bel}(t_k, t_{k-1}) dt_{k-1}}{\int \int \overline{bel}(t_k, t_{k-1}) dt_{k-1} dt_k}, \\
 bel_s(t_{k-1}) &= Prob(t_k | p_{1:K}, i_{1:K}, e_{1:K}, c_{1:K}, f_{1:K}, t_0) = \int \frac{\overline{bel}(t_k, t_{k-1})}{\int \overline{bel}(t_k, t_{k-1}) dt_{k-1}} bel_s(t_k) dt_k, \\
 \overline{bel}(t_k, t_{k-1}) &:= \mathcal{O}_i(t_k, t_{k-1}, i_k, e_k) \cdot \mathcal{O}_c(t_k, t_{k-1}, c_k) \cdot \mathcal{O}_f(t_k, f_k) \cdot \\
 &\quad \mathcal{P}(t_k, t_{k-1}, p_k, p_{k-1}) \cdot bel_f(t_{k-1}).
 \end{aligned} \tag{2}$$

The model quantifies the Bayesian trust belief within a sequence of K nonoverlapping time windows, where each time window $k=1:K$ lasts for W seconds. Here, $bel_f(t_k)$ is the filtered trust belief of the user's degree of trust $t_k \in [0, 1]$ at the k -th time window, given prior data $p_{1:k}, i_{1:k}, e_{1:k}, c_{1:k}, f_{1:k}$, and t_0 . The term $bel_s(t_{k-1})$ is the smoothed trust belief of the user's degree of trust t_{k-1} at the $(k-1)$ -th time window, given a previously recorded dataset $p_{1:K}, i_{1:K}, e_{1:K}, c_{1:K}, f_{1:K}$, and t_0 . More specifically, $p_k \in (0, 1)$ is the aggregated

robot performance at the k -th time window, with 0(1) representing the lowest (highest) performance. $i_k \in \{0, 1\}$ represents human intervention by switching from autonomous control mode to manual control mode, where 0 represents no intervention and 1 represents intervention. $e_k \in \{0, 1\}$ is an extrinsic factor reflecting the change of task target irrespective of trust. $c_k = \{-1, 0, +1, \emptyset\}$ is the user's reported change of trust where -1 , 0 , and 1 represent that trust has increased, remained the same, or decreased, respectively. The symbol $\emptyset$

represents no human response within the k -th time window. $f_k \in \{[0, 1], \emptyset\}$ is the user's direct trust feedback with 0(1) representing the lowest (highest) trust feedback. The term $\mathcal{O}_i$ represents the conditional probability distribution (CPD) of human intervention i_k given the current trust t_k , the change in trust $t_k - t_{k-1}$, and the extrinsic factor e_k . The term $\mathcal{O}_c$ represents the CPD of a user reported trust change c_k given change of the hidden trust state $t_k - t_{k-1}$. The term $\mathcal{O}_f$ represents the CPD of direct trust feedback f_k given the current trust state t_k . The term $\mathcal{P}$ represents the CPD of being in trust state t_k given prior trust t_{k-1} and the robot's prior and current task performances p_{k-1} and p_k . The expectation maximization (EM) algorithm is used to obtain the optimal parameters associated with these CPDs using data collected from human subject tests. A similar DBN-based trust model was adopted in Wang et al. (2018) by further considering human performance and faults as trust-impacting factors.

The Bayesian-based models also capture the dynamic evolution of trust and provide casual relationship with the trust-impacting factors. Compared to the time-series trust models, the DBN-based trust models are probabilistic and take into account the uncertainties in HRI. Furthermore, they combine human's intervention actions with actual responses/feelings as evidence. Due to the latent trust state, the EM algorithm is utilized to train the model. However, the resulting maximum likelihood solution relies on initial guess of the model parameters, which are challenging to obtain. On the other hand, higher likelihood is not always better due to overfitting.

Markov Decision Process (MDP)-Based Trust Models

Nam et al. (2019) developed MDP-based trust models in supervisory control of swarm robots with different levels of autonomy (LoAs) in a target foraging task. The study showed that swarm performance (i.e., number of targets found) may not be clearly perceivable by the human, but instead the physical appearance of the swarm (i.e., swarm's heading variance and

convex hull area) has more impact on human trust. A trust model with user intervention is modeled as an MDP whose state space includes factors that impact trust: $(S, A, R, \delta, \gamma)$, where $S = \{s_k = (hv_k, cv_k, i_k, ts_k) | hv_k \in \mathbb{R}^{\geq 0}, cv_k \in \mathbb{R}^+, i_k \in \{0, 1\}, ts_k \in [0, 1], k = 1, 2, \dots, K\}$ denotes the finite set of states, hv_k is the heading variance of the swarm, cv_k is the convex hull area, i_k indicates if an intervention command is issued or not, and ts_k is the user's subjective trust at time k ; A denotes the finite set of actions consisting of trust levels that an operator can choose; $R(s_k, a, s_{k+1}) = \sum_{j=1}^n \lambda_j \phi_j$ is a reward function defined as a weighted linear sum of features $\phi_j, j = 1, 2, \dots, n$; $\delta(s_k, a, s_{k+1}) = Prob(s_{k+1} | s_k, a)$ is a state transition probability determined based on the transition frequencies in the training set; and $\gamma \in [0, 1]$ is the discount factor. Variations of the MDP trust models were also developed with varied states s_k . Because significant individual difference was found in the experiments, an inverse reinforcement learning (IRL) algorithm was developed to learn the personalized reward function in the MDP trust model from user demonstration.

The MDP trust model is also dynamic and probabilistic and can predict trust in near real time. Furthermore, it was shown to improve prediction accuracy as compared to the DBN trust model (Nam et al. 2019). However, the explicit relation between trust and swarm parameters remains unknown in the MDP structure. A larger size state space will also make it difficult to perform online prediction. Also note that both the time-series trust $T(k)$ and the MDP-based trust ts_k are user-reported variables. They correspond to the direct trust feedback f_k instead of the hidden state t_k in the DBN model. In human subject tests, response biases are prevalent and often lead to inaccurate or false responses.

Mixed Bayesian and MDP Trust Models

All the above models seek to quantify and predict trust in real time, but none of them consider the calibration of trust to adjust the LoAs. Chen et al. (2018) introduced a trust model by integrating trust into robot decision-making, which gives insight into trust calibration for the proper

use of robot autonomy. The trust value $\theta_{k+1} \sim pdf(\theta_{k+1}|\theta_k, p_{k+1})$ is modeled as a linear Gaussian distribution $\mathcal{N}(\alpha_{p_{k+1}}\theta_k + \beta_{p_{k+1}}, \sigma_{p_{k+1}})$. Here, $p_{k+1} = performance(x_{k+1}, x_k, a_k^R, a_k^H)$ is the robot performance, where $x_k, x_{k+1} \in X$ are the world states, $a_k^R \in A^R$ represents robot action, and $a_k^H \in A^H$ represents human action. The unknown parameters $\alpha_{p_{k+1}}, \beta_{p_{k+1}}$, and $\sigma_{p_{k+1}}$ can be estimated with Bayesian inference on the model through Hamiltonian Monte Carlo sampling using the Stan probabilistic programming language.

The human-robot team is formalized as an MDP $(X, A^R, A^H, R, \delta, \gamma)$, where the probability of transition function δ is $Prob(x_{k+1}|x_k, a_k^R, a_k^H)$, and $r(x_k, a_k^R, a_k^H, x_{k+1}) \in R$ is the real-value reward. The history of interaction between robot and human until time step k is $h_k = \{x_0, a_0^R, a_0^H, x_1, r_1, \dots, x_{k-1}, a_{k-1}^R, a_{k-1}^H, x_k, r_k\}$. Denote the robot policy as π^R and the human policy as π^H , the expected total discounted reward of starting at a state x_0 and following robot and human policy is

$$v(x_0|\pi^R, \pi^H) = \mathbb{E}_{a_k^R \sim \pi^R, a_k^H \sim \pi^H} \sum_{k=0}^{\infty} \gamma^k r(x_k, a_k^R, a_k^H). \quad (3)$$

The optimal robot policy can be obtained by $\pi_*^R = \operatorname{argmax}_{\pi^R} \mathbb{E}_{\pi^H} v(x_0|\pi^R, \pi^H)$. Since the history h_k is long, it may be difficult to optimize the policy. It was envisioned that trust is a compact approximation of h_k , and hence the following approximation is used: $\pi^H(a_k^H|x_k, a_k^R, h_k) = \pi^H(a_k^H|x_k, a_k^R, \theta_k)$.

Then, a so-called trust-partially observable MDP (trust-POMDP) model is built by augmenting the state space as $s_k = (x_k, \theta_k)$, where x_k is fully observable and θ_k is only partially observable. The trust dynamics $pdf(\theta_{k+1}|\theta_k, p_{k+1})$ and human behavioral policy $\pi^H(a_k^H|x_k, a_k^R, \theta_k)$ are embedded in the transition dynamics of trust-POMDP. This enables the calibration of trust to yield human behaviors and consequent robot policy to maximize the team performance. However, the trust model only considers the influence of robot performance. The other

potential impacting factors of trust were not considered.

Summary, Discussion, and Future Research Directions

This entry provides an overview of computational trust models in HRI systems, namely, the time-series, Bayesian-based, MDP-based, and mixed Bayesian and MDP trust models. Computational trust models offer a tool to quantify human trust in robots, analyze trust dynamics, and predict trust in near real time. Trust is multidimensional and specific to the robot itself, the task environment, and the human individual interacting with the robot. Hence, all above trust models are personalized. The time-series models are based on causal reasoning of trust-impacting factors. However, the models may be over simplified and are not robust to noisy data. Both the Bayesian-based and MDP-based models offer a probabilistic alternative factoring in HRI uncertainties for trust modeling. The DBN trust model further incorporates HRI experiences as evidence. On one hand, the prediction accuracy of DBN models is not as good as the MDP models due to limits of model structure and data type. On the other hand, the MDP model fails to reveal explicit relation between trust and impacting factors due to its state formulation. It also suffers from computational inefficiency given a larger state space. Both the time-series and MDP-based trust are subjective values obtained from surveys, which are not necessarily what humans actually mean due to response biases. Furthermore, the mixed trust models integrate Bayesian trust dynamics into a MDP human-robot team model so that the human trust can be calibrated to maximize the team performance.

The computational trust models enable design of robot feedback control, motion plans, and autonomous decision aids that improve HRI and human-robot team performance. For example, Saeidi et al. utilized real-time trust estimates to dynamically adjust the ratio between manual and autonomous control inputs in a shared control scheme for teleoperated mobile robots (Saeidi

et al. 2017). Sadrifaridpour and Wang selected the path of a robot manipulator and changed its end-effector velocity based on the trust level a human collaborator has of it (Sadrifaridpour and Wang 2017). More specifically, the time-series trust dynamics are used as a constraint in a model predictive control (MPC) framework to derive the optimal robot velocity. Chen et al. plan robot actions based on calibrated trust to maximize human-robot team performance (Chen et al. 2018). Trust has also been utilized as a metric for the decision-making between manual and autonomous motion planning modes (Wang et al. 2018), multi-robot task allocation (Spencer et al. 2016), as well as the real-time scheduling of human resources in multi-agent systems (MAS) (Wang et al. 2015). Beyond these existing works, real-time quantitative trust estimates, prediction, and calibration may also be helpful in selecting general LoAs in semiautonomous robotic systems and human-robot teaming. Stability and optimality analysis of HRI systems may also involve the trust dynamics. Furthermore, in undertrust scenarios, robot behaviors can be enhanced to improve task performance and trust. In overtrust scenarios, robot behaviors can be altered to test human's situational awareness and seek lower trust.

There are several limitations of extant works and directions for future research. First of all, uniformed performance models across different scenarios in the presented trust models are lacking. In addition, the existing models assume that the robot is honest and does not perform deceitful behaviors. Most importantly, trust is directly related to risk (Hancock et al. 2011), but existing works do not quantify risk in their models. It is therefore necessary to develop a general framework to include consideration of the associated risk and all types of robots and their respective performances. Secondly, although the DBN and MDP trust models capture dynamic evolution of trust and its transition, real-time calculation is usually challenging, and approximation algorithms for fast computation are much needed, especially for MRS. Thirdly, the psychophysical measurement data may be synergized with the computational trust models

for improved accuracy in trust estimation and prediction. Fourthly, trust models for single-human-multi-robot, multi-human-single-robot, and multi-human-multi-robot systems are of interest for effective human-robot teaming applications. There is a rich literature on Bayesian-based trustworthiness models for MAS (Teacy et al. 2006; Hu et al. 2016). Although these models do not account for HRI experiences and human subject reports, they may shed light on further development of scalable trust model for MRS. Last but not the least, although more data can help generate more accurate models, human subject tests are restricted in both time and number of participants. Hence, in most cases only a small amount of data can be obtained. Human simulations and real experiments with humans-in-the-loop may be combined to overcome this issue.

Cross-References

- [Adaptive Human Pilot Models for Aircraft Flight Control](#)
- [Physical Human-Robot Interaction](#)
- [Pilot-Vehicle System Modeling](#)

Bibliography

- Chen M, Nikolaidis S, Soh H, Hsu D, Srinivasa S (2018) Planning with trust for human-robot collaboration. In: Proceedings of the 2018 ACM/IEEE international conference on human-robot interaction, pp 307–315. ACM
- Desai M (2012) Modeling trust to improve human-robot interaction. Doctoral dissertation, University of Massachusetts Lowell
- Hancock PA, Billings DR, Schaefer KE, Chen JY, De Visser EJ, Parasuraman R (2011) A meta-analysis of factors affecting trust in human-robot interaction. *Hum Factors* 53(5):517–527
- Hu H, Lu R, Zhang Z, Shao J (2016) REPLACE: a reliable trust-based platoon service recommendation scheme in VANET. *IEEE Trans Veh Technol* 66(2):1786–1797
- Lee SY (2018) Impact of human like cues on human trust in machines: brain imaging and modeling studies for human-machine interactions. Final Report for AOARD Grant FA2386-14-1-4035
- Lee J, Moray N (1992) Trust, control strategies and allocation of function in human-machine systems. *Ergonomics* 35(10):1243–1270
- Merritt SM, Ilgen DR (2008) Not all trust is created equal: dispositional and history-based trust in

- human-automation interactions. *Hum Factors* 50(2): 194–210
- Nam C, Walker P, Li H, Lewis M, Sycara K (2019) Models of trust in human control of swarms with varied levels of autonomy. *IEEE Trans Hum Mach Syst* 1–11
- Jian JY, Bisantz AM, Drury CG (2000) Foundations for an empirically determined scale of trust in automated systems. *Int J Cogn Ergon* 4(1):53–71
- Sadrifaridpour B, Wang Y (2017) Collaborative assembly in hybrid manufacturing cells: an integrated framework for human-robot interaction. *IEEE Trans Autom Sci Eng* 15(3):1178–1192
- Sadrifaridpour B, Saeidi H, Burke J, Madathil K, Wang Y (2016) Modeling and control of trust in human-robot collaborative manufacturing. In: *Robust intelligence and trust in autonomous systems*, pp 115–141. Springer, Boston
- Saeidi H, Wagner JR, Wang Y (2017) A mixed-initiative haptic teleoperation strategy for mobile robotic systems based on bidirectional computational trust analysis. *IEEE Trans Robot* 33(6):1500–1507
- Spencer DA, Wang Y, Humphrey LR (2016) Trust-based human-robot interaction for multi-robot symbolic motion planning. In: *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 1443–1449. IEEE
- Teacy WL, Patel J, Jennings NR, Luck M (2006) Traros: trust and reputation in the context of inaccurate information sources. *Auton Agent Multi-Agent Syst* 12(2):183–198
- Wang X, Shi Z, Zhang F, Wang Y (2015) Dynamic real-time scheduling for human-agent collaboration systems based on mutual trust. *Cyber-Phys Syst* 1(2–4): 76–90
- Wang Y, Humphrey LR, Liao Z, Zheng H (2018) Trust-based multi-robot symbolic motion planning with a human-in-the-loop. *ACM Trans Interact Intell Syst* 8(4):31
- Wagner AR (2009) The role of trust and relationships in human-robot social interaction. Doctoral dissertation, Georgia Institute of Technology
- Xu A, Dudek G (2015) OPTIMO: online probabilistic trust inference model for asymmetric human-robot collaborations. In: *Proceedings of the tenth annual ACM/IEEE international conference on human-robot interaction*, pp 221–228. ACM

Uncertainty and Robustness in Dynamic Vision

Mario Sznaier and Octavia Camps
Electrical and Computer Engineering
Department, Northeastern University, Boston,
MA, USA

Abstract

Dynamic vision is a subfield of computer vision dealing explicitly with problems characterized by image features that evolve in time according to some underlying dynamics. Examples include sustained target tracking, activity classification from video sequences, and recovering 3D geometry from 2D video data. This article discusses the central role that systems theory can play in developing a robust dynamic vision framework, ultimately leading to vision-based systems with enhanced autonomy, capable of operating in stochastic, cluttered environments.

Keywords

Event detection · Multiframe tracking ·
Structure from motion

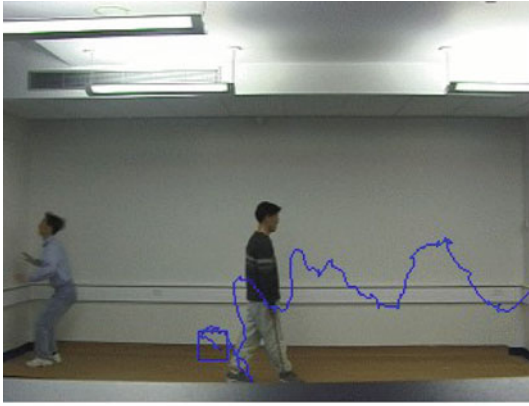
Background

In this article, we represent linear time-invariant (LTI) systems by their associated transfer matrix

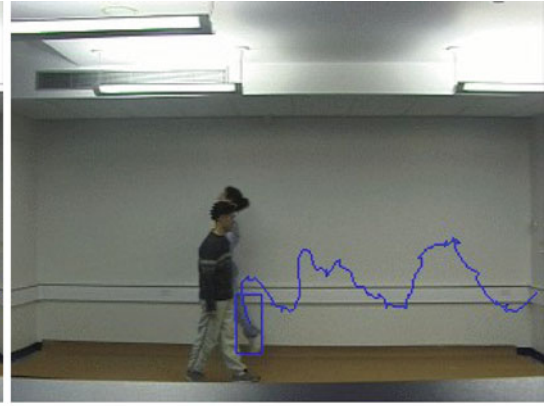
$G(z)$. The “size” of $G(z)$, which plays a key role in assessing the effects of uncertainty, will be measured using the $\mathcal{H}_\infty$ norm, defined as $\|G\|_\infty \doteq \sup_\omega \bar{\sigma}(G(e^{j\omega}))$, where $\bar{\sigma}(\cdot)$ denotes maximum singular value. For scalar systems, this reduces to the peak value of the frequency response (i.e., the maximum gain of the system). In the matrix case, this definition takes into account both the worst-case frequency and spatial direction. Background material on the $\mathcal{H}_\infty$ norm, its computation and its significance in the context of robust control theory, is given in Sánchez-Peña and Sznaier (1998). A general coverage of linear systems theory, including alternative representations of linear systems and their associated properties, can be found, for instance, in Rugh (1996).

Multiframe Tracking

A requirement common to most dynamic vision applications is the *ability to track* objects across frames, in order to collect the data required by a subsequent activity analysis step. Current approaches integrate correspondences between individual frames over time, using a combination of some assumed simple target dynamics (e.g., constant velocity) and empirically learned noise distributions (Isard and Blake 1998; North et al. 2000). However, while successful in many scenarios, these approaches are vulnerable to model uncertainty, occlusion, and appearance changes, as illustrated in Fig. 1.



Frame 150



Frame 95

Uncertainty and Robustness in Dynamic Vision, Fig. 1 Unscented particle filter-based tracking in the presence of occlusion

As shown next, the fragility noted above can be avoided by modeling the motion of the target as the output of a dynamical system, to be identified directly from the available data, along with bounds on the identification error. In the sequel, we consider two different cases: (i) the motion of the target is known to belong to a relatively small set of a priori known motion modalities; and (ii) no prior knowledge is available.

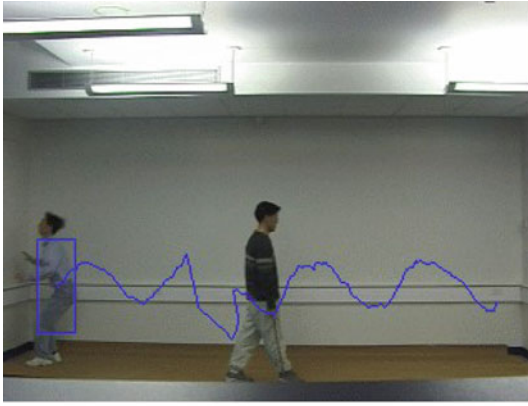
The case of known motion models Consider first the case where a set of models known to span all possible motions of the target is known a priori, as it is often the case with human motion. In this case, the position y_k of a given target can be modeled as $y(z) = \mathcal{F}(z)e(z) + \eta(z)$ where e and η_k denote a suitable input and measurement noise, respectively, and where $\mathcal{F}$ admits an

expansion of the form $\mathcal{F} = \underbrace{\sum_{j=1}^{N_p} p_j \mathcal{F}^j}_{\mathcal{F}_p} + \mathcal{F}_{np}$. Here

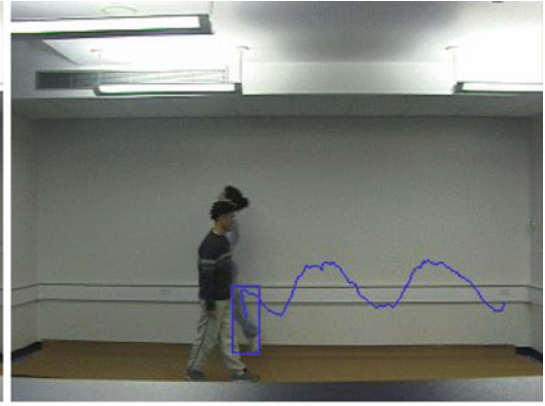
$\mathcal{F}^j$ represent the (known) motion modalities of the target and $\|\mathcal{F}_{np}\|_\infty \leq K$, e.g., a bound on the maximum admissible approximation error of the expansion $\mathcal{F}_p$ to $\mathcal{F}$ is available. In the reminder of this article, we will further assume that a set membership description $\eta_k \in \mathcal{N}$ is available and, without loss of generality, that $e(z) = 1$ (i.e.,

motion of the target is modeled as the impulse response of the unknown operator $\mathcal{F}$).

In this context, the next location of the target feature y_k can be predicted by first identifying the relevant dynamics $\mathcal{F}$ and then using it to propagate its past values. In turn, identifying the dynamics entails finding an operator $\mathcal{F}(z) \in \mathcal{S} \doteq \{\mathcal{F}(z): \mathcal{F} = \mathcal{F}_p + \mathcal{F}_{np}\}$ such that $y - \eta = \mathcal{F}$, precisely the class of interpolation problem addressed in Parrilo et al. (1999). As shown there, finding such an operator reduces to solving a linear matrix inequality (LMI) feasibility problem. Once this operator is found, it can be used in conjunction with a particle (or a Kalman) filter to predict the future location of the target. Figure 2 shows the tracking results obtained using this approach. Here, we used a combination of a priori information: (i) 5% noise level and (ii) $\mathcal{F}_p \in \text{span}[\frac{1}{z-1}, \frac{z}{z-a}, \frac{z}{(z-1)^2}, \frac{z^2}{(z-1)^2}, \frac{z^2 - \cos \omega z}{z^2 - 2 \cos \omega z + 1}, \frac{\sin \omega z^2}{z^2 - 2 \cos \omega z + 1}]$ where $a \in \{0.9, 1, 1.2, 1.3, 2\}$ and $\omega \in \{0.2, 0.45\}$. The experimental information consisted of the position of the target in $N = 20$ frames, where it was not occluded. Note that, by exploiting predictive power of the identified model, the Kalman filter is now able to track the target past the occlusion, eliminating the need for using a (more computationally expensive) particle filter.



Frame 150



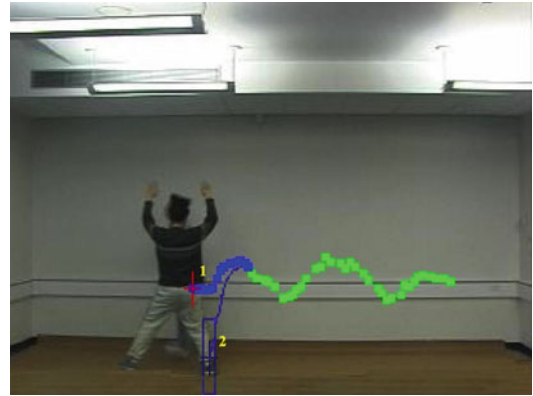
Frame 95

Uncertainty and Robustness in Dynamic Vision, Fig. 2 Using the identified model in combination with Kalman filter allows for robust tracking in the presence of occlusion

Unknown motion models This case could be addressed in principle by performing a purely nonparametric worst-case identification (Parrilo et al. 1999) and then proceeding as above. However, a potential difficulty here stems from the high order of the resulting model (recall that the order of the central interpolant is the number of experimental data points). If a bound n on the order of the underlying models is available, this difficulty can be avoided by recasting the prediction problem into a rank minimization form, which in turn can be relaxed to a semi-definite optimization. To this effect, recall that (Ding et al. 2008), in the absence of noise, given $2n$ values of $\{\mathbf{y}_k\}_{k=t-2n+1}^t$, its next value $\mathbf{y}_{t+1}$ is the unique solution to the following rank minimization problem:

$$\mathbf{y}_{t+1} = \underset{\mathbf{y}}{\operatorname{argmin}} \{ \operatorname{rank} [\mathbf{H}_{n+1}(\mathbf{y})] \} \text{ where } \mathbf{H}_{n+1}(\mathbf{y}) \doteq \begin{bmatrix} \mathbf{y}_{t-2n+1} & \mathbf{y}_{t-2n+2} & \cdots & \mathbf{y}_{t-n} \\ \mathbf{y}_{t-2n+2} & \mathbf{y}_{t-2n+3} & \cdots & \mathbf{y}_{t-n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{t-n+1} & \mathbf{y}_{t-n+2} & \cdots & \mathbf{y} \end{bmatrix} \quad (1)$$

Clearly, the same result holds if multiple elements of the sequence $\mathbf{y}$ are missing, at the price of considering longer sequences (the total number of data points should exceed $2n$). This result allows for handling both noisy and missing



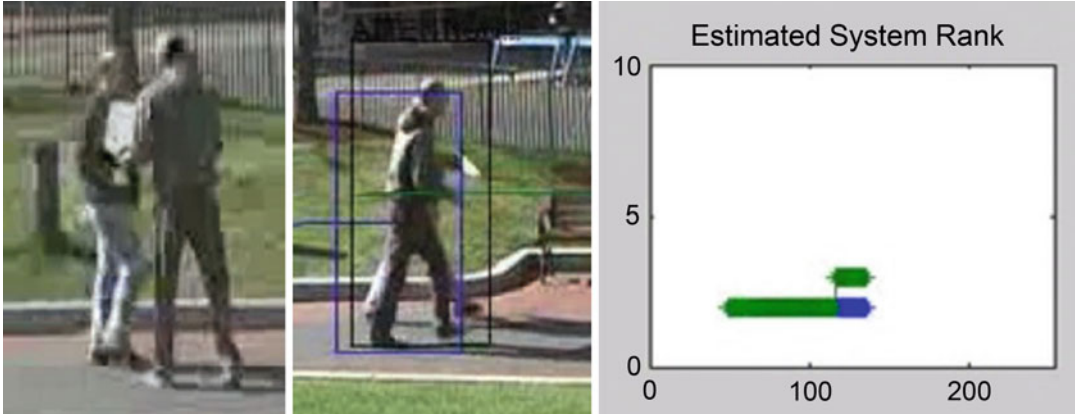
Uncertainty and Robustness in Dynamic Vision, Fig. 3 Trajectory prediction. Rank minimization (1) versus Kalman filtering (2)

data (due, for instance, to occlusion), by simply solving

$$\min_{\zeta} \{ \operatorname{rank} [\mathbf{H}(\zeta)] \} \text{ subject to } \mathbf{v} \in \mathcal{N}_v$$

$$\text{where } \zeta_i = \begin{cases} \mathbf{y}_i - \mathbf{v}_i & \text{if } i \in \mathcal{I}_a \\ \mathbf{x}_i & \text{if } i \in \mathcal{I}_m \end{cases}$$

where $\mathcal{I}_a$ and $\mathcal{I}_m$ denote the set of available (but noisy) and missing measurements, respectively, and where $\mathcal{N}_v$ is a set membership description of the noise $\mathbf{v}$. In the case where $\mathcal{N}_v$ admits a convex description, using the nuclear norm as a surrogate for rank (Fazel et al. 2003) allows for reducing this problem to a convex semi-definite



Uncertainty and Robustness in Dynamic Vision, Fig. 4 The jump in the rank of the Hankel matrix corresponds to the time instant where the subjects meet and exchange a bag

program. Examples of these descriptions are balls in ℓ_∞ , e.g., $\mathcal{N} \doteq \{v: |v_k| \leq \epsilon\}$, or constraints on the norm of $\mathbf{H}_v$, the Hankel matrix of the noise sequence, which under mild ergodicity assumptions are equivalent to constraints on the magnitude of the noise covariance. Figure 3 illustrates the effectiveness of this approach. As shown there, the rank minimization-based filter successfully predicts the location of the target, while a Kalman filter-based tracker fails due to the substantial occlusion.

Event Detection and Activity Classification

Using the trajectories generated by the tracking step for activity recognition entails (i) segmenting the data into homogeneous segments each corresponding to a single activity and (ii) classifying these activities, typically based on exemplars from a database of known activities. As shown in the sequel, both steps can be efficiently accomplished by exploiting the properties of the underlying system. The starting point is to model these activities as the output of a switched piecewise linear system. In this context, under suitable dwell time constraints, each switch (indicating a change in the underlying activity) can be identified by simply searching for points associated with discontinuities in the rank of the associated Hankel matrix, as illustrated in

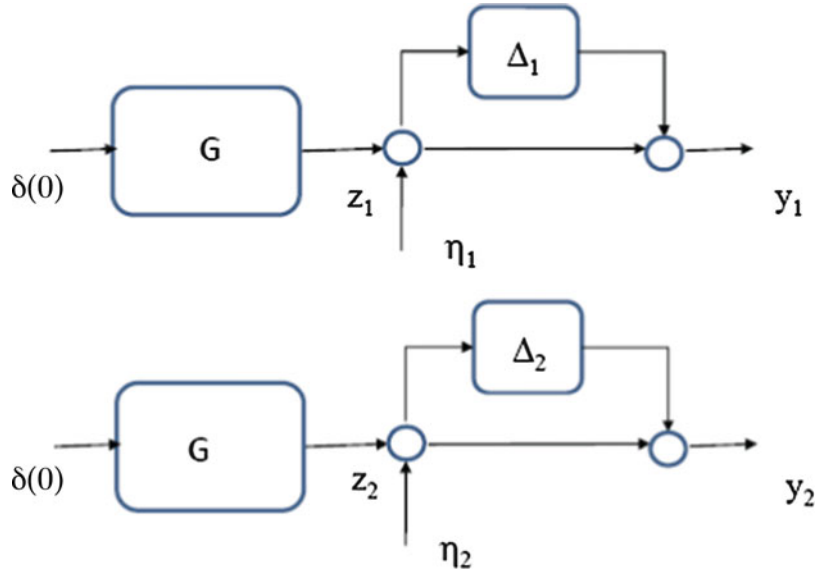
Fig. 4. Further, in this framework, the problem of classifying each subactivity can be recast into the behavioral model (in)validation setup shown in Fig. 5. Here $y_i(\cdot)$ represents the impulse response of the (unknown) LTI system G , affected by measurement noise $\eta_i \in \mathcal{N}$ and uncertainty $\Delta_i \in \mathcal{D}$ that accounts for the variability intrinsic to two different realizations of the same activity. Two different time series are considered to be realizations of the same activity if there exists at least one pair $(\eta_1, \eta_2) \in \mathcal{N}^2$, one pair $(\Delta_1, \Delta_2) \in \mathcal{D}^2$, a LTI system G with McMillan degree at most n_G , and suitable initial conditions $\mathbf{x}_1, \mathbf{x}_2$ resulting in the observed data. Remarkably, this model (in)validation problem can be reduced to a rank minimization form. In the simpler case where $\Delta_i = 0$, the problem can be solved using the following algorithm (Sznaier and Camps 2011):

Next, consider the more realistic case where the trajectories are also affected by bounded model uncertainty Δ , $\|\Delta\|_\infty \leq \gamma$, where γ is given as part of a priori information. In this scenario, the internal signal z is given by $z(t) = \zeta(t) - \eta(t)$, $\eta \in \mathcal{N}$, where the signal ζ satisfies

$$y = (1 + \Delta) * \zeta, \text{ for some } \Delta \in \mathcal{D} \quad (2)$$

where $*$ denotes convolution. Exploiting Theorem 2.3.6 in Chen and Gu (2000) leads to an LMI condition in the variables z, η , for fea-

Uncertainty and Robustness in Dynamic Vision, Fig. 5 Model (in)validation setup



sibility of (2). Thus, the only modification to Algorithm 1 required to handle model uncertainty is to incorporate this additional (convex) constraint to the rank minimization problems. Table 1 shows the results of applying this approach to two video sequences, walking and running, from the KTH database (Laptev et al. 2008). Sample frames from these sequences are shown in Fig. 6. In order to reduce the dimensionality of the data, the frames were first projected into a three-dimensional space using principal component analysis (PCA), and the resulting time series were used as the input to Algorithm 1, assuming 10% noise and 10% model uncertainty. As shown

Uncertainty and Robustness in Dynamic Vision, Table 1 Activity classification results. Sequences (a)–(c) correspond to walking and (d) to running

Activity pair	Rank($\mathbf{H}_1$)	Rank($\mathbf{H}_2$)	Rank($[\mathbf{H}_1 \ \mathbf{H}_2]$)
(a, b)	4	4	4
(a, c)	4	4	4
(a, d)	4	8	8

in Table 1, the algorithm correctly identifies the subsequences (a)–(c) as being generated by the same underlying activity (walking).

Summary and Future Directions

Vision-based systems are uniquely positioned to address the needs of a growing segment of the population. Aware sensors endowed with scene analysis capabilities can prevent crime, reduce time response to emergency scenes, and render viable the concept of ultra-sustainable buildings. Moreover, the investment required to accomplish these goals is relatively modest since a large number of cameras are already deployed and networked. Arguably, at this point, one of the critical factors limiting widespread use of these systems is their potential fragility when operating in unstructured scenarios. This article

Algorithm 1 Behavioral model (in)validation

Data: noisy measurements y_1, y_2

A priori information: noise description $\eta_i \in \mathcal{N}$

1. Solve the following rank minimization problems:
 $r_1^{\min} = \min_{\eta_1} \text{rank}(\mathbf{H}_{y_1} - \mathbf{H}_{\eta_1})$
subject to: $\eta_1 \in \mathcal{N}$.
 $r_2^{\min} = \min_{\eta_2} \text{rank}(\mathbf{H}_{y_2} - \mathbf{H}_{\eta_2})$
subject to: $\eta_2 \in \mathcal{N}$.
 $r_{12}^{\min} = \min_{\eta_1, \eta_2} \text{rank}([\mathbf{H}_{y_{1n}} \ \mathbf{H}_{y_{2n}}])$
subject to: $\eta_1, \eta_2 \in \mathcal{N}$
 $\mathbf{H}_{y_{1n}} = \mathbf{H}_{y_1} - \mathbf{H}_{\eta_1}$
 $\mathbf{H}_{y_{2n}} = \mathbf{H}_{y_2} - \mathbf{H}_{\eta_2}$
2. The given trajectories were generated by the same LTI system with McMillan degree $\leq n_G$ iff:
 $r_1^{\min} = r_2^{\min} = r_{12}^{\min} \leq n_G$



Uncertainty and Robustness in Dynamic Vision, Fig. 6 Sample frames from KTH activity video database. (a) Walking. (b) Running

illustrates the key role that control theory can play in developing a comprehensive, provably robust dynamic vision framework. In turn, computer vision provides a rich environment both to draw inspiration from and to test new developments in systems theory.

to recover 3D structure from 2D data is covered in Ayazoglu et al. (2010). Finally, further details on the connection between identification and the problem of extracting actionable information from large data streams can be found, for instance, in Sznaiier (2012).

Cross-References

- ▶ [Estimation, Survey on](#)
- ▶ [Particle Filters](#)
- ▶ [Vision-Based Motion Estimation](#)
- ▶ [2.5D Vision-Based Estimation](#)

Recommended Reading

Details on how to select good features to track can be found in Szeliski (2010). Using dynamics

Bibliography

- Ayazoglu M, Sznaiier M, Camps O (2010) Euclidean structure recovery from motion in perspective image sequences via Hankel rank minimization. LNCS, vol 6312. Springer, Berlin/New York, pp 71–84
- Chen J, Gu G (2000) Control oriented system identification, An $\mathcal{H}_\infty$ approach. Wiley, New York
- Ding T, Sznaiier M, Camps O (2008) Receding horizon rank minimization based estimation with applications to visual tracking. In: Proceedings of the 47th IEEE conference on decision and control, Cancún, Dec 2008, pp 3446–3451

- Fazel M, Hindi H, Boyd SP (2003) Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In: Proceedings of the 2003 ACC, Denver, pp 2156–2162
- Isard M, Blake A (1998) Condensation – conditional density propagation for visual tracking. *Int J Comput Vis* 29(1):5–28
- Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: IEEE computer vision and pattern recognition, Anchorage, pp 1–8
- North B, Blake A, Isard M, Rittscher J (2000) Learning and classification of complex dynamics. *IEEE Trans Pattern Anal Mach Intell* 22(9):1016–1034
- Parrilo PA, Sánchez-Peña RS, Szaiaier M (1999) A parametric extension of mixed time/frequency domain based robust identification. *IEEE Trans Autom Control* 44(2):364–369
- Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice Hall, Upper Saddle River
- Sánchez-Peña RS, Szaiaier M (1998) Robust systems theory and applications. Wiley, New York
- Szeliski R (2010) Computer vision: algorithms and applications. Springer, New York
- Szaiaier M (2012) Compressive information extraction: a dynamical systems approach. In: Proceeding of the 2012 symposium on systems identification (SYSID 2012), July 2012, Brussels, pp 1559–1568
- Szaiaier M, Camps O (2011) A rank minimization approach to trajectory (in)validation. In: 2011 American control conference, pp 675–680

of freedom (DOF) that it seeks to control such that the number of DOF equals the number of independent control inputs. The control problem is then a fully actuated control problem – something that simplifies the control design problem significantly – but special attention then has to be given to the inherent internal dynamics that has to be carefully analyzed. The other approach to design control systems for underactuated marine vessels seeks to control all DOF using only the limited number of control inputs available. The control problem is then an underactuated control problem and is quite challenging to solve. In this article, it is shown how line-of-sight methods can solve the underactuated control problems that arise from path following and maneuvering control of underactuated marine vessels.

Keywords

Marine vessels · Underactuated marine control problems · Underactuated marine vessels · Underactuation

Underactuated Marine Control Systems

Kristin Y. Pettersen
Department of Engineering Cybernetics,
Norwegian University of Science and
Technology (NTNU), Trondheim, Norway

Abstract

For underactuated marine vessels, the dimension of the configuration space exceeds that of the control input space. This article describes underactuated marine vessels and the control challenges they pose. In particular, there are two main approaches to design control systems for underactuated marine vessels. The first approach reduces the number of degrees

Introduction

Marine systems are often equipped with fewer independent actuators than degrees of freedom. Examples include conventional ships/surface vessels that are typically equipped with a main thruster and a rudder or with two independent main thrusters, but without a side thruster. As a result, we have no control force in the sideways direction. This means that the forward motion (the surge motion) and the orientation (the yaw motion) can be controlled directly, while there is no direct way to influence the sideways motion of the surface vessel (the sway motion). The vessel is then said to be underactuated in sway. It is an underactuated system since it has only two independent control inputs, giving force and torque in surge and yaw, while the system has three degrees of freedom: surge, sway, and yaw. This underactuation leads to challenges when it comes to designing the control system.

Definition: Underactuated Marine Vessels

In order to properly define what we mean by underactuated marine vessels, we need the mathematical model (► [“Mathematical Models of Ships and Underwater Vehicles”](#); Fossen 2011):

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \begin{bmatrix} \tau \\ 0 \end{bmatrix}$$

$$\dot{\eta} = J(\eta)v$$

where the configuration vector $\eta \in \mathbb{R}^n$, the velocity vector $v \in \mathbb{R}^n$, while the vector of independent control inputs $\tau \in \mathbb{R}^m$. The vessel is underactuated because $n > m$, i.e., the dimension of the configuration space exceeds that of the control input space (Oriolo and Nakamura 1991; Pettersen and Egeland 1996).

The underactuation leads to a second-order nonholonomic constraint

$$M_u \dot{v} + C_u(v)v + D_u(v)v + g_u(\eta) = 0$$

where M_u denotes the last $n - m$ rows of the matrix M and $C_u(v)$, $D_u(v)$, and $g_u(\eta)$ are defined similarly.

Definitions of nonholonomic and holonomic constraints can be found in Goldstein (1980). More facts about these kinds of constraints and conditions for when this second-order nonholonomic can be integrated to either a first-order nonholonomic or a holonomic constraint can be found in Tarn et al. (2003).

Control of Underactuated Marine Vessels

As we have seen above, the underactuation leads to a constraint, and this gives challenges when it comes to designing the control system. In particular, it can be shown that if $g_u(\eta)$ has a zero element, then there exists no continuous or discontinuous state feedback law that can asymptotically stabilize the equilibrium point $(\eta, v) = (0, 0)$ (Pettersen and Egeland 1996). This means

that in order to stabilize an equilibrium point, control methods from linear or classical nonlinear control theory cannot be applied.

There are two main classes of approaches to control underactuated marine vessels. The first class approaches the control problem by reducing the number of degrees of freedom that are to be controlled, while the other class seeks to control all degrees of freedom using the limited number of control inputs available.

If we reduce the number of degrees of freedom (DOF) that we seek to control, such that the number of DOF agrees with the number of independent control inputs, then we have a fully actuated control problem although the vessel is underactuated. This may at first sight look like a very simple way to design a control system for underactuated marine vessels. Note, however, that then, there will inherently be internal dynamics that needs to be examined carefully (Isidori 1995; Nijmeijer and van der Schaft 1990). Say, for instance, that we only care about controlling the position of the ship, and we choose not to care very much about the orientation of the ship. We do, for instance, want the ship to follow a straight line trajectory $(x_r(t), y_r(t))$, where x and y give the ship's position in an earth-fixed coordinate system, and the angle giving the ship orientation, ψ , is not so important to us. It is quite straightforward to use, for instance, output feedback linearization to this end (Isidori 1995; Nijmeijer and van der Schaft 1990). The resulting dynamics of the subsystem (x, y) is then called the external dynamics. We have full control over this using the two independent control inputs and can make it track any smooth trajectory $(x_r(t), y_r(t))$. Everything looks simple when considering the external dynamics only, but the internal dynamics can frequently be hard to predict. The orientation of the ship, given by the yaw angle ψ , also needs to be analyzed. The controlled motion will not necessarily have the ship aligned with the tangent of the trajectory, for instance. Firstly, the ship control system that only focuses on the position variables (x, y) may equally well result in the ship moving backward along the line: a behavior that is not really desirable with respect to energy efficiency or for passenger comfort. Secondly,

there will always be environmental disturbances, currents, wind, and waves, and we need to make a thorough stability analysis of the internal dynamics in order to guarantee sufficient robustness to these. So to conclude, if you reduce the number of DOF that you seek to control, in order to achieve a fully actuated control problem, then you need to consider the internal dynamics very carefully when dealing with underactuated marine vessels.

If we follow the other approach to controlling underactuated marine vessels, where we seek to control more degrees of freedom than we have independent control inputs, then we not only have an underactuated marine vessel at hand, but we also have an underactuated control problem. This is a challenging control problem, and we will now see how this can be solved for path following and maneuvering control.

Path Following and Maneuvering Control of Underactuated Marine Vessels

For path following control systems, the control objective is to make the vessel follow a given path $\mathcal{P}$, often defined as a parametrized path

$$Y_d := \{y \in \mathbb{R}^m : \exists \theta \in \mathbb{R} \text{ such that } y = y_d(\theta)\}$$

where $m \leq n$ and y_d is continuously parametrized by the path variable θ . The control objective is thus to force the output y to converge to the desired path $y_d(\theta) : \lim_{t \rightarrow \infty} |y(t) - y_d(\theta(t))| = 0$. This constitutes a geometric task. When there is also a dynamic task, for instance, a speed assignment like forcing the path speed $\dot{\theta}$ to converge to a desired speed $v_s(\theta(t), t)$

$$\lim_{t \rightarrow \infty} |\dot{\theta}(t) - v_s(\theta(t), t)| = 0$$

then the control problem is an output maneuvering problem (Skjetne et al. 2004).

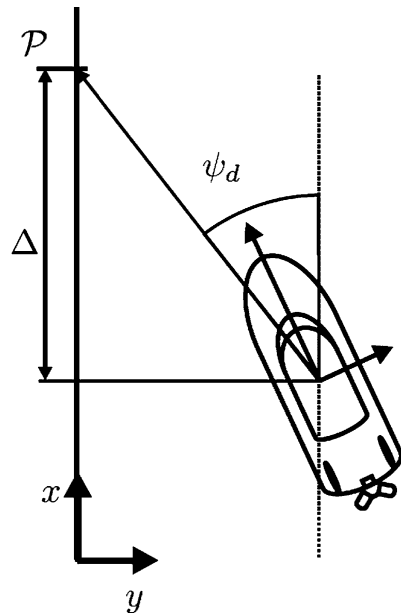
Line-of-sight (LOS) guidance control has proven to be a powerful tool for path following and maneuvering control of underactuated vessels. LOS guidance is much used in practice

for manual control of ships, where the helmsman typically will steer the vessel toward a point lying a constant distance, called the look-ahead distance, ahead of the vessel along the desired path. LOS guidance is simple, intuitive, and easy to tune, and it can be shown that it provides nice path convergence properties (Lefebvre et al. 2003; Fredriksen and Pettersen 2006; Børhaug et al. 2008; Breivik and Fossen 2004; Caharija et al. 2012). For the simplified case without any environmental disturbances and when the desired path is a straight line, the LOS guidance law for an underactuated surface vessel is given by

$$\psi_d = \psi_{\text{LOS}} = -\tan^{-1}\left(\frac{y}{\Delta}\right), \quad \Delta > 0$$

where y is the cross-track error. The angle ψ_{LOS} is called the line-of-sight (LOS) angle, and geometrically, it corresponds to the orientation of the vessel when headed toward the point that lies a distance $\Delta > 0$ ahead of the vessel along the path $y = 0$, cf. Fig. 1. The look-ahead distance Δ is a control design parameter.

In order to handle ocean currents and other environmental disturbances such as wind and



Underactuated Marine Control Systems, Fig. 1 Illustration of LOS guidance

waves, the LOS guidance law can be extended with integral action

$$\psi_{\text{LOS}}^m = -\tan^{-1} \left(\frac{y + \sigma y_{\text{int}}}{\Delta} \right), \quad \Delta > 0$$

$$\dot{y}_{\text{int}} = \frac{\Delta y}{(y + \sigma y_{\text{int}})^2 + \Delta^2}$$

where $\sigma > 0$ is a design parameter, an integral gain, and $\Delta > 0$ has the same interpretation as above. The integral effect will generate a sideslip angle that allows the vessel to stay on the desired path even though affected by environmental disturbances with components normal to the path, even though the vessel has no control forces to act in the sideways direction.

Various standard control techniques can readily be used to track the above guidance commands. LOS guidance can also be extended to the 3D case for path following/maneuvering control of underactuated autonomous underwater vehicles (AUV) (cf. the references given above).

Summary and Future Directions

Underactuated marine vessels are vessels for which the dimension of the configuration space exceeds that of the control input space. There are two main approaches to design control systems for underactuated marine vessels. The first approach reduces the number of degrees of freedom (DOF) that it seeks to control, such that the number of DOF equals the number of independent control inputs. The control problem is then a fully actuated control problem, something that simplifies the control design problem significantly, but special attention then has to be given to the inherent internal dynamics that has to be carefully analyzed. The other approach to design control systems for underactuated marine vessels seeks to control all DOF using only the limited number of control inputs available. The control problem is then an underactuated control problem, and this is a quite challenging control problem. In this entry, it is shown how line-of-sight methods can solve

the underactuated control problems that arise from path following and maneuvering control of underactuated marine vessels.

Future developments of underactuated marine control systems will include solving more underactuated control problems of marine vessels taking into account both the complete mathematical model of the vessels and also advanced mathematical models of all the environmental disturbances in both 2D and 3D.

Cross-References

- [Mathematical Models of Ships and Underwater Vehicles](#)
- [Motion Planning for Marine Control Systems](#)
- [Underactuated Robots](#)

Bibliography

- Aguiar AP, Pascoal AM (2007) Dynamic positioning and way-point tracking of underactuated AUVs in the presence of ocean currents. *Int J Control* 80:1092–1108
- Breivik M, Fossen TI (2004) Path following of straight lines and circles for marine surface vessels. In: *Proceedings of 6th IFAC conference on control applications in marine systems*, Ancona, pp 65–70
- Børhaug E, Pavlov A, Pettersen KY (2008) Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents. In: *Proceedings of 47th IEEE conference on decision and control*, Cancun, 9–11 Dec 2008, pp 4984–4991
- Caharija W, Pettersen KY, Gravdahl JT, Børhaug E (2012) Path following of underactuated autonomous underwater vehicles in the presence of ocean currents. In: *Proceedings of 51th IEEE conference on decision and control*, Maui, Dec 2012, pp 528–535
- Encarnacao P, Pascoal AM, Arcak M (2000) Path following for marine vehicles in the presence of unknown currents. In: *Proceedings of 6th IFAC international symposium on robot control*, Vienna, 21–23 Sept 2000, pp 469–474
- Fossen TI (2011) *Handbook of marine craft hydrodynamics and motion control*. Wiley, Chichester/West Sussex
- Fredriksen E, Pettersen KY (2006) Global K-exponential way-point maneuvering of ships: theory and experiments. *Automatica* 42:677–687
- Goldstein H (1980) *Classical mechanics*, 2nd edn. Addison-Wesley, Reading
- Healey AJ, Lienard D (1993) Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE J Ocean Eng* 18:327–339

- Indiveri G, Zizzari AA (2008) Kinematics motion control of an underactuated vehicle: a 3D solution with bounded control effort. In: Proceedings of 2nd IFAC workshop on navigation, guidance and control of underwater vehicles. Killaloe, Ireland
- Isidori A (1995) Nonlinear control systems, 3rd edn. Springer, London
- Lapierre L, Soetanto D, Pascoal AM (2003) Nonlinear path following with applications to the control of autonomous underwater vehicles. In: Proceedings of 42nd IEEE conference on decision and control, Maui, Dec 2003, pp 1256–1261
- Lefeber AAJ, Pettersen KY, Nijmeijer N (2003) Tracking control of an under-actuated ship. *IEEE Trans Control Syst Technol* 11:52–61
- Nijmeijer H, van der Schaft AJ (1990) Nonlinear dynamical control systems. Springer, New York
- Oriolo G, Nakamura Y (1991) Control of mechanical systems with second-order nonholonomic constraints: underactuated manipulators. In: Proceedings of 30th IEEE conference on decision and control, Brighton, Dec 1991, pp 2398–2403
- Pettersen KY, Egeland O (1996) Exponential stabilization of an underactuated surface vessel. In: Proceedings of 35th IEEE conference on decision and control, Kobe, Dec 1996, pp 967–972
- Pettersen KY, Egeland O (1999) Time-varying exponential stabilization of the position and attitude of an underactuated autonomous underwater vehicle. *IEEE Trans Autom Control* 44:112–115
- Skjetne R, Fossen TI, Kokotovic PV (2004) Robust output maneuvering for a class of nonlinear systems. *Automatica* 40:373–383
- Tarn T-J, Zhang M, Serrani A (2003) New integrability conditions for classifying holonomic and nonholonomic systems. In: Rantzer A, Byrnes CI (eds) *Directions in mathematical systems theory and optimization*, Springer, Berlin/Heidelberg, pp 317–331

an overview of controllability, stabilization, and motion planning.

Keywords

Nonholonomic constraints · Nonlinear control · Underactuation

Introduction

An *underactuated robot* is a robot with fewer actuators (control inputs) than the number of variables describing its configuration (degrees of freedom). Some robots have this property unavoidably, while others are specifically designed this way, perhaps to save the cost of actuators. Examples include:

- **A cart and pendulum (inverted pendulum).** This system has two degrees of freedom, the linear position of the cart and the angle of the pendulum, but only one control input, the acceleration of the cart.
- **A car.** A car has only two control inputs (steering and forward/backward speed) but at least three degrees of freedom: the position (x, y) and orientation θ of the chassis. If the steering and/or rolling angles of the wheels are included in the representation of the configuration, the car has even more degrees of freedom.
- **A walking robot.** When a biped steps with one foot in the air and the toes of the other foot on the ground, there is no actuator at the toes to directly control the angle between the foot and the ground.
- **A quadrotor flying robot.** A quadrotor has four control currents driving the four propellers, but its configuration is described by six variables: (x, y, z) position and roll, pitch, and yaw.
- **An underactuated robot hand.** Robot hands generally have many joints, up to four per finger for anthropomorphic hands. To reduce cost, a small number of motors (as few as one) may be used to open and close the fingers, with joint motions coupled by springs.

Underactuated Robots

Kevin M. Lynch
Mechanical Engineering Department,
Northwestern University, Evanston, IL, USA

Abstract

Underactuated robots, robots with fewer actuators than degrees of freedom, are found in many robot applications. This entry classifies underactuated robots according to their dynamics and constraints and provides

- **Robot manipulation.** When a robot arm and hand manipulates a rigid object, the entire system, taken together, has at least six more degrees of freedom than actuators – the six degrees of freedom of the object.

In all underactuated robot systems of interest, the fewer control inputs are somehow coupled to all of the degrees of freedom. This entry focuses on coupling through the inertia matrix and kinematic constraints. In addition, this entry focuses on control of the full configuration, or more generally the state, of the robot system. Other goals, such as successfully grasping an object with a compliant underactuated hand, are outside the scope of this entry.

Classification of Underactuated Robots

The robot has n degrees of freedom, and its configuration is written in local coordinates as a column vector $q \in \mathbb{R}^n$. If the robot is described as a *kinematic* system, then its state x is simply q and the control inputs are velocities. If the robot is a *mechanical* system, then its state is $x = [q^T, \dot{q}^T]^T$ and the control inputs are accelerations (forces). Let p denote the dimension of the state space, where $p = n$ for a kinematic system and $p = 2n$ for a mechanical system.

The equations of motion of an underactuated robot can be written in the control-affine form

$$\dot{x} = f(x) + \sum_{i=1}^m u_i g_i(x) \quad \text{where } m < n. \quad (1)$$

The vector field $f(x)$ is a *drift vector field* describing the unforced motion of the robot, the $g_i(x)$ are linearly independent *control vector fields* describing how the controls act on the robot, and $u = [u_1, \dots, u_m]^T$ is the control. Kinematic systems are commonly *drift-free* ($f(x) = 0$). For a mechanical system, the drift field $f(x)$ typically includes velocities acting on positions and gravity acting on velocities.

The fact that the number of controls m is less than the number of degrees of freedom n can be viewed as $n - m$ constraints on the motion. For a kinematic system, these are velocity constraints. For a mechanical system, these are acceleration constraints. In addition, a mechanical system may be subject to a separate set of k velocity constraints, often called *Pfaffian* constraints, of the form

$$A(q)\dot{q} = 0, \quad (2)$$

where $A(q) \in \mathbb{R}^{n \times k}$. Such constraints arise from conservation laws and rolling without slip, for example.

Understanding the integrability of these constraints is key to understanding the controllability of underactuated robots (section “[Determining Controllability](#)”). For example, if acceleration constraints can be integrated to yield equivalent velocity constraints, then the dimension of the space of reachable velocities of the mechanical system is reduced. If velocity constraints can be integrated to yield equivalent configuration constraints, then the dimension of the reachable configuration space is reduced. If some velocity constraints are integrable to configuration constraints, we simply eliminate those configuration variables from the description of the system so we can focus on the controllable degrees of freedom. Velocity constraints that cannot be integrated are called *nonholonomic*, while configuration constraints are called *holonomic*.

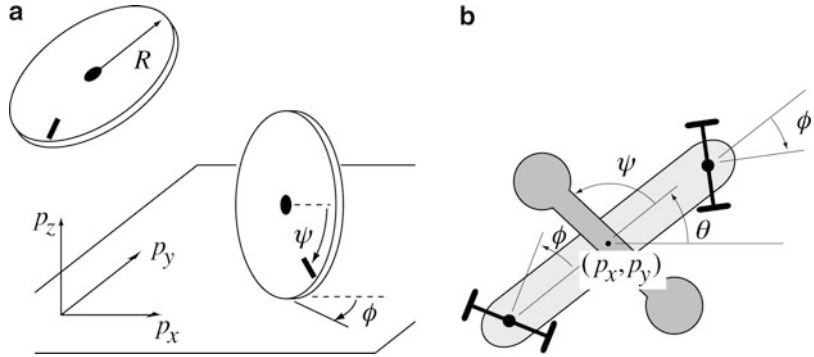
Based on the type of constraints, we can classify underactuated robots into three categories – pure kinematic, pure mechanical, and mixed kinematic and mechanical – as described below.

Pure Kinematic

This category consists of systems with velocities as inputs, as well as mechanical systems that can be modeled by a *kinematic reduction* that has time-differentiable velocities as controls (Bullo et al. 2002; Bullo and Lewis 2004). (The actual acceleration controls of the original system are the time derivatives of these velocities.) Examples of mechanical systems that can be reduced to kinematic systems include systems with actuators for every degree of freedom (fully actuated

Underactuated Robots,

Fig. 1 (a) A wheel in space, then confined to be upright on a horizontal plane with coordinates (p_x, p_y, ψ, ϕ) . (b) A top view of a robotic snakeboard. The configuration is given by (p_x, p_y, θ) for the board, the steering angle ϕ of the wheels, and the angle ψ of the reaction wheel



systems, of little interest here) and mechanical systems whose acceleration constraints can be completely integrated to equivalent velocity constraints.

Exam 1 (Upright rolling wheel) Consider a wheel of radius R rolling upright on a horizontal plane (Fig. 1a). The center of the wheel is (p_x, p_y, p_z) , and the orientation is described by its “leaning” angle θ , rolling angle ψ , and heading angle ϕ . The constraints that the wheel remain upright and touching the plane can be written differentially as $\dot{p}_z = 0$ and $\dot{\theta} = 0$, but these constraints can be integrated to the equivalent configuration constraints $p_z = R$ and $\theta = 0$, so we eliminate these variables from the description of the configuration and focus on the remaining four coordinates.

Writing the configuration vector as $q = [p_x, p_y, \psi, \phi]^T$ and the two control inputs as the rolling velocity $u_1 = \dot{\psi}$ and the heading rate of change $u_2 = \dot{\phi}$, the control system is

$$\dot{q} = u_1 g_1(q) + u_2 g_2(q),$$

where $g_1(q) = [R \cos \phi, R \sin \phi, 1, 0]^T$ and $g_2(q) = [0, 0, 0, 1]^T$. Implicit in these equations of motion are the two rolling constraints $A(q)\dot{q} = 0$, where

$$A(q) = \begin{bmatrix} 1 & 0 & -R \cos \phi & 0 \\ 0 & 1 & -R \sin \phi & 0 \end{bmatrix}.$$

These velocity constraints cannot be integrated to equivalent configuration constraints.

Exam 2 (Reaction-wheel satellite) The three-dimensional orientation of a satellite can be controlled by spinning internal reaction wheels. The controls to the reaction wheels are torques. By conservation of angular momentum, the total angular momentum P of the satellite is subject to the constraints

$$P = J\omega + \sum_i J_i \omega_i = \text{constant},$$

where J is the inertia of the satellite body, ω is its angular velocity, J_i is the inertia of momentum wheel i , and ω_i is its angular velocity. These constraints are velocity constraints – given the angular velocity of the momentum wheels, the angular velocity of the satellite is known. Thus, we can treat the original mechanical system as a kinematic system with (differentiable) angular velocities of the momentum wheels as inputs. If the system satisfies $P = 0$, the kinematic reduction is drift-free.

While satellite orientation is commonly controlled using three orthogonal reaction wheels (a fully actuated system), two reaction wheels suffice to control the orientation of the kinematic reduction in the case $P = 0$. This is apparent from the fact that successive rotations about two orthogonal body-fixed axes (e.g., body-referenced ZYZ Euler angles) are sufficient to arbitrarily orient a rigid body in space.

Pure Mechanical

This category consists of mechanical systems without any velocity constraints.

Exam 3 (3R robot arm with a passive joint) The dynamics of a robot arm are determined by its inertia matrix $M(q)$, from which the kinetic energy $K = \frac{1}{2}\dot{q}^T M(q)\dot{q}$ is derived, and its potential energy $V(q)$. If one of the joints of the arm rotates freely without an actuator, the arm is underactuated. One such robot is a planar arm with two actuated joints and one passive (Lynch et al. 2000; Bullo and Lynch 2001). For this robot, the acceleration constraint arising from the lack of an actuator cannot be integrated to an equivalent velocity constraint.

Mixed Kinematic and Mechanical

This category consists of mechanical systems with both (1) velocity constraints and (2) acceleration constraints that cannot be integrated.

Exam 4 (Snakeboard) The snakeboard is a skateboard with steerable wheels. The rider can locomote without touching the ground by twisting his or her body while steering the wheels. The configuration of a robotic model of the snakeboard and rider (Ostrowski et al. 1994) consists of the position (x, y) and orientation θ of the board, the steering angle of the wheels (assumed to be coupled to be equal and opposite), and the angle of a reaction wheel representing the rider (Fig. 1b). The controls are the steering torque to the wheels and the driving torque of the reaction wheel. This system is mixed because of the presence of the no-slipping constraint at the wheels.

While in some cases it is obvious whether velocity or acceleration constraints can be integrated to equivalent constraints on configuration or velocity, respectively, in general this is not trivial. Instead of attempting to determine the integrability of constraints, we typically study the reachable sets of the system (1). This is the topic of controllability of nonlinear systems, section “Determining Controllability”.

Underactuated robots can also be classified according to the set of available controls $\mathcal{U} \subseteq \mathbb{R}^m$. For example, the control set could be a discrete set of points in $\mathbb{R}^m$, or only nonnegative values, or a bounded set of $\mathbb{R}^m$ containing the

origin in the interior. For simplicity, assume $u \in \mathcal{U} = \mathbb{R}^m$.

Control Challenges

Determining Controllability

For linear systems of the form $\dot{x} = Ax + Bu$, $x \in \mathbb{R}^p$, $u \in \mathbb{R}^m$, there is one notion of controllability, determined by the Kalman rank condition (KRC). If the rank of the matrix

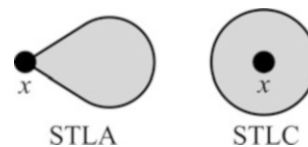
$$[B \ AB \ A^2B \ \dots \ A^{p-1}B]$$

is p , then it is possible to transfer the system from any state to any other state in finite time.

Most underactuated systems of the form (1), such as all of the examples given above, are nonlinear systems, however. For nonlinear systems, there are many possible notions of controllability (see Bullo and Lewis 2004; Lynch et al. 2011; Nijmeijer and van der Schaft 1990; Sussmann 1983). Some examples include:

- *Small-time local accessibility (STLA) at x* : For any time $T > 0$, the reachable set starting from x at times $t < T$ contains a full-dimensional subset of the state space.
- *Small-time local controllability (STLC) at x* : For any time $T > 0$, the reachable set starting from x at times $t < T$ contains a neighborhood of x .
- *Global controllability*: The robot can reach any state from any other state.

STLC is strictly stronger than STLA. Neither implies global controllability nor does global controllability imply either of the local properties. STLA and STLC are illustrated in Fig. 2.



Underactuated Robots, Fig. 2 Example reachable sets in small time for systems that are STLA and STLC at x

STLA can be tested by a Taylor expansion of flows along vector fields. A key object in this study is the *Lie bracket* of two vector fields $V_1(x)$ and $V_2(x)$, defined as the new vector field

$$[V_1, V_2] = \frac{\partial V_2}{\partial x} V_1 - \frac{\partial V_1}{\partial x} V_2.$$

If the system were to start from x and flow along V_1 for a short time ϵ , then V_2 for ϵ , then $-V_1$ for ϵ , then $-V_2$ for ϵ , a Taylor expansion shows that the net motion of the system would be $\epsilon^2[V_1, V_2](x)$ (plus terms of order ϵ^3 and higher). If this direction is neither zero nor a linear combination of V_1 and V_2 , then effectively a new motion direction has been created.

For the upright rolling wheel, the Lie bracket of g_1 (forward-backward rolling) and g_2 (turning) is

$$[g_1, g_2] = [R \sin \phi, -R \cos \phi, 0, 0]^T,$$

a sideways “parallel parking” motion. This new direction increases the dimension of the locally reachable set beyond what could be reached by a local linearization of the nonlinear system.

Roughly speaking, the *Lie algebra* of a set of vector fields $\mathcal{V}$ is the set of vector fields $\mathcal{V}$, all iterated Lie brackets of these vector fields, and their linear combination. For example, the Lie algebra of $\mathcal{V} = \{g_1, g_2\}$ includes $[g_1, g_2]$, $[g_1, [g_1, g_2]]$, $[g_2, [g_1, [g_1, g_2]]]$, etc., as well as their linear combinations. Deeper Lie brackets correspond to higher-order terms in the Taylor expansion of flows.

With these concepts, a theorem due to Chow (1939) says that a system (1) satisfies STLA at x if the dimension of the Lie algebra of $\{f, g_1, \dots, g_m\}$ at x is p , the dimension of the state space. This is known as the *Lie algebra rank condition* (LARC). Most underactuated systems of interest satisfy the LARC but not the KRC. For the upright rolling wheel, the linearization at any q fails the KRC, but the four-dimensional configuration space is spanned by $g_1, g_2, [g_1, g_2]$, and $[g_2, [g_1, g_2]]$ at all q , satisfying the LARC. Therefore the system is STLA at all points.

The STLA property can be strengthened to STLC if the system additionally satisfies certain symmetry properties, allowing it to proceed both forward and backward along Lie bracket directions. For example, if $f(x) = 0$ and the control set $\mathcal{U}$ contains the origin in the interior, the LARC implies STLC. This is the case for the upright rolling wheel. More general notions of symmetry have also been derived (e.g., Sussmann 1987).

For mechanical systems, STLC can only hold at zero-velocity states where $f(x) = 0$. In addition, velocity constraints may prevent the system from reaching a $2n$ -dimensional set in state space. A more relevant question may be whether the configuration alone can be locally controlled at zero-velocity states. Specialized Lie-algebraic controllability tests have been developed for configuration controllability of mechanical and mixed systems (Lewis 2000; Bullo and Lynch 2001; Bullo et al. 2002; Bullo and Lewis 2004).

Global controllability results often derive from STLC at all states for drift-free systems or from STLA and global properties of the vector fields or the topology of the state space (Choset et al. 2005).

Feedback Stabilization

For some underactuated robots, the linearization at a state x may satisfy the KRC. An example is an inverted pendulum linearized at a balanced equilibrium state. In this case, it is possible to derive a linear feedback controller, based on the linearization, to stabilize the balanced state.

For many underactuated systems of interest, however, the linearization at a desired state is not controllable. For such systems, a famous theorem due to Brockett (1983), plus subsequent strengthening, implies the following:

Theorem 1 *For any drift-free underactuated kinematic system of the form (1), there exists no time-invariant continuous state feedback law that stabilizes the origin.*

For example, there exists no continuous state-feedback control law that can stabilize the upright rolling wheel to a desired configuration.

This obstruction to stabilizability has resulted in a number of different approaches to feedback control of underactuated systems, including (1) time-varying feedback control laws, (2) feedback control laws that are discontinuous in the state, and (3) two-degree-of-freedom controllers consisting of a motion planner plus a feedback controller for the easier problem of stabilizing the nominal trajectory. Strategies for planning nominal motions for two-degree-of-freedom controllers are discussed next.

Motion Planning

Given an initial state $x(0) = x_{\text{start}}$ and a goal state x_{goal} , the motion planning problem for a system $\dot{x} = h(x, u)$ is to find a control history $u : [0, T] \rightarrow \mathcal{U}$ such that

$$x_{\text{goal}} = x_0 + \int_0^T h(x(s), u(s)) ds$$

while avoiding any obstacles that may be present in the environment. It may also be desired to minimize some notion of cost,

$$J = \int_0^T L(x(s), u(s)) ds.$$

One choice of $L(s)$ is $u^T(s)u(s)$, the square of the control effort.

Ideally the motion planning method would be *complete* (guaranteed to find a solution in finite time if one exists) or *probabilistically complete* (if a solution exists, the probability of finding a solution goes to one as time goes to infinity).

A variety of approaches to motion planning have been proposed in the robotics literature. Approaches that apply to underactuated systems include:

- *Search-based methods.* A popular class of search-based methods are rapidly exploring random trees (RRTs) and variants (LaValle and Kuffner 2001). These approaches offer probabilistic completeness for many systems, including systems with obstacles, but naïve implementations may be slow to find

solutions, and the solutions generally do not satisfy optimality criteria.

- *Numerical optimization.* The control history can be converted to a finite parameterization using representations such as polynomials, cubic B-splines, wavelets, and truncated Fourier series. Numerical optimization methods can then be applied to solve the two-point boundary value problem while minimizing a cost function. Gradient-based numerical optimization methods may yield locally optimal solutions, but they may suffer from numerical convergence problems, and they may get stuck in local minima depending on an initial guess. Optimization methods that do not use gradient information potentially offer globally optimal solutions, but typically at the expense of significantly longer computation times.
- *Fictitious input methods.* These methods assume that there is a direct control input available for each Lie bracket motion direction. These fictitious inputs are then converted to a sequence of feasible inputs utilizing the Campbell-Baker-Hausdorff-Dynkin expansion of flows (Lafferriere and Sussmann 1991). In general, these methods require iterative application to account for errors in the approximate conversion.
- *Trajectory transformation methods.* One way to deal with obstacles is to first use a global motion planner that is complete under the assumption that the robot has no motion constraints. Then the template unconstrained solution is iteratively subdivided into smaller pieces, with each piece replaced by an obstacle-free feasible trajectory generated by a local planner. If the system is drift-free and STLTC at all configurations, then it is possible to develop a local planner that guarantees success of the transformation from an unconstrained trajectory to a feasible trajectory as the subdivisions get small enough (Laumond et al. 1994).

Often it is possible to exploit structure of the equations of motion beyond the general form (1). Making use of extra structure can reduce the computational complexity of motion planning.

- *Chained form, sinusoidal controls, and averaging.* Certain drift-free kinematic systems can be transformed to a canonical *chained form*. For systems in such a form, sinusoidal controls of integrally related frequencies can be chosen to drive one of the configuration variables to its desired value while having zero net effect on configuration variables already at their desired value. In this way, configuration variables can be driven sequentially to their desired values (Murray and Sastry 1993).

For many underactuated systems, sinusoidal controls can be used to achieve approximate motion in each Lie bracket direction needed to complete the LARC. The resulting periodic motions are sometimes called *gaits*, and motion planning can be achieved using a finite set of gaits (Bullo and Lewis 2004; Ostrowski et al. 1994).

- *Differentially flat systems.* For certain underactuated systems with $u \in \mathbb{R}^m$, there exist a set of m functions w_i of the state, the control, and its derivatives,

$$w_i(x, u, \dot{u}, \ddot{u}, \dots, u^{(r)}), \quad i = 1 \dots m,$$

such that the states and control inputs can be expressed as functions of w and its time derivatives. The w_i are called *flat outputs*. The motion planning problem is to find $w(t), t \in [0, T]$, such that $w(0), \dot{w}(0), \ddot{w}(0), \dots$ and $w(T), \dot{w}(T), \ddot{w}(T), \dots$ satisfy the constraints specified by x_{start} and x_{goal} . The problem changes from constrained motion planning in the p -dimensional state space to finding a curve satisfying start and end constraints on w and its derivatives (Fliess et al. 1995; Sira-Ramirez and Agrawal 2004).

- *Kinematic reductions.* Motion planning in configuration space is a lower-dimensional problem than motion planning in configuration-velocity space. Therefore, when a mechanical system can be reduced to a kinematic equivalent, motion planning can be more efficient. Examples include mechanical systems that can be fully reduced to a kinematic system (like the reaction-wheel satellite) and mechanical systems

that admit rank-1 kinematic reductions – vector fields on configuration space that can be followed at any speed, despite the underactuation constraints. These vector fields become primitives for motion planning on configuration space (Bullo and Lynch 2001; Bullo et al. 2002; Bullo and Lewis 2004; Choset et al. 2005).

Summary and Future Directions

Underactuated systems arise in all areas of robotics, including robot manipulation and aerial, ground, and underwater locomotion. Underactuation raises a number of challenging issues in robot motion planning and control. While significant progress has been made, further research is needed on computationally efficient motion planning and robust stabilization of nominal trajectories. In addition, although this entry focuses on systems that can be described by a single set of dynamics, many interesting underactuated systems are hybrid systems that experience changing contact constraints. Examples include biped robots striding from one foot to the next and robot manipulators that manipulate objects with changing contact modes (grasping, rolling, pushing, etc.). Further work is needed to incorporate contact models, beyond simple kinematic constraints, and changing equations of motion in motion planning and control of hybrid underactuated systems.

Cross-References

- [Controllability and Observability](#)
- [Differential Geometric Methods in Nonlinear Control](#)
- [Feedback Linearization of Nonlinear Systems](#)
- [Feedback Stabilization of Nonlinear Systems](#)
- [Hybrid Dynamical Systems, Feedback Control of](#)
- [Lie Algebraic Methods in Nonlinear Control](#)
- [Nonlinear Zero Dynamics](#)
- [Underactuated Marine Control Systems](#)
- [Walking Robots](#)
- [Wheeled Robots](#)

Recommended Reading

Introductions to underactuated robot systems can be found in Choset et al. (2005), Lynch et al. (2011), and Murray et al. (1994).

While this entry focuses on configuration spaces modeled locally as $\mathbb{R}^n$, most robotic systems consist of rigid bodies whose positions and orientations can be described globally as elements of the Lie group $SE(3)$ or one of its subgroups: $SE(2)$, $SO(3)$, or $SO(2)$. Geometric methods for control of underactuated systems make use of the extra structure of Lie groups and their Lie algebras, symmetries, and concepts from geometric mechanics such as tangent and cotangent bundles, Riemannian metrics on manifolds, symplectic manifolds, connections, fiber bundles, covariant derivatives, etc. Excellent treatments can be found in Bloch et al. (2003), Bullo and Lewis (2004), and Murray et al. (1994).

Bibliography

- Bloch AM, Baillieul J, Brouch PE, Marsden JE (2003) Nonholonomic mechanics and control. Springer, New York
- Brockett RW (1983) Asymptotic stability and feedback stabilization. In: Brockett RW, Millman RS, Sussmann HJ (eds) Differential geometric control theory. Birkhauser, Boston
- Bullo F, Lewis AD (2004) Geometric control of mechanical systems. Springer, New York/Heidelberg/Berlin
- Bullo F, Lynch KM (2001) Kinematic controllability for decoupled trajectory planning of underactuated mechanical systems. IEEE Trans Robot Autom 17(4):402–412
- Bullo F, Lewis AD, Lynch KM (2002) Controllable kinematic reductions for mechanical systems: concepts, computational tools, and examples. In: International symposium on the mathematical theory of networks and systems, South Bend, IN, Aug 2002
- Choset H, Lynch KM, Hutchinson S, Kantor G, Burgard W, Kavraki L, Thrun S (2005) Principles of robot motion. MIT, Cambridge
- Chow W-L (1939) Über systemen von linearen partiellen differentialgleichungen erster ordnung. Math Ann 117:98–105
- Fliess M, Lévine J, Martin P, Rouchon P (1995) Flatness and defect of nonlinear systems: introductory theory and examples. Int J Control 61(6):1327–1361
- Lafferriere G, Sussmann H (1991) Motion planning for controllable systems without drift. In: IEEE international conference on robotics and automation, Sacramento, pp 1148–1153
- Laumond J-P, Jacobs PE, Taïx M, Murray RM (1994) A motion planner for nonholonomic mobile robots. IEEE Trans Robot Autom 10(5):577–593
- LaValle SM, Kuffner JJ (2001) Rapidly-exploring random trees: progress and prospects. In: Donald BR, Lynch KM, Rus D (eds) Algorithmic and computational robotics: new directions. A. K. Peters, Natick
- Lewis AD (2000) Simple mechanical control systems with constraints. IEEE Trans Autom Control 45(8):1420–1436
- Lynch KM, Shiroma N, Arai H, Tanie K (2000) Collision-free trajectory planning for a 3-DOF robot with a passive joint. Int J Robot Res 19(12): 1171–1184
- Lynch KM, Bloch AM, Drakunov SV, Reyhanoglu M, Zenkov D (2011) Control of nonholonomic and underactuated systems. In: Levine W (ed) The control systems handbook: control system advanced methods, 2nd edn. Taylor and Francis, Boca Raton
- Murray RM, Sastry SS (1993) Nonholonomic motion planning: steering using sinusoids. IEEE Trans Autom Control 38(5):700–716
- Murray RM, Li Z, Sastry SS (1994) A mathematical introduction to robotic manipulation. CRC Press, Boca Raton
- Nijmeijer H, van der Schaft AJ (1990) Nonlinear dynamical control systems. Springer, New York
- Ostrowski J, Lewis A, Murray R, Burdick J (1994) Nonholonomic mechanics and locomotion: the snakeboard example. In: IEEE international conference on robotics and automation, San Diego, CA, pp 2391–2397
- Sira-Ramirez H, Agrawal SK (2004) Differentially flat systems. CRC Press, New York
- Sussmann HJ (1983) Lie brackets, real analyticity and geometric control. In: Brockett RW, Millman RS, Sussmann HJ (eds) Differential geometric control theory. Birkhauser, Boston
- Sussmann HJ (1987) A general theorem on local controllability. SIAM J Control Optim 25(1):158–194

Underwater Robots

Gianluca Antonelli

University of Cassino and Southern Lazio,
Cassino, Italy

Abstract

Several different kinds of underwater robots are currently used for a variety of applications such as mapping, exploration, survey, and intervention on subsea infrastructures.

This entry briefly discusses their main characteristics.

Keywords

Robotics · Field robotics · Marine robotics · Underwater robotics

Introduction

Two-thirds of the Earth's surface are covered by water. The role of the oceans is crucial in our everyday life; they represent a medium of goods transportation and food resources since ancient times. Presently, also natural resources such as oil and gas are extracted from the sea bottom; large areas surveys are thus of crucial importance. The trend is to invest in offshore mining to search for metals and gas hydrates. From the environmental aspect, the study of the oceans is crucial to better understand global phenomena such as Earth warming or try to model events such as hurricanes and tsunamis. The sea is obviously also the field of encounter for military strategies; finally, a certain interest in underwater search and rescue also comes from air crash investigations.

Technology is helping us in observing, measuring, monitoring, and exploiting the oceans. Among the various technologies, automation, and in particular robotics, is playing a critical role

which is progressively gaining more and more importance. In the last decades, remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs) navigated the seas and have revolutionized the way we handle ocean exploration and subsea exploitation. Examples of such vehicles are shown in Figs. 1 and 2.

The simple difference between ROVs and AUVs, the absence of the tether, also defined as umbilical, opens the room to a huge technological gap. AUVs need to carry their own energy supplies, communication system, and onboard *intelligence* due to the limited underwater communication bandwidth. Today, approximately 200 kinds of AUVs are operational (Antonelli et al. 2016).

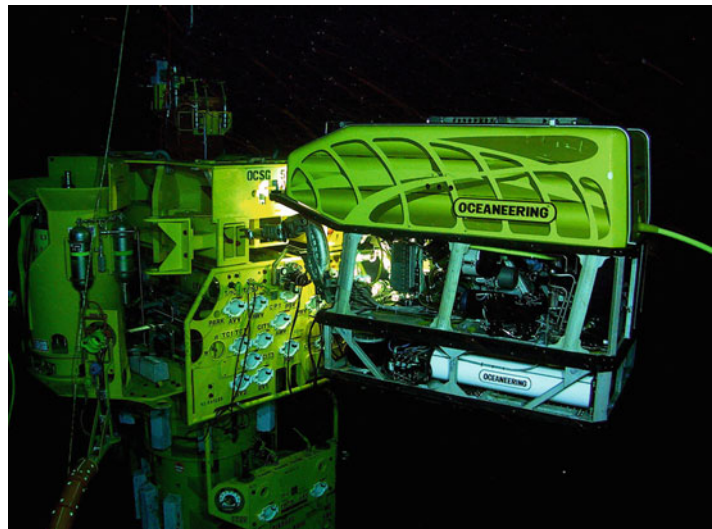
Sensor Systems

Underwater robots need to be equipped with two family of sensors, one group aimed at performing the mission it was planned for and the other needed to localize and control the vehicle itself.

Used sensors include sidescan and other sonars, thermistors, conductivity probes, fluorimeters, turbidity sensors, sensors to measure pH, and dissolved oxygen. In addition to that, compasses, gyroscopes, inertial measurement units, depth sensors, magnetometers, altitude

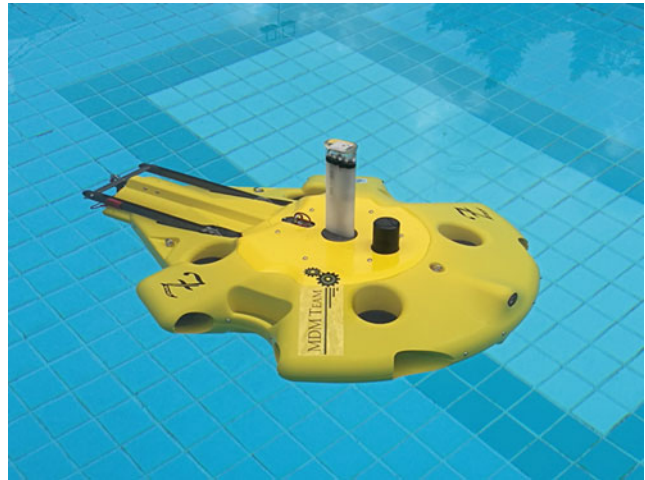
Underwater Robots,

Fig. 1 ROV at work in an underwater oil and gas field. (Credit: Frank van Mierlo)



Underwater Robots,

Fig. 2 Zeno AUV has been developed by the University of Florence (ISME node) and its spin-off company MDM Team SRL



and forward-looking sonars, Doppler velocity log, vision systems, and, finally, specific acoustic devices are needed for the localization.

Obviously, not all the sensors are available in each single vehicle, and there is a certain overlap in some of the measured variables. In order to properly use such sensors, advanced state estimate filters and fault detection and tolerance algorithms are implemented (Fossen 2002). From the dynamic systems perspective, in fact, it is worth noticing that most of the phenomenon are nonlinear; the sensors are multifrequency and often characterized by a time-varying delay.

Localization

As for any rigid body, even on the ground, there is no one single sensor that measures its position with respect to an inertial frame. In addition, underwater there is no Global Navigation Satellite System (GNSS) available. Localization, also defined as *navigation* in the underwater control community, is thus a crucial and demanding task, especially when large areas need to be travelled by the vehicle.

Due to the limited propagation of radio waves in the water, the most reliable localization methods are based on acoustic systems. Depending on the range, those are divided in long-baseline system (LBL), short-baseline

system (SBL), and ultrashort-baseline system (USBL). They are all based on the presence of a transceiver mounted on the vehicle and a certain number of transponders located in known positions. Distances among transceivers and transponders are measured upon the echo time delay. From this information, trilateration and state estimate techniques can be implemented to obtain the vehicle positions usually at rates of approximatively 1 Hz.

The systems above, commercially available, are reliable and expensive. The research is progressively moving toward cheapest techniques which do not require a surrounding infrastructure to measure the vehicle position or a GNSS-equipped surface boat.

It is worth remarking that the technique of computing the position integrating the velocity and/or the acceleration is subject to drift phenomena and can be used only for short range movements (the order of magnitude of the error is 1% of the distance travelled).

Actuation

ROVs and AUVs exhibit a certain variability in their actuation system depending on the application they have been designed for. The main propulsion is provided by thrusters or hydro-jets, the latter being limited in providing thrust in

one direction only. ROVs are often characterized by a structural pitch-roll stability and four, or more, thrusters are then aimed at controlling the remaining degrees of freedom. Since ROVs are man-driven, they are often controlled such that depth is decoupled from the motion on the horizontal plane, easier for the operator. AUVs mainly exhibit torpedolike shapes since they are commonly used for survey missions where the vehicle runs most of the time at a constant cruise velocity. In such a case, the dominant actuation design is composed by one or two thrusters parallel to the stern-bow direction and a fin and a rudder to steer the vehicle. The latter control surfaces, however, are ineffective at low/null velocities.

Guidance and Control

Following Fossen (1994), guidance has been identified as the actions of computing the course, attitude, and speed of the vehicle with respect to a proper reference frame. Control is the computation of the proper forces and moments acting at the vehicle needed to implement the required motion requested by the guidance system; those forces are finally properly converted into the required thrusters' values.

Guidance and control act at different levels; the former works at *velocity* level, i.e., it outputs the vehicle reference velocity which allows the vehicles itself to follow the desired trajectory, while the latter works at *force* level, i.e., it outputs the thrust required to follow the reference velocity coming from the guidance system.

Both the algorithms are implemented in a variety of methods and architectures strongly depending on the vehicle design.

Intervention

Some of the operations required underwater needs to interact with the environment by means of a robot, e.g., turn the valve of an underwater oil and gas infrastructure. To the purpose arm-equipped ROV or underwater

vehicle-manipulator systems (UVMSs) need to be used. The latter requires to design advanced control schemes to counteract the ocean current, to compensate for the gravity-buoyancy effects and due to the need to coordinates the vehicle and arm controllers to counteract the dynamic coupling (Antonelli 2018).

Summary and Future Directions

Manned subsea operations such as surveys or installation and maintenance of oil and gas structures are progressively being substituted by autonomous or automatic systems both for safety and economic reasons. It is likely that this trend will continue and robots will play an increasing role in underwater exploration and exploitation.

The benefit from the robotics research, in all its shades, will clearly influence also underwater robotics. To be more specific, underwater hybrid systems, i.e., alternatively wired and autonomous, may be expected in the future as well as subsea resident vehicle stations, also defined as underwater launch and recovery systems. The vehicles will be designed and programmed to work mainly autonomously, but they will have locations where charge their batteries and communicate at a large bandwidth by means of cabled communication infrastructures.

Cross-References

- ▶ [Advanced Manipulation for Underwater Sampling](#)
- ▶ [Control of Networks of Underwater Vehicles](#)
- ▶ [Control of Ship Roll Motion](#)
- ▶ [Dynamic Positioning Control Systems for Ships and Underwater Vehicles](#)
- ▶ [Mathematical Models of Marine Vehicle-Manipulator Systems](#)
- ▶ [Mathematical Models of Ships and Underwater Vehicles](#)
- ▶ [Motion Planning for Marine Control Systems](#)
- ▶ [Underactuated Marine Control Systems](#)

Recommended Reading

A short introduction to marine robotics may be found in Antonelli et al. (2016). Among the control-based books, the one of T. Fossen (1994) is one of the first dedicated to control problems of marine systems. The same author presents, in Fossen (2002), an updated and extended version of the topics developed in the first book and in Fossen (2011), an handbook on marine craft hydrodynamics and control. The monograph (Antonelli 2018) is totally devoted to UVMSs.

Bibliography

- Antonelli G (2018) Underwater robots. Springer tracts in advanced robotics, 4th edn. Springer, Heidelberg
- Antonelli G, Fossen T, Yoerger D (2016) Underwater robotics. In: Siciliano B, Khatib O (Eds) Springer handbook of robotics, 2nd edn. Springer, Heidelberg
- Fossen T (1994) Guidance and control of ocean vehicles. Chichester, New York
- Fossen T (2002) Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles. Marine Cybernetics, Trondheim
- Fossen T (2011) Handbook of marine craft hydrodynamics and motion control. Wiley, Chichester

Unmanned Aerial Vehicle (UAV)

Eric N. Johnson
Pennsylvania State University, University Park,
PA, USA

Abstract

The rapidly evolving field of unmanned aerial vehicles, drones, or, more simply, an aircraft without a pilot in it, is considered by many to be a cutting-edge technology. At the same time, it has a surprisingly long history. This article summarizes the history of the unmanned aircraft, deals with some of many definitions that confuse newcomers to this topic, and summarizes where things stand on the related technology and policy.

Keywords

Aerial robot · Autonomous aircraft · Drone · First person view · Optionally piloted vehicle · Remotely piloted vehicle · Unmanned aerial system

Introduction

The long history and recent rapidly expanding usage of unmanned aerial vehicles makes for an interesting mix. Sometimes a poster child for innovation, the field can also be a jumble of old and new terminology and definitions. This article will start by summarizing the history of the field. The subject of terminology and definitions will be dealt with head on, making clearer where the many terms relate. Following this, key attributes of these systems are described on how this relates to the underlying control technology. Finally, future trends are discussed.

History of Unmanned Aerial Vehicles

Generally referring to aircraft without people onboard, unmanned aerial vehicles (UAVs) have been around as long as aviation itself. Successful flight of unmanned versions of aircraft concepts have almost always occurred before human-carrying versions were successfully tested across all categories. This enabled risky flight testing to happen without risk to a human onboard – as well as enabling a smaller version to be evaluated prior to scaling up to something big enough to even carry a person. For example, the first flights with airplanes with no human onboard occurred before the first successful airplane flights in 1903 by the Wright brothers. In 1896, Samuel Pierpont Langley's team successfully flew a much smaller steam-powered airplane model more than 3/4 of a mile (Anderson 2008). The aircraft was not under any control during the flight – relying on inherent stability alone. All successful aircraft configurations have had both manned and unmanned examples in their histories, including airplanes,

helicopters, and powered lift and lighter-than-air aircraft.

Even early on, aircraft specifically designed to be unmanned found uses beyond flight research experiments. People have built and flown model aircraft for recreational purposes since nearly the beginning of aviation. This has provided generations with an enjoyable hobby, drone racing being the most recent addition.

World War I saw the emergence of airplane category UAVs. This led to both the cruise missile and the target drone. The former, normally not considered a UAV, has evolved ever since into highly capable weapon systems. The Kettering Bug was developed in the United States and flew in 1918. The German V-1 flying bomb became the first cruise missile used in large numbers in 1944–1945. The target drone *is* normally considered a UAV and is an aircraft intentionally shot down in order to train with or test antiaircraft weapons, a use that is probably the source of the word drone to refer to a UAV. These types of systems date back to 1935. The first important commercial use outside of the recreational

sector was aerial application (crop straying) for agriculture in Japan, which has been widespread and economically viable since the 1990s.

As the guidance and control capabilities of these aircraft improved, the practical applications of unmanned aircraft expanded, first for military use. That advent of satellite communications allowed UAVs to be remotely operated from across the globe. By the 1990s, systems such as the General Atomics MQ-1 Predator, Fig. 1, could be remotely operated for hours at a time while sending video back to its operators. In late 2001, the first systems capable of identifying, following, and then performing a precision strike on individuals from a single unmanned aircraft were put into active use.

In the early 2010s, an important convergence of trends occurred between battery and sensor technologies (both responding primarily to smartphone market pressures) that enabled much lower-cost multirotor UAVs that could obtain high-quality imagery. This created a large recreational photography and commercial market for purposes such as real estate photography and



Unmanned Aerial Vehicle (UAV), Fig. 1 MQ-1 predator unmanned aerial vehicle was an important early example of an armed reconnaissance system



Unmanned Aerial Vehicle (UAV), Fig. 2 DJI Phantom 4 is typical of present-day UAVs that can cheaply be used to perform aerial photography by the general public

surveying. The net result was that the number of UAVs being operated now rivals manned aircraft and these types of UAVs have become the average person's notion of a UAV today (e.g., DJI Phantom 4, Fig. 2). In this form, the UAV is now a common tool for cinematography, real estate photography, surveying, aerial light shows, and collecting data for scientific studies.

The ability to safely and economically operate a UAV has ignited considerable interest in many emerging applications for UAVs, such as pipeline monitoring, unexploded mine detection, chemical sensing, wildfire management, law enforcement, traffic monitoring, maritime patrol, border patrol, search and rescue, communication relays, cargo transport, passenger transport, and many more.

Unmanned Aerial Vehicle Terminology

The terminology around the UAV can create important distinctions in law (which can only be partially covered here), can be important to

communicate within the related technical communities, and now can even have important meanings and implications to the general public. Unfortunately, these communities have adopted somewhat different usage of terms.

Legal Definitions

The important distinction between a UAV and a (manned) aircraft is that there is no one on the aircraft that can take control of it. It is worth stating the full International Civil Aviation Organization (ICAO) definition here (ICAO 2011):

An unmanned aerial vehicle is a pilotless aircraft, in the sense of Article 8 of the Convention on International Civil Aviation, which is flown without a pilot-in-command on-board and is either remotely and fully controlled from another place (ground, another aircraft, space) or programmed and fully autonomous.

There is an interesting technicality here that an aircraft carrying someone incapable of taking control of it is probably a UAV, such as a device designed to move an unconscious person. Terms that can have similar meaning to UAV in some

jurisdictions/communities are remotely piloted aircraft (RPA), remotely piloted vehicle (RPV), and simply unmanned aircraft (UA). An aircraft that can be operated with or without a human pilot onboard is often referred to as an optionally piloted vehicle (OPV).

The facilities outside of the aircraft – which could include a remote pilot – are also an important part of the operation from a regulatory point of view. The legal term for the UAV plus these other controlling elements is called an unmanned aircraft system (UAS).

An important modifier here is the small UAS (sUAS), which typically refers to those that weigh less than 25 Kg. Worldwide regulations have been harmonizing through the ICAO to treat these smaller systems with typically lighter regulation, further enabling the successful applications that revolve around obtaining images from small aircraft at low altitude and short range. It is also common to distinguish UAVs through category (airplane, rotorcraft, powered lift, and others), class (seaplane vs. land plane, gyroplane vs. helicopter), and type (specific manufacturer and model).

The use of the word “unmanned” in UAV/UAS can cause confusion. The term “manned” normally means that something “has a crew.” In the case of a UAS, there could indeed still be a crew – but they are simply not onboard the aircraft. In this sense, “remote pilot” is the more precise term, if less often used.

Other Common Usage

The terms UAV and UAS have not been universally adopted outside of the aviation community. An important term in common use is the aerial robot, which simply distinguishes a robot that can fly from other types of robots. The adjective “autonomous” is often included to indicate a level of guidance and control automation high enough to deal with uncertainties in the environment. The term multirotor (Fig. 2) is one often used for aircraft consisting of three or more vertically mounted motors used to achieve vertical flight. Quadrotor or quadcopter refer to the specific case of a multirotor with four propellers. A powered lift category aircraft that includes both multirotor

elements and airplane features like wings is often called a separate lift and thrust (SLT) aircraft (Fig. 3). When an aircraft is flown by the remote pilot through video transmitted from the aircraft, this is called first-person view (FPV) flying.

To the general public, the term “drone” is often used today to describe most if not all types of UAVs, as well as (sometimes) multirotor aircraft that carry people – possibly due to a conflation of the terms drone and quadrotor.

Unmanned Aerial Vehicle Capabilities

An unmanned aircraft has important capabilities quantified just as they are for a manned aircraft, including parameters such as speed, payload, maximum altitude, endurance, range, and cost. When the aircraft is controlled off-board, one will typically also focus on additional important parameters such as communication capability and level of automation.

The Implications of Being Unmanned

A fundamental challenge for developing a UAV is achieving a desired level of reliability without a human pilot onboard. A UAV may rely on pilot inputs via a wireless communication system. Such a system is subject to failure. Many successful systems provide the option of automated control and guidance to account for potential loss of communication. The result could be a return to a specified location and landing. For the simplest systems, it may simply have its propulsion system cut off, and it will descend and impact the ground, thus hopefully preventing damage to other parties.

A second challenge associated with not having a human pilot onboard is the capabilities of the human eye. Relaying video to a remote human pilot inevitably degrades vision. Today, the ability of a human pilot to see outside is a major contributor to aircraft navigation and collision avoidance effectiveness and reliability, from both a technical and a regulatory sense. Predicting the level of safety of an aircraft system without a human eye onboard for these functions is an important open question.



Unmanned Aerial Vehicle (UAV), Fig. 3 Arcturus Jump is an example of a powered-lift category UAV, including both quadrotor and airplane configuration elements

Automation and Autonomy

Perhaps even more so than conventional aircraft, the mission capabilities and reliability of a UAS are strongly dependent on its automation and autonomy capabilities achieved through control systems. The developments in this area have enabled the recent expansion of recreational and commercial use of UAV and are similarly expected to be critical to emerging and future uses.

Manual Control with Stability Augmentation: The majority of existing UAVs have some provision for remote manual control. Many systems today also provide enough stability and control augmentation through their onboard sensors to make the handling qualities good enough that even an inexperienced pilot can effectively perform normal flight maneuvers. That is, feedback of aircraft motion added to the pilot input to alter the flying qualities of the aircraft. The important sensors that have enabled this include inexpensive gyros, accelerometers, and Global Positioning System (GPS) receivers.

Navigation and trajectory following: The use of satellite navigation (GPS) has also enabled even low-cost aircraft to have a viable position estimate (at least when operating outside away from structures), often utilizing a Kalman filter to produce a complete navigation solution. With this navigation capability, even many low-cost small systems today are able to effectively fly a prescribed course, such as to produce a survey of a chosen area of land or move along a path to a desired destination. Emerging systems can automatically estimate position utilizing other sensors, such as laser scanners and cameras.

Path planning: A variety of methods have been developed for automatic path planning or trajectory generation; and UAVs are an important application area for these algorithms, that is, coming up with a path automatically for the aircraft to fly based on information derived onboard. An important example available today is *obstacle avoidance*, where systems can automatically revise their planned future flight path upon detection of obstacles to that path. Common

sensor choices for this include radar, laser scanning, and stereo camera systems.

Another important desired path planning capability is that for *collision avoidance* and *traffic management*. This is an area of active research between UAVs they may encounter each other in flight. Perhaps even more significant would be to address the interoperability between UAVs and manned aircraft. Without the pilot's human eye onboard, a UAV is not able to meet existing requirements for flight among manned aircraft without having someone able to see the UAV (e.g., a ground observer able to detect a conflict). This is an important area of active work, looking at communicated aircraft data from participating aircraft as well as sensing other aircraft with airborne radar and cameras.

Summary and Future Directions

There has been a dramatic increase in the actual commercial use of the UAV in the past decade. At the same time, there is considerable activity going into many potential new uses of this technology, including package delivery, passenger carrying, and more.

Regulations have evolved significantly in response to the challenge; but much more needs would need to be done to fully utilize even the existing technology. Suitable air traffic management, interoperability with manned aircraft, and a suitable collision avoidance paradigm are all open areas actively being worked on today.

Although there is considerable safe and economic commercial use of UAS, it is important to realize that this is limited to small systems flown close to the ground near their operators and the achieved reliability of these operations is only acceptable due to low cost and low danger of these systems failing. To truly enable the bigger, faster, and more useful systems to do even more useful work, there are problems to solve related to the achieved reliability.

From the technology side, there are still more applications to be enabled. There is work today in the fields of aerial manipulation and docking

that should enable future systems to perform useful work in many difficult to access locations (Bonyan Khamseh et al. 2018). Military applications include swarming UAVs to create a capability greater than the sum of parts. An important emerging field is counter-UAS (C-UAS) to allow one to prevent a UAV from entering a chosen airspace.

Cross-References

- [Aircraft Flight Control](#)
- [Tactical Missile Autopilots](#)
- [Trajectory Generation for Aerial Multicopters](#)

Bibliography

- Anderson JD (2008) Introduction to flight. McGraw-Hill, New York
- Bonyan Khamseh H, Janabi-Sharifi F, Abdessameud A (2018) Aerial manipulation—a literature survey. *Robot Auton Syst* 107:221–235. <https://doi.org/10.1016/j.robot.2018.06.012>
- ICAO Cir 328 (2011) Unmanned Aircraft Systems. <https://www.icao.int/Meetings/UAS/Documents/Circular%20328.en.pdf>

Use of Gaussian Processes in System Identification

Simo Särkkä

Aalto University, Helsinki, Finland

Abstract

Gaussian processes are used in machine learning to learn input-output mappings from observed data. Gaussian process regression is based on imposing a Gaussian process prior on the unknown regressor function and statistically conditioning it on the observed data. In system identification, Gaussian processes are used to form time series prediction models such as nonlinear

finite-impulse response (NFIR) models as well as nonlinear autoregressive (NARX) models. Gaussian process state-space (GPSS) models can be used to learn the dynamic and measurement models for a state-space representation of the input-output data. Temporal and spatiotemporal Gaussian processes can be directly used to form regressor on the data in the time domain. The aim of this article is to briefly outline the main directions in system identification methods using Gaussian processes.

Keywords

Gaussian process regression · Nonlinear system identification · GP-NFIR model · GP-ARX model · GP-NOE model · Gaussian process state-space model · Temporal Gaussian process · State-space Gaussian process

Introduction

Gaussian process regression (Rasmussen and Williams 2006) refers to a statistical methodology where we use Gaussian processes as prior models for regression functions that we fit to observed data. This kind of methodology is particularly popular in machine learning although the origins of the basic ideas can be traced to geostatistics (Cressie 1993). In geostatistics, the corresponding methodology is called “kriging” which is named after South African mining engineer D. G. Krige. In system identification, Gaussian processes can be used to identify (or “learn” in machine learning terms) the input-output relationships from observed data. Even when there is no explicit input in the system, Gaussian processes can be used to identify a model for an observed time series of outputs which is a specific form of a system identification problem. Overviews of the use of Gaussian processes in system identification can be found, for example, in the monograph of Kocijan (2016) and PhD theses of McHutchon (2015) and Frigola (2016).

Gaussian Processes in System Identification

Gaussian Process Regression

Gaussian process regression is concerned with the following problem: Given a set of observed (training) input-output data $\mathcal{D} = \{(\mathbf{z}_k, y_k) : k = 1, \dots, N\}$ from an unknown function $y = f(\mathbf{z})$, predict the values of the function at new (test) inputs $\{\mathbf{z}_k^* : k = 1, \dots, M\}$. That is, the problem is a classical regression problem. However, the classical solution to the problem usually amounts to fixing a parametric function class $f(\mathbf{z}; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a set of parameters, and then fitting the parameters to the observed data. In Gaussian process regression, we take a different route; instead of fixing a parametric class of functions, we put a Gaussian process prior measure on the whole regression function and condition on the observed data using Bayes’ rule (Rasmussen and Williams 2006).

Gaussian Process Regression Problem

Mathematically, the Gaussian process regression problem can be written as

$$f(\mathbf{z}) \sim \mathcal{GP}(m(\mathbf{z}), k(\mathbf{z}, \mathbf{z}')), \quad (1a)$$

$$y_k = f(\mathbf{z}_k) + \epsilon_k, \quad \epsilon_k \sim \mathcal{N}(0, \sigma_n^2), \quad (1b)$$

where Eq. (1a) tells that, a priori, the function is a Gaussian process with mean function $m(\mathbf{z}) = \mathbb{E}[f(\mathbf{z})]$ and covariance function (or kernel) $k(\mathbf{z}, \mathbf{z}') = \text{Cov}[f(\mathbf{z}), f(\mathbf{z}')] = \mathbb{E}[(f(\mathbf{z}) - m(\mathbf{z}))(f(\mathbf{z}') - m(\mathbf{z}'))]$. Equation (1b) tells that we observe the function values at points $\mathbf{z}_k$, $k = 1, \dots, N$ and that they are corrupted by (independent) Gaussian noises with variance σ_n^2 .

The mean and covariance functions define the regressor function class, and they, or at least their parametric classes, need to be selected a priori. The mean function can typically be selected to be identically zero $m(\mathbf{z}) = 0$. The covariance function defines the smoothness properties of the functions, and a typical choice in machine learning is the squared exponential covariance function

$$k(\mathbf{z}, \mathbf{z}') = s^2 \exp\left(-\frac{\|\mathbf{z} - \mathbf{z}'\|^2}{2\ell^2}\right) \quad (2)$$

which produces infinitely differentiable (i.e., analytic) regressor functions. The parameters s and ℓ in the aforementioned covariance function define the magnitude and length scales of the regressor functions, respectively. Other common choices of covariance functions are, for example, the Matérn class of covariance functions (Matérn 1960; Rasmussen and Williams 2006).

Gaussian Process Regression Equations

Given the mean and covariance functions as well as the measurements, we can form the Gaussian process regressor. Assuming that the noises are independent of the function values, we can write the joint distribution of the observed values and the unknown function values as follows:

$$\begin{pmatrix} \mathbf{y} \\ f(\mathbf{Z}^*) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m(\mathbf{Z}) \\ m(\mathbf{Z}^*) \end{pmatrix}, \begin{pmatrix} (\mathbf{K} + \sigma_n^2 \mathbf{I}) & \mathbf{k}^\top(\mathbf{Z}^*) \\ \mathbf{k}(\mathbf{Z}^*) & k(\mathbf{Z}^*, \mathbf{Z}^*) \end{pmatrix}\right), \quad (3)$$

where $\mathbf{y} = (y_1 \dots y_N)^\top$, $m(\mathbf{Z}) = (m(\mathbf{z}_1) \dots m(\mathbf{z}_N))^\top$, $m(\mathbf{Z}^*) = (m(\mathbf{z}_1^*) \dots m(\mathbf{z}_M^*))^\top$, and $\mathbf{K}$ and $\mathbf{k}(\mathbf{Z}^*)$ denote matrices with element (i, j) given as $k(\mathbf{z}_i, \mathbf{z}_j)$ and $k(\mathbf{z}_i^*, \mathbf{z}_j)$, respectively. By conditioning this joint Gaussian distribution on the measurements $\mathbf{y}$, we get that the conditional

(i.e., posterior) distribution of the function values $f(\mathbf{Z}^*) = (f(\mathbf{z}_1^*) \dots f(\mathbf{z}_M^*))^\top$ is Gaussian with the mean and covariance

$$\mathbb{E}[f(\mathbf{Z}^*) | \mathbf{y}] = m(\mathbf{Z}^*) + \mathbf{k}(\mathbf{Z}^*)$$

$$\left[\mathbf{K} + \sigma_n^2 \mathbf{I}\right]^{-1} (\mathbf{y} - m(\mathbf{Z})),$$

$$\text{Cov}[f(\mathbf{Z}^*) | \mathbf{y}] = k(\mathbf{Z}^*, \mathbf{Z}^*)$$

$$- \mathbf{k}(\mathbf{Z}^*) \left[\mathbf{K} + \sigma_n^2 \mathbf{I}\right]^{-1} \mathbf{k}^\top(\mathbf{Z}^*). \quad (4)$$

These are the fundamental equations of Gaussian process regression. An example of Gaussian process regression with squared exponential covariance function is shown in Fig. 1.

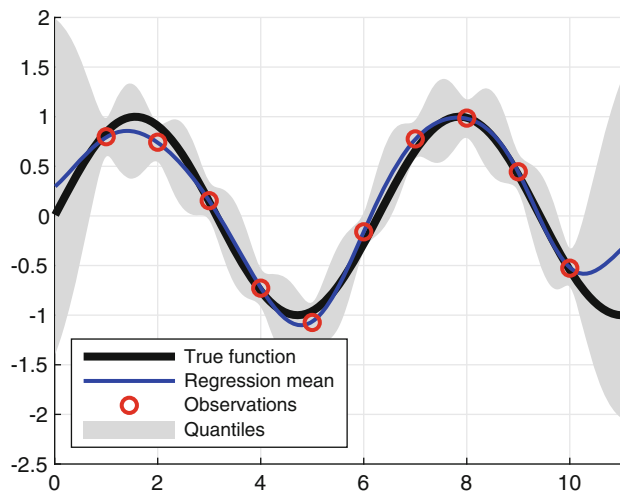
Hyperparameter Learning

Even though Gaussian process regression is a nonparametric method, for which we do not need to fix a parametric class of functions, the mean and covariance functions can have unknown hyperparameters $\boldsymbol{\varphi}$ which can be estimated from data. For example, the squared exponential covariance function in Eq. (2) has the hyperparameters $\boldsymbol{\varphi} = (s, \ell)$.

A common way to estimate the parameters is to maximize the marginal likelihood – also called evidence – $p(\mathbf{y} | \boldsymbol{\varphi})$ of the measurements or, equivalently, minimize the negative log-likelihood of the measurements

Use of Gaussian Processes in System Identification, Fig. 1

Example of Gaussian process regression with squared exponential covariance function. The true function is a sinusoidal which is observed only at 10 points that are corrupted by Gaussian noise. The quantiles provide error bars for the predicted function values



$$\begin{aligned}
-\log p(\mathbf{y} | \boldsymbol{\varphi}) &= \frac{1}{2} \log |2\pi (\mathbf{K}_{\boldsymbol{\varphi}} + \sigma_n^2 \mathbf{I})| \\
&+ \frac{1}{2} (\mathbf{y} - m_{\boldsymbol{\varphi}}(\mathbf{Z}))^\top [\mathbf{K}_{\boldsymbol{\varphi}} + \sigma_n^2 \mathbf{I}]^{-1} \\
&\times (\mathbf{y} - m_{\boldsymbol{\varphi}}(\mathbf{Z})).
\end{aligned} \quad (5)$$

The gradient of this function with respect to the hyperparameters is also available (see, e.g., Rasmussen and Williams 2006) which allows for the use of gradient-based optimization methods to estimate the parameters.

Instead of using the maximum likelihood method to estimate the parameters, it is also possible to use a Bayesian approach to the problem and consider the posterior distribution of the hyperparameters

$$p(\boldsymbol{\varphi} | \mathbf{y}) = \frac{p(\mathbf{y} | \boldsymbol{\varphi}) p(\boldsymbol{\varphi})}{\int p(\mathbf{y} | \boldsymbol{\varphi}) p(\boldsymbol{\varphi}) d\boldsymbol{\varphi}}, \quad (6)$$

where $p(\boldsymbol{\varphi})$ is the prior distribution of the hyperparameters. We can, for example, compute the maximum a posteriori estimate of the parameters by finding the maximum of this distribution or use Markov chain Monte Carlo (MCMC) methods (Brooks et al. 2011) to estimate the statistics of the distribution.

In what follows, to avoid notational clutter, we drop out the hyperparameters from the Gaussian process formulations and inference methods although they are commonly estimated as part of the Gaussian process learning.

Reduction of Computational Complexity

A limitation of Gaussian process regression in its explicit form is that the computational complexities of the regression Equations (4) and likelihood Equation (5) are cubic $\mathcal{O}(N^3)$ in the number of measurements N . This is due to the $N \times N$ matrix inversion appearing in the equations which, even when implemented with Cholesky or LU decompositions, needs a cubic number of computational steps.

Ways of solving the computational complexity problem are, for example, sparse approximations using inducing points (Quiñonero-Candela and Rasmussen 2005; Rasmussen and Williams 2006;

Titsias 2009), approximating the problem with a discrete Gaussian random field model (Lindgren et al. 2011), or use of random or deterministic basis/spectral expansions (Quiñonero-Candela et al. 2010; Solin and Särkkä 2018).

GP-NFIR, GP-NARX, GP-NOE, and Related Models

In system identification, we can use Gaussian processes to model unknown input-output relationships in time series. Several different model architectures are available for this purpose. Let us assume that we have a system with input sequence $u_1, u_2, \dots$ and output sequence $y_1, y_2, \dots$ and the aim is to predict the outputs from inputs. We also assume that we have been given a set of training data consisting of known inputs and (noisy) outputs. In the following, we present some typically used architectures that have been proposed for this purpose. More details can be found in the monograph of Kocijan (2016).

GP-NFIR Model

The Gaussian process nonlinear finite-impulse response (GP-NFIR) model (Ackermann et al. 2011; Kocijan 2016) has the form (see Fig. 2)

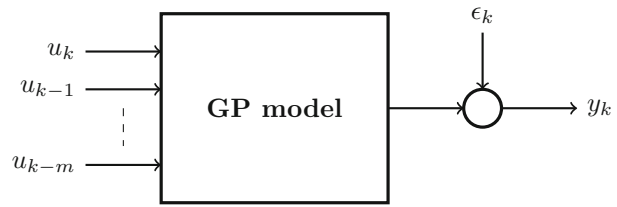
$$\hat{y}_k = f(u_{k-1}, \dots, u_{k-m}), \quad (7)$$

where $f(\cdot)$ is an unknown mapping which we model as a Gaussian process and $\hat{y}_k$ denotes the estimate produced by the regressor. In this model, we form a Gaussian process regressor that predicts the current output from a finite number of previous inputs. This model can be identified by reducing it into a Gaussian process regression model

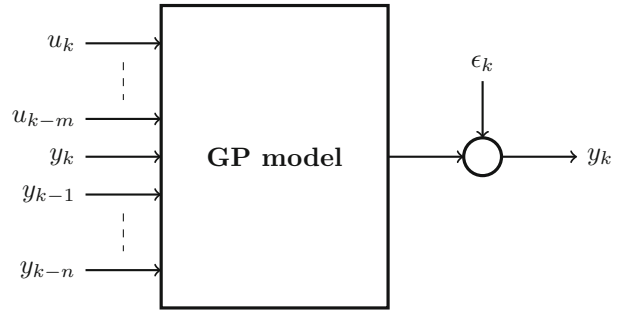
$$y_k = f(\mathbf{z}_k) + \epsilon_k, \quad (8a)$$

where $\mathbf{z}_k = (u_{k-1} \dots u_{k-m})^\top$ and by using standard Gaussian process regression methods on it.

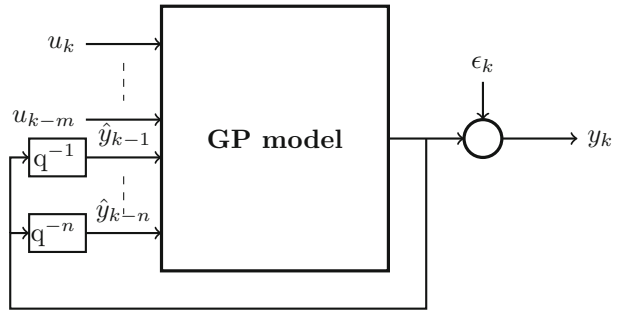
Use of Gaussian Processes in System Identification, Fig. 2 In GP-NFIR model the Gaussian process regressor is used to predict next output from previous inputs



Use of Gaussian Processes in System Identification, Fig. 3 In GP-NARX model the Gaussian process is used to predict the next output from the previous inputs and outputs



Use of Gaussian Processes in System Identification, Fig. 4 The GP-NOE model uses the previous inputs and the previous outputs of the Gaussian process regressor to predict the next output. In the figure, q^{-n} denotes an n -step delay operator



GP-NARX Model

The Gaussian process nonlinear autoregressive model with exogenous input (GP-NARX) (Kocijan et al. 2005; Kocijan 2016) is a model of the form (see Fig. 3)

$$y_k = f(y_{k-1}, \dots, y_{k-n}, u_{k-1}, \dots, u_{k-m}) + \epsilon_k, \quad (9)$$

where ϵ_k is a Gaussian random variable. This model can be reduced to a Gaussian process regression problem by setting $\mathbf{z}_k = (y_{k-1} \cdots y_{k-n} \ u_{k-1} \cdots u_{k-m})^\top$ in Eq. (8a).

GP-NOE Model

In Gaussian process nonlinear output error (GP-NOE) model (Kocijan and Petelin 2011; Kocijan 2016), we form a Gaussian process regressor for the problem (see Fig. 4)

$$y_k = f(\hat{y}_{k-1}, \dots, \hat{y}_{k-n}, u_{k-1}, \dots, u_{k-m}) + \epsilon_k, \quad (10)$$

where $\hat{y}_{k-1}, \dots, \hat{y}_{k-n}$ are the Gaussian process regressor predictions from the previous steps.

Learning in this kind of model requires further approximations because the predictions of the Gaussian process are directly used as inputs on the next step.

Other Model Architectures

As discussed in Kocijan (2016), it is also possible to extend these architectures to, for example, GP-NARMAX (nonlinear autoregressive and moving average model with exogenous input) models and NJB (nonlinear Box–Jenkins) models.

Gaussian Process State-Space (GPSS) Models

Another approach to system identification is to form a state-space model where the dynamic and

measurement models are identified using Gaussian process regression methods. This leads to so-called Gaussian process state-space models.

General GPSS Model

A Gaussian process state-space (GPSS) model (see Fig. 5) has the mathematical form (e.g., Kocijan 2016)

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, u_k) + \mathbf{w}_k, \quad (11a)$$

$$y_k = g(\mathbf{x}_k, u_k) + \epsilon_k, \quad (11b)$$

where the state vector $\mathbf{x}_k$, $k = 0, 1, 2, \dots, N$ contains the current state of the system, $u_1, u_2, \dots$ is the input sequence, and $y_1, y_2, \dots$ is the output sequence. In the model, $\mathbf{w}_k$ is a Gaussian distributed process noise. The aim is now to learn the functions $f(\mathbf{x}_k, u_k)$ and $g(\mathbf{x}_k, u_k)$, which are modeled as Gaussian processes, given the input and output sequences or, in some cases, also given direct observations of the state vector.

Learning with Fully Observed State

When the state vector $\mathbf{x}_k$ is fully observed, then both the dynamic model (11a) and measurement model (11b) become standard Gaussian process regression models. In dynamic model (11a), the training set consists of measurements $\mathbf{x}_{k+1}$ with the corresponding inputs $(\mathbf{x}_k, u_k)$, and in measurement model (11b), the measurements are y_k with the corresponding inputs $(\mathbf{x}_k, u_k)$. This kind of fully observed models are important in many applications such as robotics (Deisenroth et al. 2015).

After conditioning on the training data, the functions f and g will still be Gaussian processes, and their mean and covariance functions are given by multivariate generalizations of Eqs. (4). State estimation in this kind of models has been considered by Ko and Fox (2009) and Deisenroth et al. (2011), and it turns out that it is possible to construct closed-form Gaussian approximation (moment matching)-based filters and smoothers for these models (Deisenroth et al. 2011). Control problems related to this kind of models have been considered, for example, by Deisenroth et al. (2015).

Marginalization of the GP

When the states $\mathbf{x}_k$ are not observed, then we need to treat both the states and the Gaussian processes as unknown. There are a few different ways to cope with the model in that case, and one approach is the marginalization approach of Frigola et al. (2013, 2014a, b) and Frigola (2016). First note that if we have a method to learn f , we can learn both f and g using a state-augmentation trick (Frigola 2016): we define an augmented state as $\tilde{\mathbf{x}} = (\mathbf{x} \ \gamma)^\top$, Gaussian process $h(\boldsymbol{\gamma}, u) = (f(\mathbf{x}, u) \ g(\mathbf{x}, u))^\top$, and augmented process noise $\tilde{\mathbf{w}} = (\mathbf{w}_k \ \mathbf{0})^\top$, which reduces the model to

$$\tilde{\mathbf{x}}_{k+1} = h(\tilde{\mathbf{x}}_k, u_k) + \tilde{\mathbf{w}}_k, \quad (12a)$$

$$y_k = \gamma_k + \epsilon_k. \quad (12b)$$

This is now a model with an unknown dynamic model but with a given linear Gaussian measurement model $p(y_k | \tilde{\mathbf{x}}_k, u_k) = \mathcal{N}(y_k | \gamma_k, \sigma_n^2)$.

Thus, without a loss of generality, we can focus on models with an unknown dynamic model f and a known measurement model:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, u_k) + \mathbf{w}_k, \quad (13a)$$

$$y_k \sim p(y_k | \mathbf{x}_k, u_k). \quad (13b)$$

The aim is to learn the function $f(\mathbf{x})$ from a sequence of measurement data y_k .

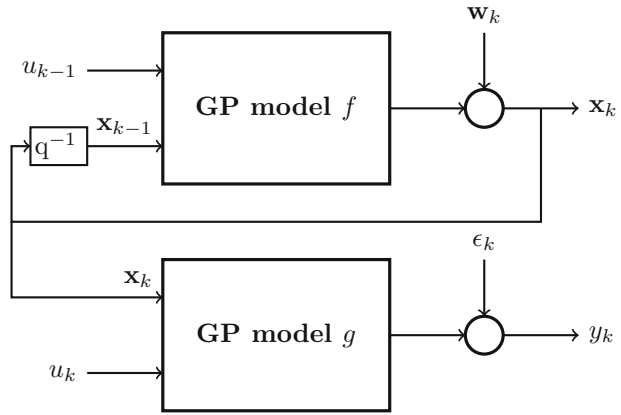
One way to understand (Frigola 2016) this model is that we could hypothetically generate data from it by sampling the (infinite-dimensional) function $f(\mathbf{x}, u)$ and then starting from $\mathbf{x}_0$ sequentially produce each $\{\mathbf{f}_1, \mathbf{x}_1, \dots, \mathbf{f}_N, \mathbf{x}_N\}$, where we have denoted $\mathbf{f}_k = f(\mathbf{x}_{k-1}, u_{k-1})$ for $k = 1, 2, \dots, N$. Each of the conditional distributions

$$p(\mathbf{f}_k | \mathbf{f}_{1:k-1}, \mathbf{x}_{0:k-1}) \quad (14)$$

turns out to be Gaussian. Note that above, we have introduced the short-hand notation $\mathbf{f}_{1:k} = (\mathbf{f}_1, \dots, \mathbf{f}_k)$ which we will use also in the rest of this entry.

The above observation allows us to integrate out (i.e., marginalize) the Gaussian process from

Use of Gaussian Processes in System Identification, Fig. 5 In GPSS model we learn a Gaussian process regressor for approximating the dynamic and measurement models in a state-space model. In this figure, q^{-1} denotes a one-step delay operator



the model in closed form. The result is the following representation (Frigola et al. 2013):

$$p(\mathbf{x}_{0:N}) = \prod_{k=1}^N \mathcal{N}(\mathbf{x}_k \mid \boldsymbol{\mu}_k(\mathbf{x}_{0:k-1}), \boldsymbol{\Sigma}_k(\mathbf{x}_{0:k-1})), \quad (15)$$

where the means $\boldsymbol{\mu}_k(\mathbf{x}_{0:k-1})$ and covariances $\boldsymbol{\Sigma}_k(\mathbf{x}_{0:k-1})$ are (quite complicated) functions of the prior mean and covariance functions of the Gaussian process f , which are evaluated on the whole previous state history. The above equation defines a non-Markovian prior model for the state sequence $\mathbf{x}_k$, $k = 1, \dots, N$.

For a given $\mathbf{x}_{0:N}$, the distribution $p(f(\mathbf{x}^*) \mid \mathbf{x}^*, \mathbf{x}_{0:N})$ for a test point $\mathbf{x}^*$ can be computed by using conventional Gaussian process prediction equation as follows:

$$p(f(\mathbf{x}^*) \mid \mathbf{x}^*, y_{1:N}) = \int p(f(\mathbf{x}^*) \mid \mathbf{x}^*, \mathbf{x}_{0:N}) p(\mathbf{x}_{0:N} \mid y_{1:N}) d\mathbf{x}_{0:N}, \quad (16)$$

which can be numerically approximated, provided that we use a convenient (e.g., Monte Carlo) approximation for $p(\mathbf{x}_{0:N} \mid y_{1:N})$.

Given the model (15), it is then possible to use, for example, particle Markov chain Monte Carlo methods (Frigola et al. 2013) to sample state trajectories from the posterior distribution $p(\mathbf{x}_{0:N} \mid y_{1:N})$ jointly with the parameters of the model, which provides a Monte Carlo

approximation to the above integral. Other proposed approaches are, for example, particle stochastic approximation expectation–maximization (EM, Frigola et al. 2014b) which uses a Monte Carlo approximation to the EM algorithm aiming at computing the maximum likelihood estimates of the parameters while handling the states as missing data.

Approximation of the GP

Another way to approach the problem where both the states and Gaussian processes are unknown is to approximate the Gaussian process as finite-dimensional parametric model and use conventional parameter estimation methods to the model.

One possible approximation considered in Svensson et al. (2016) is to employ a Karhunen–Loeve type of basis function expansion of the Gaussian process as follows:

$$f(\mathbf{x}, \mathbf{u}) = \sum_{i=1}^S c_i \phi_i(\mathbf{x}, \mathbf{u}), \quad (17)$$

where $\phi_i(\mathbf{x}, \mathbf{u})$ are deterministic basis functions (e.g., sinusoids) and c_i are Gaussian random variables. With this approximation, the model in Eq. (13b) becomes

$$\mathbf{x}_{k+1} = \sum_{i=1}^S c_i \phi_i(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k, \quad (18a)$$

$$y_k \sim p(y_k \mid \mathbf{x}_k, u_k), \quad (18b)$$

where learning of the Gaussian process f reduces to estimation of the finite number of parameters $\mathbf{c} = (c_1 \cdots c_S)^\top$ in the state-space model. The states and parameters in this model can now be determined using, for example, particle Markov chain Monte Carlo (PMCMC) methods (see Svensson et al. 2016).

Another possibility is to use inducing points (typically denoted with $\mathbf{u}$, but here we denote them with $\mathbf{f}^u$ to avoid confusion with the input sequence). In those approaches the idea is to first perform Gaussian process inference on the inducing points alone, that is, compute $p(\mathbf{f}^u \mid y_{1:N})$ and then compute the (approximate) predictions by conditioning on the inducing points instead of the original data. This leads to the approximation

$$\begin{aligned} p(f(\mathbf{x}^*) \mid y_{1:N}) &= \int p(f(\mathbf{x}^*) \mid \mathbf{x}^*, \mathbf{x}_{0:N}) \\ &\quad p(\mathbf{x}_{0:N} \mid y_{1:N}) d\mathbf{x}_{0:N} \\ &\approx \int p(f(\mathbf{x}^*) \mid \mathbf{x}^*, \mathbf{f}^u) \\ &\quad p(\mathbf{f}^u \mid y_{1:N}) d\mathbf{f}^u. \end{aligned} \quad (19)$$

In Turner et al. (2010), the inducing points are learned using expectation–maximization (EM) algorithm. Frigola et al. (2014a) propose a method for variational learning (or integration over) of the inducing points by forming a variational Bayesian approximation $q(\mathbf{f}^u) \approx p(\mathbf{f}^u \mid y_{1:N})$, which further results in

$$\begin{aligned} p(f(\mathbf{x}^*) \mid y_{1:N}) \\ \approx \int p(f(\mathbf{x}^*) \mid \mathbf{x}^*, \mathbf{f}^u) q(\mathbf{f}^u) d\mathbf{f}^u \end{aligned} \quad (20)$$

and turns out to be analytically tractable as the optimal variational distribution $q(\mathbf{f}^u)$ is Gaussian.

Spatiotemporal Gaussian Process Models

Temporal Gaussian Processes

Another way of modeling time series using Gaussian processes is by considering them as functions of time (Hartikainen and Sarkka 2010) which are sampled at certain time instants $t_1, t_2, \dots$:

$$\begin{aligned} f(t) &\sim \mathcal{GP}(m(t), k(t, t')), \\ y_k &= f(t_k) + \epsilon_k. \end{aligned} \quad (21)$$

That is, instead of attempting to form a predictor from the previous measurements or inputs, the idea is to condition the temporal Gaussian process on its observed values and use the conditional Gaussian process for predicting values at new time points.

Unfortunately, due to the cubic computational scaling of the Gaussian process regression, this quickly becomes intractable when time series length increases. However, temporal Gaussian process regression is closely related to the classical Kalman filtering and Rauch–Tung–Striebel smoothing problems (e.g., Särkkä 2013), which can be used to reduce the required computations. It turns out that provided the Gaussian process is stationary, that is, the covariance function only depends on the time difference $k(t, t') = k(t - t')$, then, under certain restrictions, the Gaussian process regression problem is essentially equivalent to state estimation in a model of the form

$$\begin{aligned} \frac{d\mathbf{x}(t)}{dt} &= \mathbf{A} \mathbf{x}(t) + \mathbf{B} \boldsymbol{\eta}(t), \\ y_k &= \mathbf{C} \mathbf{x}(t_k) + \epsilon_k, \end{aligned} \quad (22)$$

where $\boldsymbol{\eta}(t)$ is a white noise process and the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are selected suitably to match the original covariance function. For example, the Matérn covariance functions with half-integer smoothness parameters have exact representations as state-space models (Hartikainen and Sarkka 2010).

The Gaussian process regression problem can be now solved by applying a Kalman filter and Rauch–Tung–Striebel smoother on this problem. These methods have the fortunate property that their complexity is linear $\mathcal{O}(N)$ with respect to the number of measurements N as opposed to the cubic complexity of direct Gaussian process regression solution.

Spatiotemporal Gaussian Processes

A similar state-space approach also works for spatiotemporal Gaussian process models with

a covariance function of a stationary form $k(\mathbf{z}, t; \mathbf{z}', t') = k(\mathbf{z}, \mathbf{z}'; t - t')$. In that case, the state-estimation problem becomes infinite dimensional, that is, a distributed parameter system:

$$\begin{aligned} \frac{d\mathbf{x}(\mathbf{z}, t)}{dt} &= \mathcal{A} \mathbf{x}(\mathbf{z}, t) + \mathbf{B} \boldsymbol{\eta}(\mathbf{z}, t), \\ y_k &= \mathcal{C} \mathbf{x}(\mathbf{z}, t_k) + \epsilon_k, \end{aligned} \quad (23)$$

where $\mathcal{A}$ is a matrix of operators and $\mathcal{C}$ is a matrix of functionals (Särkkä et al. 2013). The solution of the Gaussian process regression problem in this form requires the use of methods from partial differential equations, but in many cases we can obtain an exact $\mathcal{O}(N)$ inference procedure from this route.

Latent Force Models

In so-called latent force models (Álvarez et al. 2013), the idea is to infer latent force $\xi(t)$ in a differential equation model such as

$$\frac{d^2 x(t)}{dt^2} + \gamma \frac{dx(t)}{dt} + v^2 = \xi(t), \quad (24)$$

where $\xi(t)$ is an unknown function which is modeled as a Gaussian process. The inference in this model can be recast as Gaussian process regression with a modified covariance function. The idea can be further generalized to partial differential equation models and nonlinear models when approximation methods such as Laplace approximation are used.

The inference in this kind of models can be further restated in the state-space form (Särkkä et al. 2019), which also allows for the study of control problems on latent force models. This formulation also allows the analysis of observability and controllability properties of latent force models.

classes: (1) GP-NFIR, GP-NARX, and GP-NOE type of models which directly aim at learning the function from the previous inputs and outputs to the current output, (2) Gaussian process state-space models which aim to learn the dynamic and measurement models in the state-space model, and (3) Gaussian process regression methods for the spatiotemporal time series by direct or state-space Gaussian process regression.

Active problems in research in this area appear to be, for example, joint learning and control in all the model types outlined here. Another direction of further study is the analysis of observability, identifiability, and controllability. In practice, the Gaussian process-based system identification methods are very similar to classical methods, and hence they can be expected to inherit many limitations and theoretical properties of the classical methods.

An emerging research area in Gaussian process-based models is the so-called deep Gaussian processes (Damianou and Lawrence 2013) which borrow the idea of deep neural networks by forming hierarchies of Gaussian processes. This kind of models could also turn out to be useful in system identification. Furthermore, Gaussian processes are also easily combined with first-principle models (cf. latent force models described above) which allow for flexible gray box modeling using Gaussian processes.

One of the main obstacles in Gaussian process regression is still the computational scaling in the number of measurements, which is also inherited by all the new developments. Although several good approaches to tackle this problem have been proposed, the problem still is that they inherently replace the original model with an approximation. New better approaches to this problem are likely to appear in the near future.

Summary and Future Directions

In this article, we have briefly outlined the main directions in system identification using Gaussian processes. The methods can be divided into three

Cross-References

- [Nonlinear System Identification: An Overview of Common Approaches](#)
- [System Identification: An Overview](#)

Bibliography

- Ackermann ER, De Villiers JP, Cilliers P (2011) Nonlinear dynamic systems modeling using Gaussian processes: predicting ionospheric total electron content over South Africa. *J Geophys Res Space Phys* 116(10):13
- Álvarez MA, Luengo D, Lawrence ND (2013) Linear latent force models using Gaussian processes. *IEEE Trans Pattern Anal Mach Intell* 35(11):2693–2705
- Brooks S, Gelman A, Jones GL, Meng XL (2011) *Handbook of Markov chain Monte Carlo*. Chapman & Hall/CRC, Boca Raton
- Cressie NAC (1993) *Statistics for spatial data*. Wiley, New York
- Damianou AC, Lawrence ND (2013) Deep Gaussian processes. In: *International conference on artificial intelligence and statistics (AISTATS)*, pp 207–215
- Deisenroth MP, Turner RD, Huber MF, Hanebeck UD, Rasmussen CE (2011) Robust filtering and smoothing with Gaussian processes. *IEEE Trans Autom Control* 57(7):1865–1871
- Deisenroth MP, Fox D, Rasmussen CE (2015) Gaussian processes for data-efficient learning in robotics and control. *IEEE Trans Pattern Anal Mach Intell* 37(2):408–423
- Frigola R (2016) *Bayesian time series learning with Gaussian processes*. Ph.D thesis, University of Cambridge
- Frigola R, Lindsten F, Schön TB, Rasmussen CE (2013) Bayesian inference and learning in Gaussian process state-space models with particle MCMC. In: *Advances in neural information processing systems*. Curran Associates, Inc., Red Hook, pp 3156–3164
- Frigola R, Chen Y, Rasmussen CE (2014a) Variational Gaussian process state-space models. In: *Advances in neural information processing systems*. Curran, Red Hook, pp 3680–3688
- Frigola R, Lindsten F, Schön TB, Rasmussen CE (2014b) Identification of Gaussian process state-space models with particle stochastic approximation EM. *IFAC Proc Vol* 47(3):4097–4102. *Proceedings of the 19th IFAC world congress*
- Hartikainen J, Sarkka S (2010) Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In: *IEEE international workshop on machine learning for signal processing (MLSP)*, pp 379–384
- Ko J, Fox D (2009) GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. *Auton Robot* 27(1):75–90
- Kocijan J (2016) *Modelling and control of dynamic systems using Gaussian process models*. Springer, Cham
- Kocijan J, Petelin D (2011) Output-error model training for Gaussian process models. In: *International conference on adaptive and natural computing algorithms*. Springer, Berlin, pp 312–321
- Kocijan J, Girard A, Banko B, Murray-Smith R (2005) Dynamic systems identification with Gaussian processes. *Math Comput Model Dyn Syst* 11(4):411–424
- Lindgren F, Rue H, Lindström J (2011) An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *JRSS B* 73(4):423–498
- Matérn B (1960) *Spatial variation*. Technical report, Meddelanden från Statens Skogsforskningsinstitut, band 49 – Nr 5
- McHutchon AJ (2015) *Nonlinear modelling and control using Gaussian processes*. Ph.D thesis, University of Cambridge
- Quiñonero-Candela J, Rasmussen CE (2005) A unifying view of sparse approximate Gaussian process regression. *JMLR* 6:1939–1959
- Quiñonero-Candela J, Rasmussen CE, Figueiras-Vidal AR et al (2010) Sparse spectrum Gaussian process regression. *J Mach Learn Res* 11:1865–1881
- Rasmussen CE, Williams CK (2006) *Gaussian processes for machine learning*. MIT Press, Cambridge
- Särkkä S (2013) *Bayesian filtering and smoothing*. Cambridge University Press, Cambridge
- Särkkä S, Solin A, Hartikainen J (2013) Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing. *IEEE Sig Process Mag* 30(4):51–61
- Särkkä S, Álvarez MA, Lawrence ND (2019, to appear) Gaussian process latent force models for learning and stochastic control of physical systems. *IEEE Trans Autom Control* 64(7):2953–2960
- Solin A, Särkkä S (2018) Hilbert space methods for reduced-rank Gaussian process regression. *ArXiv*: 1401.5508
- Svensson A, Solin A, Särkkä S, Schön T (2016) Machine Learning Research. In: *Proceedings of the 19th International Conference on Artificial intelligence and statistics*, Vol 51, pp 213–221
- Titsias M (2009) Machine Learning Research. In: *Proceedings of the Twelfth International Conference on Artificial intelligence and statistics*, Vol 5, pp 567–574
- Turner R, Deisenroth M, Rasmussen C (2010) State-space inference and learning with Gaussian processes. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp 868–875

Validation and Verification Techniques and Tools

Christine M. Belcastro
NASA Langley Research Center, Hampton,
VA, USA

Abstract

Validation and verification (V&V) of advanced control systems is required for their use in fielded systems. A comprehensive V&V process involving analysis, simulation, and experimental testing should be used to assess closed-loop system performance and identify system limitations. This entry discusses current V&V methods and tools as well as future research directions for safety-critical control applications.

Keywords

Closed-loop system stability and performance · Software verification · Stability and performance robustness · Uncertainties and uncertainty models · Validation of safety-critical systems

Introduction

Control system validation and verification (V&V) is an assurance that the closed-loop system (i.e.,

the control system acting on the plant being controlled) remains stable and performs within acceptable performance metrics across the operational region of application. Basic definitions of validation and verification are given below (IEEE 2011).

Validation: The assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers.

Verification: The evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process.

Control system validation can therefore be thought of as a confirmation that the algorithms are performing their intended functions and an affirmation of their effectiveness in performing these functions. Validation is a rigorous evaluation process that should involve clearly identifying system limitations, including regions of operation within which stability or acceptable levels of performance are not guaranteed. Verification can be thought of as a confirmation that the system implementation in software and hardware is correctly executing the algorithms as designed (and validated). This includes a rigorous evaluation of system requirements and specifications and a clear determination of whether or not they have been met. V&V methods include analysis, simulation, and experimental testing, which are (ideally) applied in an integrated or iterative manner to corroborate results across methods of

evaluation. In the case of aircraft flight control and other safety-critical control applications, the V&V process must ultimately lead to system certification. The following subsections summarize control system V&V analytical, simulation, and experimental test methods in terms of the current (or recommended) state of practice. A summary and some future research directions for safety-critical control applications are also provided.

Control System Validation

Validation methods, involving analysis, simulation, and experimental testing, are utilized to ensure against errors and significant deficiencies in the underlying control system algorithms under realistic operational conditions for the intended application. System weaknesses and limitations are also identified through the validation process. Control system validation begins with an analysis of closed-loop system stability, performance, and robustness. Linear systems theory forms the basis for analytical stability and robustness methods and the associated software tools that are currently available for closed-loop system validation. In current practice, stability of nonlinear systems is determined by evaluating the stability of the linearized closed-loop system at numerous equilibrium points across the operating range (or envelope) of the system. Closed-loop stability is assessed by computing the eigenvalues of the linearized closed-loop system at a number of equilibrium points in the region of operation. It should be noted, however, that stability is not guaranteed between the operating points analyzed. Moreover, if the control system utilizes gain scheduling across the operating envelope, stability cannot be guaranteed for interpolated gains between the design points. Relative stability is determined by gain and phase margins for single-input, single-output (SISO) systems and by the multivariable stability margin (see, e.g., Lavretsky and Wise 2013) for multiple-input, multiple-output (MIMO, or multivariable) systems. Time-delay margins, defined as the minimum time delay required to destabilize the closed-loop system, can be determined

from phase margin and verified in nonlinear simulation. Stability robustness is assessed in terms of uncertainties, including parametric uncertainties and unmodeled dynamics in the mathematical model of the plant. Advanced robustness analysis methods based on the structured singular value (Zhou et al. 1996) require the uncertainties to be separated from the nominal plant into what is termed a linear fractional transformation (LFT). Advanced robustness methods can be used to assess stability and performance robustness, as well as worst-case combinations of uncertainties that result in destabilization, loss of performance, or the lowest robustness margins. LFT models can be formulated for the analysis of nonlinearities, expressed as multivariate polynomials, around a trim condition or over subregions within the operational envelope. Stability over a region or subregion of the operating envelope can be guaranteed using linear parameter-varying (LPV) methods (see Packard 1994; Wu et al. 1996; Apkarian and Gahinet 1995; Rugh and Shamma 2000). Nonlinear stability and control are addressed more fully in Slotine and Li (1991) and Khalil (2002), as well as in numerous other texts. Analytical methods and software tools are available in Matlab[®], using the Control System Toolbox[™] and Robust Control Toolbox[™]. LFR/LFT modeling methods and software tools are described in Magni (2004, 2006), Hecker et al. (2005), Varga et al. (1998), and Belcastro et al. (2005) and provided in the Robust Control Toolbox[®]. An LPV Toolbox[™] is also under development and will become available soon (see Balas et al. 2013b).

Performance is usually assessed using a high-fidelity simulation of the plant under expected operational conditions. The simulation should include nonlinear plant dynamics, noise, disturbances, and any other phenomena anticipated under operation. Nominal performance is assessed in terms of the control system design objectives, which typically include (at a minimum) closed-loop steady-state tracking and transient response characteristics. Transient response characteristics typically include delay time, rise time, peak time, maximum overshoot,

and settling time (see, e.g., Ogata 1970). Steady-state tracking error is also typically assessed. Performance robustness can be evaluated using Monte Carlo simulation techniques (see, e.g., Kroese et al. 2011), in which parameters and operational conditions are varied over numerous simulation runs in order to statistically evaluate an extensive set of uncertainties and operational variations. Stability robustness can also be assessed using Monte Carlo simulation techniques and by utilizing worst-case uncertainties and time-delay margins obtained during analysis. If the plant is a vehicle or robotic manipulator to be operated by a human, the handling qualities must also be evaluated to assess human-system interfaces and interactions. A real-time high-fidelity simulation with a human interface representative of the operational environment is required for this evaluation. For aircraft, piloted simulation evaluations are conducted using a cockpit mock-up, and handling qualities are assessed under various scenarios using the Cooper-Harper Scale (Cooper and Harper 1969). Susceptibility to operator-induced oscillations, for example, resulting from time delays in the controlled response, may typically be uncovered using operator-in-the-loop simulation evaluations.

Experimental testing should be conducted under realistic conditions that cover the entire (potential) operational space of the plant being controlled in order to assess realistic operational performance. For aircraft, this includes flight testing using full-scale and/or subscale test vehicles under nominal and off-nominal conditions. If the analytical and simulation evaluations provide a good match to the experimental evaluations, the test matrix can be comprised of key high-risk conditions to confirm desired behavior.

Validation methods should be applied in an iterative manner comparing results from the analysis, simulation, and experimental tests and going back to reevaluate in one domain based on results from another. For example, analysis results should provide good predictions of results seen in simulation and experimental testing. If a good match is not obtained, the analysis model may have to be improved or another analysis method

utilized to reduce conservatism in the result. Similarly, simulation results should provide a good prediction of experimental test results.

Control System Verification

System verification is ideally performed by or in collaboration with a computer science specialist to ensure against errors in the software/hardware implementation of the control algorithms. Control system verification begins with an analysis (Rushby 1995, 2009) of the software and hardware implementation requirements to ensure completeness and accuracy in the system specification. Several refinement steps are taken to transform the control system requirements to those implementable on the actual avionics hardware to be fielded. Verification, consisting of tests and analyses, is required to confirm requirements traceability and compliance from one refinement to another, to confirm accuracy of the algorithms, and to assure compliance and robustness of the final code with respect to the original control algorithms, to ensure that no errors are introduced from the refinement itself or related to the target computing platform. Formal analysis methods can be used to evaluate software logic and other software mechanisms for correctness under all operational conditions and to provide correctness proofs. Formal methods can also be utilized for model checking to verify system properties through an exhaustive search of all possible states that can be entered during execution (Berard et al. 1998). One software verification tool is called PVS (Owre et al. 1992), developed by SRI International, and is available on the Internet as an open-source software tool. Other methods and tools are also available from SRI International, as well as other sources.

Simulation techniques are used to evaluate software code, modules, subsystems, and the full system. Once the software has been verified, it is implemented on actual or representative hardware and evaluated using hardware-in-the-loop simulation and experimental testing. Experimental testing should include laboratory evaluations under all possible operational conditions and in

a relevant application environment. For aircraft, this would include flight testing of the control system on the actual avionics hardware to be fielded.

Summary and Future Directions

This entry has summarized current and recommended practices for control system V&V, including methods and tools for analysis, simulation, and experimental testing. The V&V process ensures against errors and deficiencies in the underlying system algorithms (validation) and in its software/hardware implementation (verification). Analysis, simulation, and experimental testing are performed iteratively to utilize and confirm results between evaluation techniques.

A comprehensive validation process is performed to assure control system effectiveness across the operational envelope of the plant and to identify system limitations and weaknesses. Current analysis methods for control system validation are typically based on linear systems theory and focus on nominal operations under model uncertainties and anticipated disturbances (e.g., noisy measurement signals). This analysis includes stability, performance (e.g., tracking accuracy), and robustness. Advanced robust control analysis methods have been applied to safety-critical applications, such as aircraft (see Fielding et al. 2002; Varga et al. 2012), to assess robust stability and performance. These two references provide a global optimization-based worst-case approach as a “necessary condition” technique for flight control system validation or as a “sufficient condition” technique for invalidation. High-fidelity nonlinear simulation evaluations are performed in batch and real time to assess robustness under system and operational uncertainties and to assess interface effectiveness for human-in-the-loop operations (if applicable). Experimental testing under realistic operationally relevant conditions is performed across key operating conditions to confirm analytical and simulation predictions.

System verification is performed to ensure correctness of the hardware/software implementation. Various analysis and testing methods, including advanced formal analysis methods, are used to assess completeness of the system requirements and specification and software elements (e.g., logic). Model checking techniques are used to verify system properties. Code is tested in simulation and on representative or actual hardware under realistic operationally relevant conditions.

Future research directions will enable the V&V of nonlinear and adaptive control systems (see, e.g., Hovakimyan and Cao 2010; Tallant et al. 2004) that improve performance under highly uncertain conditions, as well as the V&V of complex integrated safety-critical systems for operation under off-nominal and hazardous conditions (see Belcastro 2010, 2012). These systems will include diagnostic and prognostic algorithms for integrated vehicle health management, resilient control systems that enable the detection and mitigation of multiple hazards, supervisory systems that provide safety assurance (for safety-critical operations), and intelligent interface and decision-based systems that enable human-optional and fully autonomous operations. These systems will inherently involve stochastic decision-making and nonlinear and adaptive control algorithms. V&V of these future systems poses significant technical challenges and is the subject of current research. Some of these challenges include the following: (1) development and validation of multidisciplinary simulation models for characterizing hazardous condition effects; (2) validation of adaptive, diagnostic/prognostic, and reasoning algorithms under numerous off-nominal and hazardous conditions; (3) verification of software-intensive highly complex systems; and (4) determining a level of confidence in V&V results for hazardous application domains that cannot be fully replicated during the evaluations.

Research on modeling and simulation methods is being performed to characterize multidisciplinary effects of off-nominal and hazardous conditions, and validation of these models can be difficult. For aircraft applications, hazardous

conditions relate to aircraft loss of control (LOC) and precursor conditions. LOC is a complex and highly nonlinear phenomenon for which there is little available data. Hazardous conditions considered in this research include vehicle upset conditions (e.g., stall/departure), vehicle impairment conditions (e.g., failure, icing, and damage), and external disturbances (e.g., inclement weather and wake vortices). Multidisciplinary models under development include aerodynamic, propulsion, and airframe structure effects. For example, simulation models for characterizing aircraft flight dynamics and control effects under upset conditions are currently being developed (see, e.g., Foster et al. 2005; Groen et al. 2012), as well as propulsion effects resulting from the associated reduced flow conditions (Liu et al. 2013). Model validation is being performed using available flight and accident data, as well as experimental testing in the laboratory and through sub-scale and full-scale flight testing. The enhanced high-fidelity simulation models resulting from this research will be used in the development and validation of onboard control systems designed to detect and mitigate these hazards.

Current research efforts for validating the above future systems include nonlinear robustness analysis methods and software tools (see (Chakraborty et al. 2011a, b), Packard et al. 2010; Summers et al. 2013; Balas et al. 2013a), nonlinear analysis methods for controlled systems (Kwatny et al. 2013; Gill et al. 2012), uncertainty quantification and robustness analysis methods for mixed uncertainties and multiple objectives (Kenny et al. 2012), and the analysis of stochastic filters (see, e.g., Reif et al. 1999; Rhudy et al. 2013a, b, c). The term “mixed uncertainty” refers to aleatory and epistemic uncertainties. Aleatory uncertainties are typically stochastic (or statistical) and represent operational or environmental uncertainties (e.g., turbulence) that cannot be altered or controlled during experiments or fielded applications. Epistemic uncertainties are typically deterministic and arise from lack of knowledge about the plant resulting from modeling assumptions, neglected effects (e.g., unmodeled dynamics), and parametric uncertainties resulting from inaccurate measurements or

operational variability. These analysis methods and tools will be used iteratively with simulation evaluations and experimental testing methods, as described herein, to comprehensively assess nonlinear and adaptive control systems that enable resilience under multiple hazards.

Current research efforts on software verification focus on argument-based safety assurance for highly complex integrated systems of systems; assessment tools for evaluating the safety and coordination of authority and autonomy assignments; methods for ensuring safety-critical properties of distributed systems; and the development of tools and techniques for assessing software-intensive systems in meeting performance safety objectives. Research on software-intensive systems includes the development of methods and tools to detect, diagnose, and predict adverse events due to a software fault or failure once the software has been verified and is in operation. Some recent references on this work include Holloway (2012), Xu et al. (2013), Driscoll et al. (2012), Person et al. (2011), and Latorella and Feary (2011).

Research has been initiated on developing methodologies for determining (i.e., quantifying) the predictive capability of the validation process for systems designed to operate under conditions that cannot be fully replicated during evaluations. Predictive capability assessment is an evaluation of the validity and level of confidence that can be placed in the validation process and results under nominal and hazardous conditions (and their associated boundaries). The need for this evaluation arises from the inability to fully evaluate these technologies under actual hazards to be encountered by the fielded system. A detailed disclosure is required of model, simulation, and emulation validity for the off-nominal conditions being considered in the validation, interactions that have been neglected, assumptions that have been made, and uncertainties associated with the models and data. Cross-correlations should be utilized between analytical, simulation, and ground test and flight test results in order to corroborate the results and promote efficiency in covering the very large space of operational and off-nominal and

hazardous conditions being evaluated. The level of confidence in the validation process and results must be established for subsystem technologies as well as the fully integrated system. This includes an evaluation of error propagation effects across subsystems and an evaluation of integrated system effectiveness in mitigating hazardous conditions and preventing cascading errors, faults, and failures across subsystems. Metrics for performing this evaluation are also needed.

Cross-References

- [Computer-Aided Control Systems Design: Introduction and Historical Overview](#)
- [Interactive Environments and Software Tools for CACSD](#)
- [Robust Synthesis and Robustness Analysis Techniques and Tools](#)

Bibliography

- Apkarian P, Gahinet P (1995) A convex characterization of gain-scheduled Hinf controllers. *IEEE Trans Autom Control* AC-40(5):853–864
- Balas G, Chiang R, Packard A, Safonov M, Robust control toolbox™, Matlab® product family. The Mathworks Inc., Natick, MA, 1994–2014
- Balas G, Packard A, Seiler P, Topcu U (2013a) Robustness analysis of nonlinear systems. University of Minnesota. Website <http://www.aem.umn.edu/~AerospaceControl/>
- Balas GJ, Chaing R, Packard AK, Safonov M (2013b) Robust control toolbox. The Mathworks Inc., Natick, MA
- Belcastro CM, Khong TH, Shin J-Y, Balas GJ, Kwatny HG, Chang B-C (2005) Uncertainty modeling for robustness analysis of control upset prevention and recovery systems. In: AIAA guidance, navigation, and control conference and exhibit, AIAA-2005-6427, San Francisco
- Belcastro CM (2010) Validation and verification of future integrated safety-critical systems operating under off-nominal conditions. In: AIAA guidance, navigation and control conference, Toronto, Aug 2010
- Belcastro CM (2012) Validation of safety-critical systems for aircraft loss-of-control prevention and recovery. In: AIAA guidance, navigation, and control conference, Minneapolis, Aug 2012
- Berard B, Bidoit M, Finkel A, Laroussinie F, Petit A, Petrucci L, Schnoebelen P (1998) Systems and software verification: model-checking techniques and tools. Springer, Berlin
- Chakraborty A, Seiler P, Balas GJ (2011a) Susceptibility of F/A-18 flight controllers to the falling-leaf mode: linear analysis. *J Guid Control Dyn* 34(1):57–72
- Chakraborty A, Seiler P, Balas GJ (2011b) Susceptibility of F/A-18 flight controllers to the falling-leaf mode: nonlinear analysis. *J Guid Control Dyn* 34(1):73–85
- Chen C-T (1998) Linear system theory and design, 3rd edn. Oxford University Press, New York
- Control System Toolbox™, Matlab® product family. The Mathworks Inc., Natick, MA
- Cooper GE, Harper RP Jr (1969) The use of pilot rating in the evaluation of aircraft handling qualities. AGARD report 567, Apr 1969
- Doyle JC, Francis BA, Tannenbaum AR (1992) Feedback control theory, Macmillan, New York
- Driscoll K, Madl G, Hall B (2012) Modeling and analysis of mixed synchronous/asynchronous systems. NASA/CR-2012-217756, Sept 2012
- Fielding C, Varga A, Bennis S, Selier M (eds) (2002) Advanced techniques for clearance of flight control laws. Springer, Berlin
- Foster JV, Cunningham K, Fremaux CM, Shah GH, Stewart EC, Rivers RA, Wilborn JE, Gato W (2005) Dynamics modeling and simulation of large transport airplanes in upset conditions. In: AIAA guidance, navigation, and control conference, San Francisco
- Gill SJ, Lowenberg MH, Krauskopf B, Puyou G, Coetzee E (2012) Bifurcation analysis of the NASA GTM with a view to upset recovery. In: AIAA guidance, navigation, and control conference, Minneapolis, Aug 2012
- Groen E, Ledegang W, Field J, Smaili H, Roza M, Fucke L, Nooij S, Goman M, Mayrhofer M, Zaichik L, Grigoryev M, Biryukov V (2012) SUPRA – enhanced upset recovery simulation. In: AIAA guidance, navigation, and control conference, Minneapolis, Aug 2012
- Hartmann AK (2009) Practical guide to computer simulations. World Scientific, Hackensack, New Jersey
- Hecker S, Varga A, Magni J (2005) Enhanced LFR toolbox for Matlab. *Aerosp Sci Technol* 9(2):173–180
- Holloway CM (2012) Towards understanding the DO-178C/ED-12C assurance case. In: Proceedings of the IET 7th international conference on system safety, Edinburgh, Oct 2012
- Hovakimyan N, Cao C (2010) L1 adaptive control theory. Society for Industrial and Applied Mathematics, Philadelphia
- IEEE (2011) IEEE guide–adoption of the Project Management Institute (PMI®) standard a guide to the Project Management Body of Knowledge (PMBOK® Guide), 4th edn. IEEE, p 452. doi:10.1109/IEEESTD.2011.6086685. Retrieved 7 Dec 2012
- Kenny SP, Crespo LG, Giesy DP (2012) UQ tools: the uncertainty quantification toolbox – introduction and tutorial. NASA TM-2012-217561, Apr 2012

- Khalil HK (2002) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River, New Jersey
- Kwatny HG, Dongmo J-ET, Chang B-C, Bajpai G, Yasar M, Belcastro C (2013) Nonlinear analysis of aircraft loss of control. *J Guid Control Dyn* 36(1):149–162
- Kroese DP, Taimre T, Botev ZI (2011) Handbook of Monte Carlo methods. Wiley series in probability and statistics. Wiley, New York
- Latorella KA, Feary M (2011) NASA aviation safety programs: human factors focused work. In: 16th International symposium on aviation psychology, Dayton, 2–5 May 2011
- Lavretsky E, Wise KA (2013) Robust and adaptive control. Springer, London
- Liu Y, Claus RW, Litt JS, Guo T-H (2013) Simulating Effects of High Angle of Attack on Turbofan Engine Performance. In: 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Grapevine, Texas, 7–10 January 2013
- Magni J-F (2004) Linear fractional representation toolbox – modeling, order reduction, and gain scheduling. ONERA technical report TR 6/08162 DSCD, Systems Control and Flight Dynamics Department, ONERA, July 2004
- Magni J-F (2006) User manual of the linear fractional representation toolbox (version 2.0). Technical report 5/10403.01F, ONERA/DCSD. <http://www.onera.fr/staff-en/jean-marc-biannic/docs/lfrtv20s.zip>
- Matlab®, The Mathworks Inc., Natick, MA
- Ogata K (1970) Modern control engineering. Prentice-Hall, Englewood Cliffs
- Owre S, Shankar N, Rushby J (1992) PVS: a prototype verification system. In: CADE 11, Saratoga Springs, June 1992
- Packard AK (1994) Gain-scheduling via linear fractional transformations. *Syst Control Lett* 22:79–92
- Packard A, Topcu U, Seiler P, Balas G (2010) Help on SOS. *IEEE Control Syst Mag* 30(4):18–23
- Person SJ, Yang G, Rungta N, Khurshid S (2011) Directed incremental symbolic execution. In: 32nd ACM SIGPLAN conference on programming design and implementation, San Jose, June 4–8 2011
- Reif K, Gunther S, Yaz E, Unbehauen R (1999) Stochastic stability of the discrete-time extended Kalman filter. *IEEE Trans Autom Control* 44(4):714, 728
- Rhudy M, Gu Y, Napolitano MR (2013a) An analytical approach for comparing linearization methods in EKF and UKF. *Int Journal of Adv Robot Syst*, 2013, Vol. 10, 208. doi:10.5772/56370
- Rhudy M, Gu Y, Napolitano MR (2013b) Does the unscented Kalman filter converge faster than the extended Kalman filter? A counter example. AIAA guidance navigation and control conference, Boston, Aug 2013
- Rhudy M, Gu Y, Gross J, Gururajan S, Napolitano MR (2013c) Sensitivity analysis of extended and unscented Kalman filters for attitude estimation. *AIAA J Aerosp Inf Syst* 10(3):131–143
- Rugh J, Shamma J (2000) A survey of research on gain scheduling. *Automatica* 36:1401–1425
- Rushby J (1995) Formal methods and their role in digital systems validation for airborne systems. NASA Contractor report 4673, Aug 1995
- Rushby J (2009) Software verification and system assurance. In: 7th IEEE international conference on software engineering and formal methods (SEFM), Hanoi, Nov 2009
- Simulink®, The Mathworks Inc., Natick, MA
- Slotine J-JE, Li W (1991) Applied nonlinear control. Pearson education. Prentice Hall, Upper Saddle River, New Jersey
- Summers E, Chakraborty A, Tan W, Topcu U, Seiler P, Balas GJ, Packard AK (2013) Quantitative local L_2 -gain and reachability analysis for nonlinear systems. *Int J Robust Nonlinear Control*, 23:1115–1135
- Tallant GS, Hull RA, Bose P, Johnson T, Buffington JM, Krogh B, Crum VW, Prasanth R (2004) Validation & verification of intelligent and adaptive control systems. IEEEAC paper #1487, Dec 2004
- Varga A, Looye G, Moormann D, Grubel G (1998) Automated generation of LFT-based parametric uncertainty descriptions from generic aircraft models. *Math Comput Model Dyn Syst* 4:249–274
- Varga A, Hansson A, Puyou G (2012) Optimization based clearance of flight control laws. Springer, Berlin
- Wu F, Packard AK, Becker G (1996) Induced L_2 -norm control for LPV systems with bounded parameter variation rates. *Int J Control* 6(9/10):983–998
- Xu X, Ulrey M, Brown JA, Mast J, Lapis MB (2013) Safety sufficiency for NextGen: assessment of selected existing safety methods, tools, processes, and regulations. NASA/CR-2013-217801, Feb 2013
- Zhou K, Doyle JC (1997) Essentials of robust control. Prentice Hall, Englewood Cliffs, New Jersey
- Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, Englewood Cliffs, New Jersey

Vehicle Dynamics Control

H. Eric Tseng

Ford Motor Company, Dearborn, MI, USA

Abstract

The current prevailing control technology enables Vehicle Dynamics Control through powertrain torque manipulation and individual wheel braking. Longitudinal control can maintain vehicle acceleration/braking capability within the physical limits that the road condition can support, while vehicle lateral control can preserve vehicle steering/handling

capability up to the maximum capacity offered by the road/tire interaction. Since most of these controllers are driver-assist systems, their objective is to retain the vehicle dynamic state in operating regions familiar to drivers. In general, this implies that the controller will keep the tire in its linear region and avoid excessive slipping, skidding, or sliding.

Keywords

Traction control · Traction assist · Electronic stability control · Active yaw control · Vehicle stability assist · Lateral dynamics · Evasive maneuvers

Introduction

Vehicle Dynamics Control (VDC) generally refers to the active modification of longitudinal and lateral tire forces, and the corresponding dynamics of ground vehicles using sensors and actuators. While it may also include vehicle active or semi-active suspension control (Hrovat 1997), Vehicle Dynamics Control in this article will focus on Traction Control (TC) – vehicle longitudinal control and Electronic Stability Control (ESC) – combined vehicle longitudinal and lateral control.

Simply speaking, tire force is generated when there exists a velocity difference between tire tread and the ground, also known as tire slip. As illustrated in Fig. 1, the longitudinal tire force first grows proportionally with the tire slip, in a so-called “linear region,” and then saturates as tire slip passes beyond a certain threshold. The figure also shows the coupling effect between longitudinal and lateral tire forces. That is, the available lateral force (as a function of tire slip angle) decreases when the longitudinal tire slip increases, and the available longitudinal force (as a function of tire slip) decreases as the lateral tire slip angle increases. This coupling effect is essential for understanding vehicle dynamics and leads to numerous control applications.

Traction Control

Since vehicle motion relies on the tire/ground interaction, it is important for the purpose of vehicle controllability to maintain tire/road interaction in a linear and predictable way. Antilock Braking Systems (ABS), and Traction Control (TC) in particular, monitor and control the tire slip so that the longitudinal tire force can best support and balance the corresponding brake torque (during ABS intervention) or driveline torque (during TC intervention) delivered to the wheels. Without the wheel/tire slip control, tire force may saturate, resulting in both the reduction of longitudinal and lateral tire force capacity, with the corresponding reduction in decelerating/accelerating capability, or loss of road grip/lateral tire force capacity.

Control Design

The objective of a TC system is to ensure longitudinal tire force capacity while maintaining a good margin on available lateral force road grip (see Fig. 1). Based on the tire force-slip characteristics, this can be achieved by regulating the longitudinal tire slip, roughly defined as the relative velocity between the contact patch of the tire and the road surface. This can be expressed as the difference between the vehicle traveling speed and tire rotational speed, as defined in Eq. 1, according to the Society of Automotive Engineers (SAE),

$$s = \frac{V - R\omega}{V}, \quad (1)$$

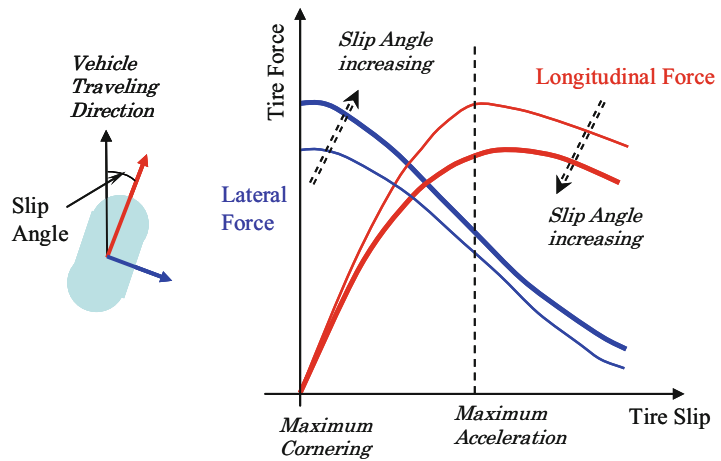
where V is the vehicle speed, ω is the angular speed of the tire, and R is the effective tire rolling radius. The effective rolling radius is defined so that vehicle speed equals the product of R and ω (i.e., $V = R\omega$) when there is no torque applied to the wheel and the tire is free rolling.

At low slip, the longitudinal tire force grows as the slip increases (Carlson and Gerdes 2003), while at high slip, it passes its peak and begins to decrease (Deur et al. 2004). High slips occur when wheels are locked during braking or are

Vehicle Dynamics

Control, Fig. 1

Longitudinal and lateral tire force as a function of tire slip and slip angle



over-spinning during acceleration. A Traction Control system uses feedback control to regulate wheel/tire slip.

Sensors and Actuators

To regulate driven wheel slips effectively with closed loop control, wheel speed sensors at the non-driven wheels are utilized for vehicle speed estimation (V in Eq. 1). In the case of all-wheel drive or four-wheel drive systems, a longitudinal accelerometer is typically added for the speed estimate. As the amount of desired wheel slip may vary depending on maneuvers, accelerator pedal, steering angle, and yaw rate signals may be used as well. Some systems deploy steering wheel angle and yaw rate sensors for direct signal assessment and signal sharing competency, while others estimate these signals based on the speed difference between left and right wheels, for subsystem modularity across various vehicle configurations and platforms, as well as calibration independency.

Powertrain and brake torque modulation are typically used for actuation to regulate driven wheel slips.

Control System Behavior

Wheel/tire slip targets are typically adjusted based on vehicle driveline configurations as well as vehicle maneuvers. When a vehicle is

cornering, a low slip target is generated to assure sufficient margin in lateral tire force capacity. Similarly, rear wheel drive vehicles may warrant a lower slip target than front or all-wheel drive vehicles. When a driver presses hard on the accelerator pedal, the slip target can be raised to accommodate higher acceleration. In addition, the target can be adjusted based on vehicle speed and estimated road available friction, all in an attempt to optimize the longitudinal traction force while keeping sufficient margin on lateral grip (Fodor et al. 1998; Hrovat et al. 2000).

Uniform Friction Surface: (Uniform μ)

Unless equipped with advanced driveline mechanism such as Active Limited Slip Differential or Torque Vectoring Differential (Deur et al. 2010), a vehicle is typically equipped with open differential, thus transmitting the powertrain torque evenly to both left and right driven wheels. Since there is no difference between the left and right wheel torque that can be supported by the uniform driving surface, wheel slip regulation can be quite effectively achieved by modulating only the powertrain torque. One successful example of this is Ford's engine-only Traction Control system, introduced in 2006 on its Fusion and F150 models, which was well received by media experts and customers (Healey 2005).

Nonuniform Friction Surface: (Split μ)

For driving surfaces offering different tire/road characteristics, different wheel torque can be

supported on different sides. In this case, a vehicle equipped with an open differential would transmit only the minimum torque (set by the low friction side) to the road. Any additional driveline torque that cannot be supported by the road surfaces results in spinning the wheel on the low friction side. Since open differential transmits equal amount of torque left and right, the additional tire force available on the higher friction side would not be fully utilized with powertrain only actuation. In this case, by applying additional brake torque at the low road friction side, the driveline torque can be balanced at the higher level offered by the high friction side. Care must be taken to avoid aggressive brake application which can cause driveline and/or half shaft oscillations (Fodor et al. 1998; Hrovat et al. 2000).

Control Challenges

Given that road/tire interaction varies with multiple environmental factors, the tire force/slip relationship depicted in Fig. 1 is only a qualitative characterization, and the actual optimal slip for a desired traction force is difficult to accurately establish. While the peak traction force and the corresponding road friction potential (i.e., $\mu_{\max}$) can be detected once the wheel starts to spin, it is difficult to do so prior to a wheel spin event (Gustafsson 1996).

Since the powertrain actuation is less perceptible yet occasionally sluggish and the brake application can be fast but intrusive at times, it can be a control challenge to optimize the actuation combination and bandwidth. Opportunities of better performance arise with the introduction of electric vehicle where the driven torque has higher bandwidth than the conventional powertrain (Hori et al. 1998). The higher bandwidth facilitates the rapid reduction of tire slip (especially for the “first spin” when a vehicle suddenly encounters low-friction surface) and the search for optimal slip ratio.

If the knowledge of friction potential and optimal slip can be learned online, detailed powertrain/driveline actuation delay and dynamics can be modeled, and optimal actuation

combination and bandwidth can be incorporated; it is conceivable that further improvement in wheel slip and Traction Control can be achieved, using advanced control approaches such as Model Predictive Control (MPC) (Borrelli et al. 2006), for example.

Electronic Stability Control

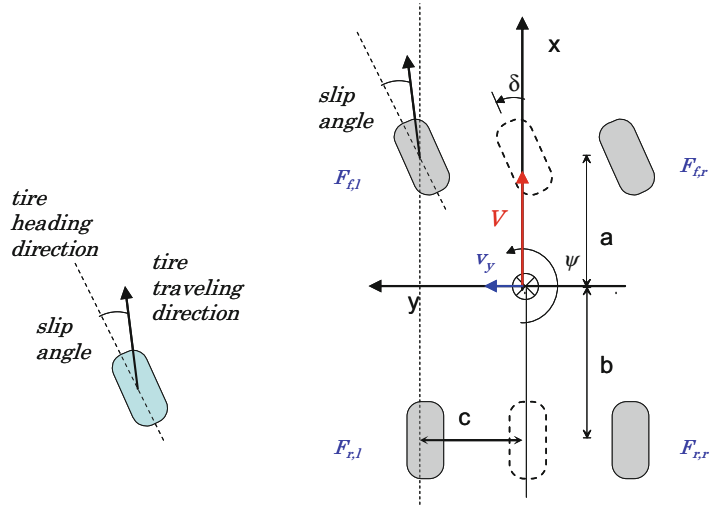
According to the Society of Automotive Engineers (SAE), an Electronic Stability Control (ESC) system is a computer-controlled system that augments vehicle directional stability by applying and adjusting individual wheel braking. It is operational over the full-speed range of the vehicle and is capable of monitoring both driver steering input and vehicle yaw rate to limit vehicle understeering and oversteering, as appropriate.

The wide proliferation of ESC in the recent years (Van Zanten 2000) across the vehicle fleet has allowed various evaluation studies of its effectiveness in real-world environments. Among them, the United States National Highway Traffic Safety Administration (NHTSA) study (Dang 2004) concluded that ESC reduces fatal single-vehicle crashes by 35%, while single-vehicle crashes involving Sport Utility Vehicles (SUVs) are reduced by 67%. Similar conclusions were reached in other subsequent studies, including the statement that “Electronic stability control could prevent nearly one-third of all fatal crashes...” by the Insurance Institute for Highway Safety organization (IIHS 2006). Many of these effectiveness studies are summarized in a literature review by Ferguson (2007).

Control Design

The objective of an ESC system is to provide vehicle controllability and predictability to assist the driver. This can be achieved by preventing excessive deviations between the intended and actual lateral response of the vehicle, especially during critical maneuvers such as a sudden encounter with a slippery/icy road.

**Vehicle Dynamics
Control, Fig. 2** Vehicle
cornering model



During driving, a driver relies on a mental model of the vehicle's response to his/her steering input developed from previous driving experience. A vehicle model, as described in Eq. 2 and Fig. 2, is often used to describe nominal lateral vehicle behaviors:

$$\begin{aligned}
 m\dot{v}_y &= -mV\dot{\psi} + F_{yf}(v_y, V, \delta, F_{xf}) \\
 &\quad + F_{yr}(v_y, V, \delta, F_{xr}) \\
 I\ddot{\psi} &= 2aF_{yf} - 2bF_{yr} \\
 &\quad + c(-F_{xf,l} + F_{xf,r} - F_{xr,l} + F_{xr,r})
 \end{aligned} \quad (2)$$

where m is the vehicle mass; V and V_y are the vehicle longitudinal and lateral velocities, correspondingly; $\dot{\psi}$ is vehicle yaw rate; and δ is the steering angle. The parameters a and b are the distances between the vehicle center of gravity to the front and rear axles, and c is the half-track width. F_x and F_y denote the longitudinal and lateral/cornering tire force, with subscript indicating longitudinal (x) or lateral (y) direction, as well as the specific corner of the vehicle (front, rear, left, and right).

Note that the corresponding longitudinal dynamics can be described as

$$m\dot{V} = mv_y\dot{\psi} + F_{xf,l} + F_{xr,l} + F_{xf,r} + F_{xr,r} \quad (3)$$

As the available road friction is not always known, a nominal vehicle lateral response derived from a hi- μ surface may not be feasible and may not best represent a driver's intent. To modify the feedback to best adapt to the road condition, ESC would do one or more of the following: adjust the driver intended yaw rate according to detected lateral acceleration capability (Tseng et al. 1999), balance between yaw rate and lateral acceleration error when both are compared to a nominal vehicle model (Manning and Crolla 2007), or balance between yaw rate error and detected excessive side slip angle (Di Cairano 2013).

Sensors and Actuators

In order to effectively provide vehicle controllability through an embedded controller, ESC systems are equipped with a steering angle sensor, wheel speed sensors, a yaw rate sensor, and a lateral accelerometer. Additional sensors such as a longitudinal accelerometer and a roll rate sensor may be installed to better observe the vehicle dynamic states and provide improved fidelity for estimated vehicle behaviors. The actuators of an ESC system are the individual wheel brakes and powertrain torque.

Control System Behavior

A vehicle can exhibit understeering and/or oversteering behaviors during aggressive lateral maneuvers. Figure 3 illustrates how a vehicle equipped with ESC may provide better controllability.

Understeering – When a vehicle does not turn in as much as desired by the driver (see upper Fig. 3). In this case, the vehicle yaw rate, an ESC measured/monitored signal would be less than the driver desired value. For example, a vehicle on ice may experience extreme understeering that keeps the vehicle moving straight even when the steering wheel is turned. In this case, ESC applies corrective yaw moment to increase the yaw rate through individual wheel braking. Most of the longitudinal braking force is applied on rear axle inside wheel in the attempt to increase the lateral force capacity on the front axle while decreasing the lateral force capacity on the rear axle. As such, the vehicle experiences not only the yaw moment correction but also the reduction of understeering tendency with ESC brake application.

Oversteering – When a vehicle turns too much, i.e., yaws with a smaller turning radius than the one needed to negotiate the road (see lower Fig. 3). In this case, the vehicle yaw rate would be larger than the driver desires. The vehicle tends to build up a large side slip angle, resulting in a spinout due to the saturation of rear tire force. In this case, ESC applies corrective yaw moment to decrease the size of yaw rate through individual wheel braking. The longitudinal braking force is applied mostly on the front axle outside wheel to preserve the lateral force capacity on the rear axle and decrease the lateral force capacity on the front axle. As such, the vehicle experiences not only the yaw moment correction but also the reduction of oversteering tendency with ESC brake application.

Evasive Maneuver – During an evasive maneuver, such as an aggressive double lane change, the vehicle may first turn in one direction, followed by an oversteer in the other direction. Due to delay and lag of actuator response in practice, feedforward control is typically used to ensure that brake application would generate corrective yaw moment in the

Vehicle Dynamics Control, Fig. 3 Vehicle going through a hairpin turn



appropriate direction. Further improvement could be possible by using road/traffic preview along with (semi)autonomous intervention based on advanced optimal control such as Model Predictive Control (Falcone et al. 2008), for example.

Skidding and Oversteering – When both front and rear tires experience large tire slip angle, both tire forces are saturated. In this case, the vehicle is operating in a region where the rear tire slip angle can grow rapidly. Unless the front steering is delicately and quickly balanced, the excessive rear tire slip angle could cause the vehicle to spinout. In this case, in addition to applying corrective yaw moment similar to the above oversteering case, ESC may command light braking on all four wheels in an attempt to further slow down the vehicle.

Rollover Mitigation – An ESC system may be extended to further provide a more controllable vehicle behavior and mitigate rollover risks in evasive maneuvers that demand a large and sudden lateral force. For example, Roll Stability ControlTM system introduced at Ford in 2003 monitors vehicle roll behavior in addition to vehicle yaw behavior to assist the driver (Lu et al. 2007).

Control Challenges

In order to best provide the assistance to drivers' desire, it is important to assess the vehicle dynamic state and driver intention with high fidelity. This can be challenging in the presence of various factors that directly influence the vehicle behavior or sensor readings but are not or cannot be directly measured. For example, the road bank angle information is typically unavailable, but it has a direct influence on the lateral accelerometer measurement and could be misinterpreted as a discrepancy between vehicle yaw rate and lateral force (Tseng 2001; Tseng et al. 2007). In addition, despite its criticality in Vehicle Dynamics Control, the available road surface friction capacity and the vehicle side slip angle typically cannot be measured (Tseng 2002; Ryu et al. 2002; Ahn et al. 2013). As for the driver intent, the controller maps the steering

wheel angle to a driver desired yaw rate with a dynamic vehicle model and utilizes the lateral acceleration measured along with the yaw rate and vehicle speed to prevent overestimation of the driver desired yaw rate. While this driver intent interpretation is sufficient to stabilize the vehicle, it cannot always keep the vehicle in a driver intended lane since it does not have such knowledge with the steering wheel being the only human machine interface for lateral maneuvers. In addition, the controller should detect when a sensor is misbehaving and giving out false or biased readings (Xu and Tseng 2007). While advanced observers have been developed to address these challenges, it is foreseeable that optimization in these areas could further improve the observer fidelity and overall ESC performance.

Opportunity: Computer Vision as Sensors

With the rear-view camera becoming a standard equipment on all vehicles and the wide spread introduction of front-view camera on vehicles equipped with advanced driver-assist systems such as lane keeping and lane following assist, there comes the opportunity of added sensing capability to compliment the conventional sensor sets of VDC systems, including both aforementioned Traction Control and Electronic Stability Control systems. Since the detection of vehicle longitudinal and lateral velocity is now possible with the advancement of computer vision, these important vehicle state information can aid the on-board estimation of vehicle dynamic states (Song et al. 2008; Seegmiller and Wettergreen 2011), especially when all four wheels are slipping and none of the wheel speeds reflect vehicle's longitudinal traveling speed or when the vehicle is slowly drifting and the lateral velocity derivative (i.e., the difference between lateral acceleration and the product of yaw rate and vehicle speed) is small, resulting in difficulties for robust estimation of lateral velocity with conventional sensors. Therefore, maturing computer vision technologies for

the detection of 2D vehicle speed is a future opportunity for further advancement of VDC.

Opportunity: VDC for Automated Driving

With recent advancements in sensing and perception technologies, it is believed that automated driving will be an important mode for future transportation. As the operational design domain of automated driving grows from low longitudinal/lateral acceleration (low “G”-force) maneuvers to high longitudinal/lateral acceleration (high “G”-force) maneuvers, from low-speed to high-speed maneuvers, or from driving on dry asphalt to snow and ice, Vehicle Dynamics Control (VDC) will play an important role in assisting the autonomous vehicle to follow its planned path closely and safely, similar to the assistance VDC currently provided to human drivers (Fig. 4). There are more performance opportunities in assisting autonomous vehicles (AVs) than in assisting human drivers since the planned path information of AVs is accessible to and can be previewed by VDC, and VDC can feedback the sensed vehicle handling limit information to the path planning module.

VDC can provide assistance to human drivers by coordinating active steering and differential braking to stabilize the vehicle based on the driver’s steering input and its corresponding desired vehicle yaw rate (Di Cairano 2013). In comparison, VDC can provide assistance to autonomous vehicles by optimizing steering (Falcone et al. 2007) or both steering and braking

(Falcone et al. 2008) in the prediction horizon to best avoid obstacles while maintaining vehicle stability based on not only the current steering command but also the preplanned trajectory.

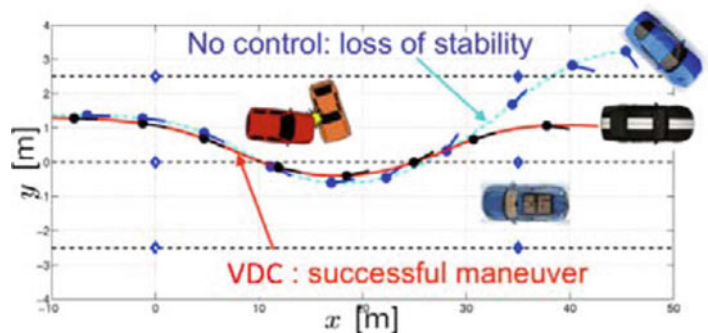
In the pursuit of VDC for autonomous vehicles, Turri et al. (2013) utilized linear Model Predictive Control (MPC) for both longitudinal and lateral dynamics to find the optimal steering, throttle, and braking on low curvature roads. Gao et al. (2010) designed a two-level MPC to communicate between VDC and Path Planner. Lee and Tseng (2018) utilized shadow trolley concept to integrate longitudinal speed and lateral path trajectory planning through scenario based MPC that works well for bending roads and switchbacks. Brown et al. (2017) further developed an integrated MPC that ensures both the planned and actual trajectories remain within two safe envelopes, one corresponding to conditions for stability and the other to obstacle avoidance. The safe envelope concepts are adopted and utilized in ToyotaTM Guardian Autonomous Driving Technology (Eustice 2019).

Summary and Future Directions

The advancement in sensing and connectivity (e.g., vehicle to vehicle/infrastructure) technologies that enable better perception of the surrounding driving environment, coupled with the proliferation of the electric motor that offers more direct and possibly individual wheel control, ultimately creates future opportunities for vehicle dynamics control. It is expected that VDC will be leveraged to further increase the performance

Vehicle Dynamics

Control, Fig. 4 VDC assists human driver in performing an evasive maneuver illustrated in *The Impact of Control Technology* (Samad and Annaswamy 2014)



envelope of vehicle dynamics as well as the safety of autonomous vehicles.

Cross-References

- [Lane Keeping Systems](#)
- [Motorcycle Dynamics and Control](#)

Bibliography

- Ahn C, Peng H, Tseng HE Robust estimation of road frictional coefficient. *Control Syst Technol IEEE Trans* 21(1):1
- Borrelli F, Bemporad A, Fodor M, Hrovat D (2006) An MPC/hybrid system approach to traction control. *Control Syst Technol IEEE Trans* 14(3):541–552
- Brown M, Funke J, Erlien S, Gerdes JC (2017) Safe driving envelopes for path tracking in autonomous vehicles. *Control Eng Pract* 61:307–316
- Carlson CR, Gerdes JC (2003) Nonlinear estimation of longitudinal tire slip under several driving conditions. In: Paper Presented in American Control Conference, June 2003
- Dang JN (2004) Preliminary results analyzing the effectiveness of electronic stability control (ESC) systems, Report no. DOT HS-809–790. National Highway Traffic Safety Administration, Washington
- Deur J, Asgari J, Hrovat D (2004) A 3D brush-type dynamic tire friction model, vehicle system dynamics. *Int J Veh Mech Mobil* 42(3):133–173
- Deur J, Ivanović V, Hancock M, Assadian F (2010) Modeling and analysis of active differential dynamics. *ASME J Dyn Syst Meas Control* 132(6):1–13
- Di Cairano S, Tseng HE, Bernardini D, Bemporad A (2013) Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain. *Control Syst Technol IEEE Trans* 21(4):1236, 1248
- Eustice R (2019) AI guarding the human: Toyota's guardian approach to automated driving. University of Michigan Control Seminar sponsored by Electrical and Computer Engineering at U-M, Bosch, Ford, and Toyota, hosted on 15 Mar 2019. <https://www.youtube.com/watch?v=QnUHg3ZpVWU>
- Falcone P, Borrelli F, Asgari J, Tseng HE, Hrovat D (2007) Predictive active steering control for autonomous vehicle systems. *IEEE Trans Control Syst Technol* 15(3):566–580
- Falcone P, Tseng HE, Borrelli F, Asgari J, Hrovat D (2008) MPC-based yaw and lateral stabilization via active front steering and braking. *Veh Syst Dyn* 46(S1): 611–628
- Ferguson SA (2007) The effectiveness of electronic stability control in reducing real-world crashes: a literature review. *Traffic Inj Prev* 8(4):329–338
- Fodor M, Yester J, Hrovat D (1998) Active control of vehicle dynamics. In: Paper presented in 17th AIAA/IEEE/SAE digital avionics systems conference, Seattle
- Gao Y, Lin T, Borrelli F, Tseng E, Hrovat D (2010) Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads. In: ASME 2010 Dynamic Systems and Control Conference. American Society of Mechanical Engineers, pp 265–272
- Gustafsson F (1996) Estimation and change detection of tire-road friction using the wheel slip. In: Computer-aided control system design, Proceedings of the 1996 IEEE International Symposium, pp 99–104
- Healey JR (2005) Ford's 2006 fusion review. Traction control on the V-6 test car was just right... http://usatoday30.usatoday.com/money/autos/reviews/healey/2005-10-27-fusion_x.htm posted on 27 Oct 2005. Accessed on 30 Aug 2013
- Hori Y, Toyoda Y, Tsuruoka Y (1998) Traction control of electric vehicle: basic experimental results using the test EV "UOT electric march". *IEEE Trans Indus Appl* 34(5):1131–1138
- Hrovat D (1997) Survey of advanced suspension developments and related optimal control applications. *Automatica* 33(10):1781–1817
- Hrovat D, Asgari J, Fodor M (2000) Automotive mechatronic systems. In: Leondes CT (ed) *Mechatronic systems techniques and applications: volume 2 – transportation and vehicular systems*. Gordon and Breach Science, pp 1–98
- Insurance Institute for Highway Safety (IIHS) (2006) Electronic stability control could prevent nearly one-third of all fatal crashes and reduce rollover risk by as much as 80%; effect is found on single- and multiple-vehicle crashes, News Release 13 June 2006. <http://www.iihs.org/news/rss/pr061306.html>. Accessed 30 Aug 2013
- Lee S, Tseng HE (2018) Trajectory planning with shadow trolleys for an autonomous vehicle on bending roads and switchbacks. In 2018 IEEE Intelligent Vehicles Symposium (IV). pp 484–489
- Lu J, Messih D, Salib A, Harmison D (2007) An enhancement to an electronic stability control system to include a rollover control function. *SAE Trans* 116:303–313
- Manning WJ, Crolla DA (2007) A review of yaw rate and sideslip controllers for passenger vehicles. *Trans Inst Meas Control* 29(2):117–135
- Ryu J, Rossetter EJ, Gerdes JC (2002) Vehicle sideslip and roll parameter estimation using GPS. In: Proceedings of AVEC 2002 6th International Symposium of Advanced Vehicle Control
- Samad T, Annaswamy AM (2014) The impact of control technology. *IEEE Control Syst Soc* 1:246. Hrovat D, Tseng H, Di Cairano S. Addressing automotive industry needs with model predictive control
- Seegmiller N, Wettergreen D (2011) Optical flow odometry with robustness to self-shadowing. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp 613–618

- Song X, Song Z, Seneviratne LD, Althoefer K (2008) Optical flow-based slip and velocity estimation technique for unmanned skid-steered vehicles. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp 101–106
- Tseng HE (2001) Dynamic estimation of road bank angle. *Veh Syst Dyn* 36(4–5):307–328
- Tseng HE (2002) A sliding mode lateral velocity observer. In: Proceedings of AVEC 2002 6th International Symposium on Advanced Vehicle Control. Hiroshima, pp 387–392
- Tseng HE, Ashrafi B, Madau D, Brown AT, Recker D (1999) The development of vehicle stability control at Ford. *Mech IEEE/ASME Trans* 4(3):223–234
- Tseng HE, Xu L, Hrovat D (2007) Estimation of land vehicle roll and pitch angles. *Veh Syst Dyn* 45(5):433–443
- Turri V, Carvalho A, Tseng HE, Johansson KH, Borrelli F (2013) Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads. In: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). IEEE, pp 378–383
- Van Zanten AT (2000) Bosch ESP systems: 5 years of experience. *SAE Trans* 109(7):428–436
- Xu L, Tseng HE (2007) Robust model-based fault detection for a roll stability control system. *Control Syst Technol IEEE Trans* 15(3):519–528

Vehicular Chains

Mihailo R. Jovanović
Department of Electrical and Computer
Engineering, University of Minnesota,
Minneapolis, MN, USA

Abstract

Even since the pioneering work of Levine and Athans and Melzer and Kuo, control of vehicular formations has been a topic of active research. In spite of its apparent simplicity, this problem poses significant engineering challenges, and it has often inspired theoretical developments. In this article, we view vehicular formations as a particular instance of dynamical systems over networks and summarize fundamental performance limitations arising from the use of local feedback in formations subject to stochastic disturbances. In topology of regular lattices, it is impossible to have

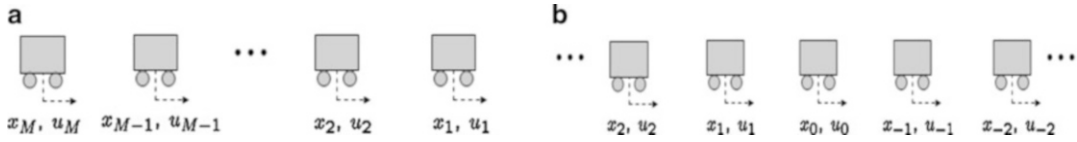
coherent large formations, which behave like rigid lattices, in one and two spatial dimensions; yet this is achievable in 3D. This is a consequence of the fact that, in 1D and 2D, local feedback laws with relative position measurements are ineffective in guarding against disturbances with slow temporal variations and large spatial wavelength.

Keywords

Fundamental performance limitations · Localized control · Optimal control · Relative information exchange · Spatially invariant systems · Toeplitz and circulant matrices · Vehicular formations

Introduction

Control of vehicular strings has been an active area of research for almost five decades (Levine and Athans 1966; Melzer and Kuo 1971a,b; Varaiya 1993; Swaroop and Hedrick 1996, 1999; Seiler et al. 2004; Middleton and Braslavsky 2010; Lin et al. 2012). This problem represents a special instance of more general vehicular formation problems which are encountered in the control of unmanned aerial vehicles, satellite formations, and groups of autonomous robots (Bullo et al. 2009; Mesbahi and Egerstedt 2010). Even for the simplest control objective, in which it is desired to maintain a constant cruising velocity and a constant distance between the neighboring vehicles, it has been long recognized that limited information exchange between the vehicles imposes fundamental performance limitations for control design. In particular, look-ahead strategies that rely only on relative spacing information with respect to the preceding vehicle suffer from *string instability*. This phenomenon is characterized by unfavorable amplification of disturbances downstream the vehicular string (Swaroop and Hedrick 1996, 1999; Seiler et al. 2004; Middleton and Braslavsky 2010). In order to avoid this unfavorable spatial application, it is typically required to broadcast the state of the leader to the rest of the formation.



Vehicular Chains, Fig. 1 Finite and infinite strings of vehicles

While a precise characterization of fundamental performance limitations in the control of vehicular formations is still an open question, in this article we review recent progress in this area. We begin by highlighting performance limits that arise even in optimally controlled vehicular strings. The LQR problem for vehicular strings was originally formulated in pioneering papers by Levine and Athans (1966) and Melzer and Kuo (1971a,b). These formulations were revisited in Jovanović and Bamieh (2005) where it was shown that the time constant of the optimally controlled closed-loop system increases linearly with the number of vehicles. This reference also employed spatially invariant theory (Bamieh et al. 2002) to demonstrate the lack of exponential stability in the limit of an infinite number of vehicles and to explain the arbitrarily slowing rate of convergence observed in numerical studies of finite strings of increasing sizes. We then summarize a recent result that viewed vehicular strings as the 1D version of vehicular formations on regular lattices in arbitrary spatial dimensions and established fundamental performance limitations of spatially invariant localized feedback strategies with relative position measurements (Bamieh et al. 2012). It was shown that it is impossible to achieve robustness to stochastic disturbances with only localized feedback in 1D and 2D; yet this can be achieved in 3D. This is a consequence of the fact that, in 1D and 2D, local feedback laws are ineffective in guarding against disturbances with slow temporal variations and large spatial wavelength. An “accordion” type of motion experienced by these spatiotemporal modes compromises formation throughput, and it may occur even in formations that are string stable. Since the phenomenon that we describe also occurs in distributed averaging algorithms,

global mean first passage time of random walks, effective resistance in electrical networks, and statistical mechanics of harmonic solids, it is relevant for a broad class of networked dynamical systems.

Optimal Control of Vehicular Strings

We next summarize a linear quadratic regulator problem for vehicular strings (Levine and Athans 1966; Melzer and Kuo 1971a,b) and demonstrate that strategies that penalize only relative position errors between neighboring vehicles yield nonuniform rates of convergence towards the desired formation (Jovanović and Bamieh 2005). In particular, the time constant of the optimally controlled closed-loop system increases linearly with the number of vehicles, and the formation loses exponential stability in the limit of infinite vehicular strings.

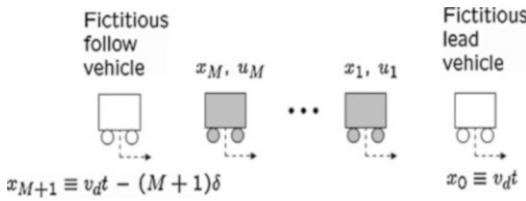
Optimal Control of Finite Strings

A string consisting of M identical unit mass vehicles is shown in Fig. 1a. Each vehicle is modeled as a point mass that obeys the double-integrator dynamics:

$$\ddot{x}_n = u_n, \quad n \in \{1, \dots, M\} \quad (1)$$

where x_n is the position of the n th vehicle and u_n is the control applied on the n th vehicle. A control objective is to provide the desired constant cruising velocity $\bar{v}$ and to keep the constant distance δ between the neighboring vehicles. By introducing the absolute position and velocity error variables

$$\begin{aligned} p_n(t) &:= x_n(t) - \bar{v}t + n\delta \\ v_n(t) &:= \dot{x}_n(t) - \bar{v}, \quad n \in \{1, \dots, M\} \end{aligned}$$



Vehicular Chains, Fig. 2 Finite string with fictitious lead and follow vehicles

system (1) can be brought into the state-space form (Melzer and Kuo 1971a, b):

$$\begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u =: A\psi + Bu \quad (2)$$

where $p := [p_1 \cdots p_M]^T$, $v := [v_1 \cdots v_M]^T$, and $u := [u_1 \cdots u_M]^T$.

Following Melzer and Kuo (1971a, b), fictitious lead and follow vehicles, respectively, indexed by 0 and $M+1$, are added to the formation; see Fig. 2. These two vehicles are constrained to move at the desired velocity $\bar{v}$, and the relative distance between them is assumed to be equal to $(M+1)\delta$ for all times. A quadratic performance index that penalizes control effort, relative position, and absolute velocity error variables is associated with system (2):

$$\begin{aligned} J &= \frac{1}{2} \int_0^\infty \left(\sum_{n=1}^{M+1} q_p (p_n(t) - p_{n-1}(t))^2 \right. \\ &\quad \left. + \sum_{n=1}^M (q_v v_n^2(t) + r u_n^2(t)) \right) dt \\ &= \frac{1}{2} \int_0^\infty (\psi^*(t) Q \psi(t) + u^*(t) R u(t)) dt. \quad (3) \end{aligned}$$

The control problem (2) and (3) is in the standard LQR form with the state and control weights:

$$Q := \begin{bmatrix} Q_p & 0 \\ 0 & q_v I \end{bmatrix}, \quad Q_p := q_p T_M, \quad R := r I.$$

Here, T_M is an $M \times M$ symmetric Toeplitz matrix with the first row given by $[2 \ -1 \ 0 \ \cdots \ 0] \in \mathbb{R}^M$.

We next briefly summarize the explicit solution to the LQR problem (2) and (3) and refer the reader to Jovanović and Bamieh (2005) for additional details. By performing a spectral decomposition of the Toeplitz matrix T_M ,

$$\begin{aligned} T_M &= U \Lambda_T U^*, \quad U U^* = U^* U = I \\ \Lambda_T &= \text{diag}\{\lambda_1(T_M), \dots, \lambda_M(T_M)\} \\ \lambda_n(T_M) &= 2 \left(1 - \cos \frac{n\pi}{M+1} \right), \quad n \in \{1, \dots, M\} \end{aligned} \quad (4)$$

the solution to the LQR algebraic Riccati equation can be represented as

$$\begin{aligned} P &:= \begin{bmatrix} P_1 & P_0 \\ P_0 & P_2 \end{bmatrix}, \quad P_0 = U \Lambda_0 U^*, \quad P_2 = U \Lambda_2 U^*, \\ P_1 &= U \Lambda_1 U^*. \end{aligned} \quad (5)$$

Here,

$$\begin{aligned} \Lambda_0 &= \sqrt{r q_p} \Lambda_T^{1/2} \\ \Lambda_2 &= \sqrt{r} \left(2 \sqrt{r q_p} \Lambda_T^{1/2} + q_v I \right)^{1/2} \\ \Lambda_1 &= \sqrt{q_p} \Lambda_T^{1/2} \left(2 \sqrt{r q_p} \Lambda_T^{1/2} + q_v I \right)^{1/2} \end{aligned}$$

and the eigenvalues of the closed-loop A -matrix are determined by the solutions to the following system of the uncoupled quadratic equations:

$$\begin{aligned} s_n^2 + b_n s_n + c_n &= 0, \quad n \in \{1, \dots, M\} \\ c_n &:= (\lambda_n(T_M) q_p / r)^{1/2} \\ b_n &:= (2c_n + q_v / r)^{1/2}. \end{aligned} \quad (6)$$

From the above expression, it can be shown that in large-scale formations, the least-stable eigenvalue of the closed-loop system approaches the imaginary axis at the rate that is inversely proportional to the number of vehicles. As can be seen from the PBH detectability test, this is because the pair (Q, A) gets closer to losing its detectability as the number of vehicles increases. This clearly indicates that the resulting optimal control strategy leads to closed-loop systems with arbitrarily slow decay rates as the number of vehicles increases. As summarized in section “Optimal Control of Infinite Strings,” the

absence of a uniform rate of convergence for a finite number of vehicles manifests itself as the absence of exponential stability in the limit of infinite vehicular strings.

Optimal Control of Infinite Strings

The LQR problem for a system of identical unit mass vehicles in an infinite string (see Fig. 1b) was originally studied in Melzer and Kuo (1971a). As summarized below, using the theory for spatially invariant linear systems (Bamieh et al. 2002), it was shown in Jovanović and Bamieh (2005) that the resulting LQR controller does not provide exponential stability of the closed-loop system due to the lack of detectability of the pair (Q, A) .

The infinite dimensional equivalent of (2) is given by

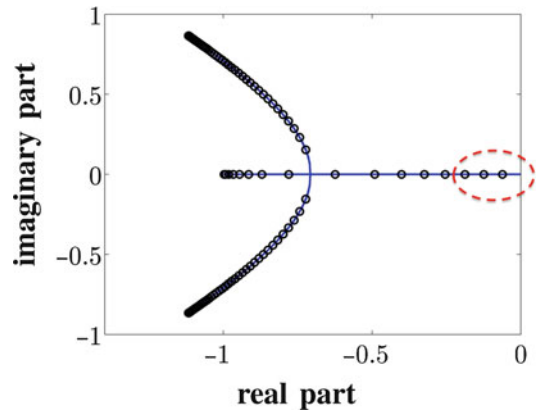
$$\begin{bmatrix} \dot{p}_n \\ \dot{v}_n \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_n \\ v_n \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u_n =: A_n \psi_n + B_n u_n, \quad n \in \mathbb{Z} \quad (7)$$

$$J = \frac{1}{2} \int_0^\infty \sum_{n \in \mathbb{Z}} \left(q_p (p_n(t) - p_{n-1}(t))^2 + q_v v_n^2(t) + r u_n^2(t) \right) dt \quad (8)$$

with q_p , q_v , and r being positive design parameters. Spatial invariance over a discrete spatial lattice $\mathbb{Z}$ can be used to establish that the solution to the LQR problem does not provide an exponentially stabilizing feedback for system (7). In particular, the spectrum of the closed-loop generator in an LQR-controlled spatially invariant string of vehicles (7) with performance index (8) is given by the solutions to the following θ -parameterized quadratic equation:

$$\begin{aligned} s_\theta^2 + b_\theta s_\theta + c_\theta &= 0, \\ c_\theta &:= (2(q_p/r)(1 - \cos \theta))^{1/2} \\ b_\theta &:= (2c_\theta + q_v/r)^{1/2} \end{aligned} \quad (9)$$

where $\theta \in [0, 2\pi)$ denotes the spatial wave number. By comparing (4), (6) and (9), we see that the closed-loop eigenvalues of the finite string are points in the spectrum of the closed-loop infinite



Vehicular Chains, Fig. 3 The spectra of the closed-loop generators in LQR-controlled finite (symbols) and infinite (solid line) strings of vehicles with $M = 50$ and $q_p = q_v = r = 1$. The closed-loop eigenvalues of the finite string are points in the spectrum of the closed-loop infinite string. As the number of vehicles increases, the number of eigenvalues that accumulate in the vicinity of the stability boundary gets larger and larger

string. Furthermore, from these equations it follows that as the size of the finite string increases, this set of points becomes dense in the spectrum of the infinite string closed-loop A -operator. The spectrum of the closed-loop generator, shown in Fig. 3 for $q_p = q_v = r = 1$, illustrates the absence of exponential stability.

Coherence in Large-Scale Formations

Fundamental performance limitations arising from the use of local feedback in networks subject to stochastic disturbances were recently examined in Bamieh et al. (2012). For consensus and vehicular formation control problems in topology of regular lattices, it was shown that it is impossible to guarantee robustness to stochastic exogenous disturbances in one and two spatial dimensions. Yet it was proved that this is achievable in 3D. This phenomenon is a consequence of the fact that, in 1D and 2D, local feedback laws are ineffective in guarding against disturbances with large spatial wavelength, and it has also been observed in global mean first passage time of random walks, effective resistance in electrical networks, and statistical

mechanics of harmonic solids. We next briefly summarize the implications of these results for the control of vehicular formations and refer the reader to Bamieh et al. (2012) for details.

Stochastically Forced Vehicular Formations with Local Feedback

Let us consider $M := N^d$ identical vehicles arranged in a d -dimensional torus, Z_N^d , with the double integrator dynamics:

$$\ddot{x}_n = u_n + w_n \quad (10)$$

where $n := (n_1, \dots, n_d)$ is a multi-index with each $n_i \in Z_N := \{0, \dots, N-1\}$, u is the control input, and w is a mutually uncorrelated white stochastic forcing. Each position vector x_n is a d -dimensional vector with components $x_n := [x_n^1 \cdots x_n^d]^T$. The control objective is to have the n th vehicle follow the absolute desired trajectory $\bar{x}_n$:

$$\bar{x}_n := \bar{v}t + n\delta \Leftrightarrow \begin{bmatrix} \bar{x}_n^1 \\ \vdots \\ \bar{x}_n^d \end{bmatrix} := \begin{bmatrix} \bar{v}^1 \\ \vdots \\ \bar{v}^d \end{bmatrix} t + \begin{bmatrix} n_1 \\ \vdots \\ n_d \end{bmatrix} \delta.$$

In other words, it is desired that all vehicles move with constant heading velocity $\bar{v}$ while maintaining their respective position in a Z_N^d grid with spacing of δ in each dimension.

By introducing the position and velocity deviations from the desired trajectory,

$$p_n := x_n - \bar{x}_n, \quad v_n := \dot{x}_n - \bar{v}$$

and by confining our attention to static-feedback policies,

$$u(t) = -[K_p \ K_v] \begin{bmatrix} p(t) \\ v(t) \end{bmatrix} \quad (11)$$

equations of motion for the controlled system (10) can be brought into the state-space form

$$\begin{aligned} \begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} &= \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \\ w &=: A\psi + Bw \\ z &=: C\psi. \end{aligned} \quad (12)$$

Here, p and v are the position and velocity vectors of all vehicles, z is the performance output, and w is the forcing vector.

An Example

In one-dimensional formations with nearest neighbor relative position and velocity measurements, the control acting on the n th vehicle is given by

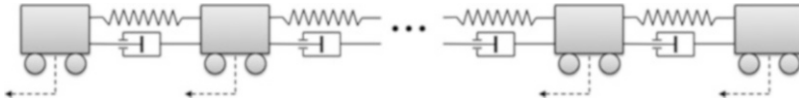
$$\begin{aligned} u_n(t) = & -k_p^-(p_n(t) - p_{n-1}(t)) \\ & -k_p^+(p_n(t) - p_{n+1}(t)) \\ & -k_v^-(v_n(t) - v_{n-1}(t)) \\ & -k_v^+(v_n(t) - v_{n+1}(t)) \end{aligned} \quad (13)$$

where $k_p^\pm$ and $k_v^\pm$ are positive design parameters. For a system that evolves over a 1D lattice, the feedback gain matrices K_p and K_v are tridiagonal Toeplitz matrices implying that the closed-loop systems have been effectively converted into a mass-spring-damper system shown in Fig. 4. Figure 5 shows the results of a stochastic simulation for the closed-loop system (12) and (13) with 100 vehicles with desired inter-vehicular spacing $\delta = 20$ and $k_p^\pm = k_v^\pm = 1$. These plots indicate the lack of formation coherence. This is only discernible when one “zooms out” to view the entire formation. The length of the formation fluctuates stochastically, but with a distinct slow temporal and long spatial wavelength signature. In contrast, the zoomed-in view in Fig. 5 shows a relatively well-regulated vehicle-to-vehicle spacing. In general, small-scale (both temporally and spatially) disturbances are well regulated, while large-scale disturbances are not. This indicates that a local feedback strategy (13) cannot regulate against large-scale disturbances.

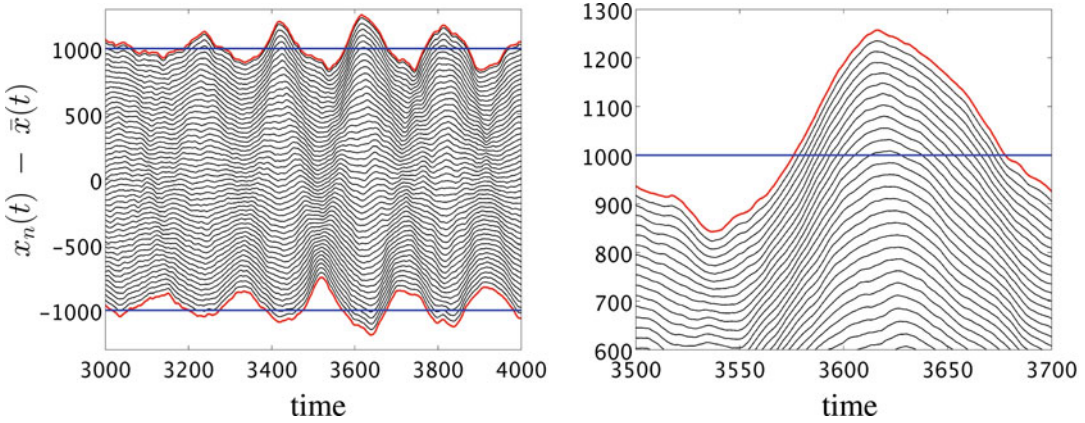
Structural Assumptions

We now list the assumptions on the operators K_p , K_v , and C in (12) under which asymptotic scaling trends summarized in section “Scaling of Variance per Vehicle with System Size” are obtained.

- (A1) **Spatial invariance.** Operators K_p , K_v , and C in (12) are spatially invariant with respect to Z_N^d .



Vehicular Chains, Fig. 4 Finite string of vehicles with a nearest neighbor relative position and velocity feedback



Vehicular Chains, Fig. 5 Position trajectories of a stochastically forced formation with 100 vehicles controlled with nearest neighbor strategy (13). *Left plot*

demonstrates accordion-like motion of the entire formation; *right plot* shows that vehicle-to-vehicle distances are relatively well regulated

- (A2) **Spatial localization.** The feedback (11) uses only local information from a neighborhood of width $2q$, where q is independent of N .
- (A3) **Reflection symmetry.** The interactions between vehicles exhibit mirror symmetry.
- (A4) **Coordinate decoupling.** For $d \geq 2$, control in each coordinate direction depends only on measurements of position and velocity error vector components in that coordinate.

While assumptions (A3) and (A4) were made to simplify calculations, assumptions (A1) and (A2) were essential for the developments in Bamieh et al. (2012).

Performance Measures

We next examine the dependence of the steady-state variance of stochastically forced system (12) on the number of vehicles. In the presence of relative position or velocity measurements, the matrix A in (12) is not necessarily Hurwitz, and the state ψ may not have finite steady-state

variance. However, for connected networks, the performance output z that does not penalize the motion of the mean will have finite steady-state variance; this is because the modes of A at the origin will be unobservable from z . The steady-state variance of z ,

$$V := \sum_{n \in \mathbb{Z}_N^d} \lim_{t \rightarrow \infty} \mathcal{E} \left(z_n^T(t) z_n(t) \right) \quad (14)$$

is quantified by the square of the H_2 norm of the system (12) from w to z , and it can be determined from the solution of the algebraic Lyapunov equation.

We next summarize two different performance measures for stochastically forced vehicular formations.

- (P1) **Local error.** This is a measure of the difference of neighboring vehicles positions from the desired spacing. In 1D, the performance output of the n th vehicle is given by

$$z_n := p_n - p_{n-1}.$$

In d -dimensions, the performance output vector contains as its components the local error in each respective dimension. Since this output involves quantities local to any vehicle within a formation, the corresponding steady-state variance is referred to as a *microscopic performance measure*, V_{micro} .

- (P2) **Deviation from average.** This is a measure of the deviation of each vehicle's position error from the average of the overall position error.

$$z_n := p_n - \frac{1}{M} \sum_{j \in \mathbb{Z}_N^d} p_j. \quad (15)$$

Since this output determines deviation from average, and thereby quantities that are far apart in the network, the corresponding steady-state variance is referred to as a *macroscopic performance measure*, V_{macro} .

Scaling of Variance per Vehicle with System Size

We next summarize asymptotic bounds for both microscopic and macroscopic performance measures derived in Bamieh et al. (2012). The upper bounds result from simple feedback laws similar to the one given in (13). In the situations where either absolute position or velocity measurement are available, additional terms proportional to p_n and v_n will appear in (13). The lower bounds have been obtained for any linear static feedback

control policy satisfying the structural assumptions (A1)–(A4) and the following constraint on control variance at each vehicle:

$$\mathcal{E}(u_n^T u_n) \leq U_{\max}. \quad (16)$$

Under this constraint, the equivalence between scaling trends of lower and upper bounds can be established. As illustrated in Table 1, the dependence of the asymptotic bounds on the number of vehicles is strongly influenced by the underlying spatial dimension d .

Since the macroscopic performance measure captures how well the formation regulates against large-scale disturbances, the scaling results presented in Table 1 demonstrate that local feedback with relative position measurements is unable to regulate against these large-scale disturbances in 1D. To the contrary, in higher spatial dimensions, local feedback can regulate against large-scale disturbances and provide formation coherence. As shown in Table 1, the “critical dimension” needed to achieve network coherence depends on the type of feedback strategy: dimension 3 for relative position and absolute velocity feedback and dimension 5 for relative position and velocity feedback.

Summary and Future Directions

For stochastically forced vehicular formations in topology of regular lattices, we have summarized fundamental performance limitations resulting

Vehicular Chains, Table 1 Asymptotic scalings of microscopic and macroscopic performance measures in terms of the total number of vehicles $M = N^d$, the spatial

dimensions d , and the control effort per vehicle $U_{\max}$. Quantities listed are up to a multiplicative factor that is independent of M or $U_{\max}$:

Feedback type	V_{micro}/M	V_{macro}/M
Absolute position	$\frac{1}{U_{\max}}$	$\frac{1}{U_{\max}}$
Absolute velocity		
Relative position	$\frac{1}{U_{\max}}$	$\frac{1}{U_{\max}} \begin{cases} M & d = 1 \\ \log(M) & d = 2 \\ 1 & d \geq 3 \end{cases}$
Absolute velocity		
Relative position		$\frac{1}{U_{\max}^2} \begin{cases} M^3 & d = 1 \\ M & d = 2 \\ M^{1/3} & d = 3 \\ \log(M) & d = 4 \\ 1 & d \geq 5 \end{cases}$
Relative velocity	$\frac{1}{U_{\max}^2} \begin{cases} M & d = 1 \\ \log(M) & d = 2 \\ 1 & d \geq 3 \end{cases}$	

from the use of local feedback. Even for formations that are string stable, local feedback is not capable of guarding against slowly varying disturbances with long spatial wavelength in 1D and 2D. The observed phenomenon also arises in distributed averaging and estimation algorithms, global mean first passage time of random walks, effective resistance in electrical networks, and statistical mechanics of harmonic solids. Since performance measures that we used to quantify robustness to disturbances are easily extensible to networks with arbitrary topology and more complex node dynamics, they can be used to evaluate performance of a broad class of networked dynamical systems in future studies.

Cross-References

- [Averaging Algorithms and Consensus](#)
- [Flocking in Networked Systems](#)
- [Networked Systems](#)
- [Oscillator Synchronization](#)

Bibliography

- Bamieh B, Paganini F, Dahleh MA (2002) Distributed control of spatially invariant systems. *IEEE Trans Autom Control* 47(7):1091–1107
- Bamieh B, Jovanović MR, Mitra P, Patterson S (2012) Coherence in large-scale networks: dimension dependent limitations of local feedback. *IEEE Trans Autom Control* 57(9): 2235–2249
- Bullo F, Cortés J, Martínez S (2009) Distributed control of robotic networks. Princeton University Press, Princeton
- Jovanović MR, Bamieh B (2005) On the ill-posedness of certain vehicular platoon control problems. *IEEE Trans Autom Control* 50(9):1307–1321
- Levine WS, Athans M (1966) On the optimal error regulation of a string of moving vehicles. *IEEE Trans Autom Control* AC-11(3):355–361
- Lin F, Fardad M, Jovanović MR (2012) Optimal control of vehicular formations with nearest neighbor interactions. *IEEE Trans Autom Control* 57(9):2203–2218
- Melzer SM, Kuo BC (1971a) Optimal regulation of systems described by a countably infinite number of objects. *Automatica* 7:359–366
- Melzer SM, Kuo BC (1971b) A closed-form solution for the optimal error regulation of a string of moving vehicles. *IEEE Trans Autom Control* AC-16(1): 50–52
- Mesbahi M, Egerstedt M (2010) Graph theoretic methods in multiagent networks. Princeton University Press, Princeton
- Middleton RH, Braslavsky JH (2010) String instability in classes of linear time invariant formation control with limited communication range. *IEEE Trans Autom Control* 55(7):1519–1530
- Seiler P, Pant A, Hedrick K (2004) Disturbance propagation in vehicle strings. *IEEE Trans Autom Control* 49(10):1835–1842
- Swaroop D, Hedrick JK (1996) String stability of interconnected systems. *IEEE Trans Autom Control* 41(2):349–357
- Swaroop D, Hedrick JK (1999) Constant spacing strategies for platooning in automated highway systems. *J Dyn Syst Meas Control* 121(3):462–470
- Varaiya P (1993) Smart cars on smart roads: problems of control. *IEEE Trans Autom Control* 38(2):195–207

Vibration Control System Design for Buildings

Hidekazu Nishimura

Graduate School of System Design and Management, Keio University, Yokoyama, Japan

Abstract

This entry provides building vibration control utilizing the mechanism of energy dissipation and seismic isolation including full active control devices and semi-active or passive devices. Vibration control of buildings subjected to dynamic loadings such as large earthquakes, strong winds, or heavy traffic is one of the most important factors to take into consideration for occupant safety. Since energy dissipation is the key technology in vibration control, many kinds of devices have been developed for structural mitigation. Seismic retrofit of buildings is critical because long-period earthquakes occur at considerable distances from the seismic center. Here, we introduce the application of specific devices to the vibration control of real buildings, especially in Japan, where there are many earthquakes.

Keywords

Vibration control · Base isolation · Seismic response control · Energy dissipation · Active control · Semi-active control · Seismic retrofit

Introduction

Vibration control of buildings subjected to dynamic loadings such as large earthquakes, strong winds, or heavy traffic is one of the most important factors to consider to secure the buildings' users. Energy dissipation is the key technology in vibration control, and many kinds of devices have been developed for structural mitigation (Spencer and Nagarajaiah 2003; Soong and Spencer 2002). In Japan, the 2011 earthquake occurred on the Pacific coast of Tohoku, which lasted for an extended period to the Tokyo area 400 km away from the seismic center, and caused fatal damages to the buildings of the surrounding areas.

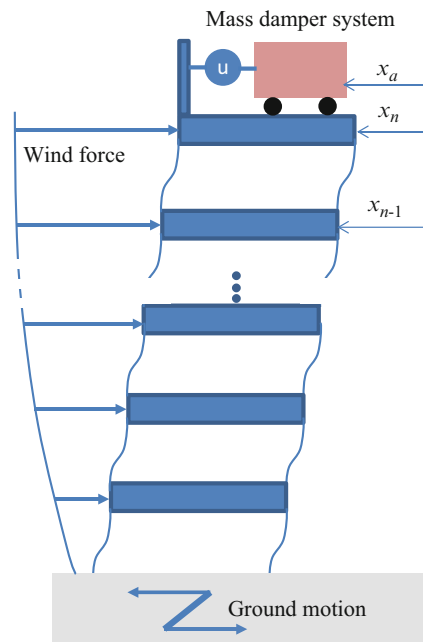
Therefore, seismic retrofitting of buildings is critical in Japan because long-period earthquakes occur at sites far away from their seismic center as well. In particular, old super-high-rise buildings are concentrated in the central ward of Tokyo, Shinjuku, and they have been built on the basis of the theory of flexible structures. During a long-period earthquake, those super-high-rise buildings have very large displacement (about 0.5 m) because of resonance vibration and may need a few minutes to dissipate the structural vibrations. These buildings need to be retrofitted by adding some energy dissipating devices, such as active mass dampers (AMDs), tuned mass dampers, rotating inertial mass dampers, and passive/semi-active base isolation devices.

This entry provides building vibration control utilizing the mechanism of energy dissipation and seismic isolation including full active control devices and semi-active or passive devices. We introduce the application of specific devices to control vibration of real buildings.

Active Mass Damper

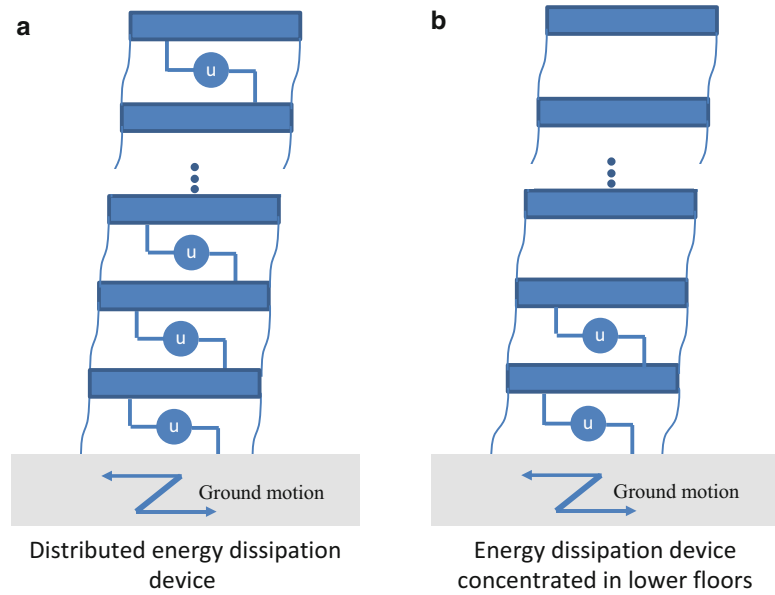
Active, semi-active, or passive mass damper systems have been installed in a large number of high-rise buildings as shown in Fig. 1 (Spencer and Nagarajaiah 2003; Soong and Spencer 2002). Tuned mass damper (TMD) is well known in the application to vibration control of buildings as a passive mass damper whose construction is same as a dynamic vibration absorber (Connor 2003). The dynamical model of building-like structures to be used for vibration control design is generally a multi-degree-of-freedom model. If the modal analysis of vibration is applied to the model, the first mode has a fundamental frequency, and the mode shape is such as shown in Fig. 1. Since the displacement of the top mass of the building is largest in the first mode shape, TMD equipped with on the top mass can effectively absorb the first mode vibration.

In 1979, Citicorp Center has been equipped with TMD with a hydraulic system controlled



Vibration Control System Design for Buildings, Fig. 1 Active, semi-active, or passive mass damper system installed in an n -storied building subjected to wind force and ground motion

Vibration Control System Design for Buildings, Fig. 2 Seismic retrofitting using rotational inertia mass dampers. (a) Distributed energy dissipation device. (b) Energy dissipation device concentrated to lower floors



so as to provide the desired natural frequency and the desired damping to suppress vibration of the building under wind excitation (Petersen 1980). Although active mass dampers have historically used electro-mechanical ball-screw-type actuators, the IHI Corporation has now developed AMDs driven by a linear motor, making the production of a long stroke type easier than that in the case of the ball-screw-type actuator (Koike et al. 2011). Other advantages of using a linear actuator are lesser noise and vibration, light weight, and compactness. Thanks to these advantages, it is expected that linear motor-type AMDs will be installed in existing buildings as seismic retrofitting devices. To avoid reaching the stroke length limit of the actuator because of a large earthquake, a displacement control of the mass is usually applied using a phase lead compensation in response to the displacement signal.

The two AMDs have been installed in the DoCoMo Tohoku Building of Japan in the same direction to improve by 9.5% the damping ratio of the first mode of the translational vibration. The weight of the mass is 20,000 kg, and the total weight of the device is about 25,000 kg. The control experiment was performed by exciting the building with AMDs, and a damping ratio of 11%

was obtained by activating the vibration control with AMDs. Another application of AMD with a linear motor is for a suspension bridge during the construction to mitigate vortex-induced vibration caused by a wind and vibration generated by climbing crane operation (Inoue and Kawakami 2015).

Rotational Inertia Mass Damper

Shimizu Corporation modified the super-high-rise building (height=100m) in the Shibaura ward of Tokyo, Japan, by installing rotational inertia mass dampers. The rotational inertia mass damper (Sugimoto et al. 2013) has a mechanism consisting of a ball screw and a rotational inertia mass, with which the relative translational displacement between stories can be changed to rotational motion of the damper to efficiently increase the dissipation of the kinetic energy.

Although in previous seismic retrofitting many dampers have been distributed in each floors as shown in Fig. 2a, in a concentrated way, Shimizu Corporation located the rotational inertia mass dampers on the lower floors of the building (e.g., 1–7) as shown in Fig. 2b. The seismic response against the 2011 Tohoku earthquake would now

be reduced by about 35% not only relative to the maximum displacement but also to the maximum acceleration of the top floor. Moreover, the duration time was reduced to 220 s instead of 400 s. The method of retrofitting super-high-rise buildings is very unique because the lower floors behave as isolation layers of the base isolation system. Although the displacement of the lower floors becomes slightly larger than that before the retrofit, the whole building has a good seismic response performance.

Semi-active Base Isolation

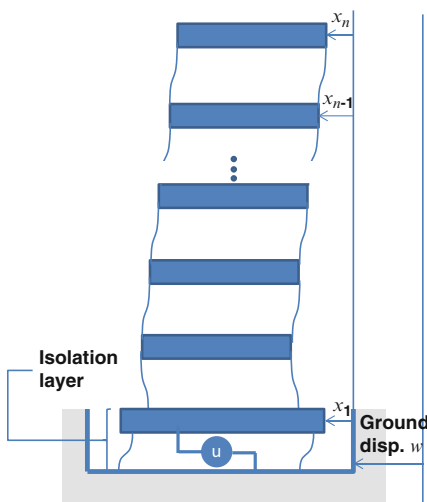
The semi-active base isolation system as shown in Fig. 3 has been mounted in a building for the first time in 2000. The building is located on the Yagami campus of the Keio University in Yokohama, Japan, and the base isolation system in two directions consists of eight semi-active hydraulic dampers that can change the damping coefficient in four steps using a controllable orifice. The maximum damping force is 640 kN, while the switching law of damping coefficients is based on the optimal bilinear control theory. The damper is modeled on the lines of the Maxwell model,

where a spring and a damper are connected in series, and the objective function on the kinetic energy of the building and the constraint function of the squared damping force are adopted (Yoshida and Fujio 2000).

In 2008, another type of semi-active base isolation system has been installed by Collaboration Complex in the Hiyoshi campus of the Keio University in Yokohama, Japan. The system consists of eight semi-active dampers along with eight conventional hydraulic fluid dampers in each direction of the X–Y axes. While the maximum force of the semi-active damper and the conventional hydraulic fluid damper is about 1,000 kN, the semi-active damper can change the damping coefficient in two steps, high side 3.68 MNs/m and low side 1.23 MNs/m. When an earthquake manifests, the high damping coefficient in normal status is switched to the low side. This switch enables the suppression of the acceleration response of the building at the early stages of the earthquake. After the early stage, according to the acceleration response filtered on the isolation layer, the low damping coefficient should be switched to the high side again to avoid the collision of the building with the foundations.

Magneto-rheological (MR) fluid dampers have been studied by many Researchers, and in 2001, two 300 kN MR fluid dampers have been installed in Nihon-Kagaku-Miraikan, the Tokyo National Museum of Emerging Science and Innovation. Similarly in 2003, 400 kN MR fluid dampers have been installed in a residential building in Japan (Fujitani et al. 2003).

Although MR fluid dampers have been controlled by various laws (Jansen and Dyke 2000), a gain-scheduled control method was introduced to control the electric current generated by the electromagnet of the MR damper (Nishimura et al. 2002). A system controlled by a damping force is a bilinear control system, where the control input depends not only on the relative velocity but also on the damping coefficient. A virtual semi-active damper model was proposed that is capable of changing the damping coefficient with the valve open ratio, which is assumed to be governed by the input to the dynamics of second-order system.



Vibration Control System Design for Buildings,
Fig. 3 Semi-active base isolation system

In this device, the optimized variable damping coefficient is determined by the input. Moreover, the valve opening ratio is limited to certain values to constrain the damping force to the maximum value. However, the controllability of the bilinear control system using the variable damping force depends on the relative velocity of the damper. If the relative velocity equals zero, then the system is uncontrollable. Thus, the systems relative to the positive and the negative sides of the relative velocity are separated. Furthermore, the current–force relationship of the MR damper is considered.

The control method using an MR damper was verified on a 9-m-high building-like structure. The structure had four degrees of freedom and a total weight of about 33,000 kg. The MR damper has a maximum force of about 40 kN, and its current–force relationship is nonlinear (Watakabe et al. 2008). The experimental results demonstrated a good seismic isolation performance in comparison with the skyhook control. The gain-scheduled control proposed gently varied the damping force according to the input current.

Full Active Base Isolation

Full active base isolation systems have been studied by many researchers (Nishimura and Kojima 1999) who observed that they are affected by the saturation of the force generated by the actuator following a large earthquake. The seismic isolation performance should be held even though the force saturation occurred. To control the vibrations, it was proposed to use a hyperbolic function for representing the saturation to smooth the input force (Itagaki and Nishimura 2005).

In 2010, the Obayashi Corporation implemented the active base isolation system in real buildings (Endo et al. 2011). Two hydraulic actuators are connected to the building through a spring in each direction of the X–Y axes to avoid the transmission of the high-frequency vibration from the actuator to the building. The control system is based on the displacement control of the hydraulic actuator and achieves absolute

seismic control. The control force is necessary to eliminate the spring and damper forces in the isolation layer, and the skyhook damper force is added to the control force for the stabilization of the whole system.

A trigger mechanism using a friction damper is equipped with serial hydraulic actuators and can avoid the transmission of the excess input force from the actuator to the building. If the excess input force is generated from the actuator in fail, the friction damper can absorb a force of about 1,000 kN so as not to damage the building and the actuator itself. The maximum force of the hydraulic actuator is 1,100 kN, the maximum displacement of the hydraulic actuator is 200 mm, the maximum displacement of the lead–rubber bearing is 500 mm, the maximum displacement of the trigger mechanism with the friction damper is 750 mm, the spring constant of the connected spring is 16,300 kN/mm, and the maximum stroke is 58 mm. Compared to passive isolation, simulations demonstrated that the base isolation system performed well, especially during earthquakes with maximum acceleration less than 200 cm/s^2 .

Energy Harvesting from Vibration Control Device

Energy dissipation is done when a mechanical damper converts its kinetic energy to thermal energy. Energy harvesting technique utilizes electromagnetic induction to store the kinetic energy of a mass damper or an inertia mass damper as the electric energy (Zuo and Tang 2013). A mass damper system with a spring vibrating with a dominant frequency determined by the theory of dynamic vibration absorber is suitable to recover the electricity from the kinetic energy of the mass.

Summary and Future Directions

Seismic retrofitting may become increasingly important to protect buildings from large and long-period earthquakes. The optimization of the

structural mitigation as a whole system must be the objective of future studies aiming to achieve an effective energy dissipation and seismic isolation of buildings. Energy harvesting from vibration control or three-dimensional isolation devices is also an active topic of research.

Cross-References

- [H-Infinity Control](#)
- [Linear Quadratic Optimal Control](#)
- [LMI approach to Robust Control](#)
- [Modeling of Dynamic Systems from First Principles](#)
- [Stochastic Linear-Quadratic Control](#)

Recommended Reading

Vibration control system design for buildings has been summarized in many journal papers as years go by Spencer and Sain (1997), Soong and Spencer (2002), and Spencer and Nagarajaiah (2003) give applications of vibration control systems to buildings or bridges to support social infrastructures. Rossetto and Duffour (2012) and Saatcioglu (2012) give resistant design or structural mitigation to earthquake in structural control including passive devices.

Bibliography

- Connor JJ (2003) Tuned mass damper systems. In: Introduction to structural motion control. Upper Saddle River, N.J.: Prentice Hall Pearson Education
- Endo F, Yamanaka M, Watnabe T, Kageyama M, Yoshida O et al (2011) Advanced technologies applied at the new “Techno Station” building in Tokyo, Japan. *Struct Eng Int* 21(4):508–513
- Fujitani H, Sodeyama H et al (2003) Development of 400 kN magnetorheological damper for a real base-isolated building. In: Proceeding SPIE. 5052, Smart structures and materials 2003: damping and isolation, pp 265–276
- Inoue M, Kawakami T et al (2015) Izmit Bay suspension bridge – finding and consideration for vibration control of tower by active mass damper. In: IABSE conference – Structural engineering: providing solutions to global challenges, Geneva, 23–25 Sept 2015
- Itagaki N, Nishimura H (2005) Disturbance-accommodating gain-scheduled control taking account of actuator saturation. In: Proceedings of the 2005 IEEE conference on control applications, Toronto, 28–31 Aug 2005
- Jansen LM, Dyke SJ (2000) Semi-active control strategies for MR dampers: a comparative study. *J Eng Mech* 126(8):795–803
- Koike Y, Imaseki M, Kazama M (2011) Vibration control using rail-guided full-active mass dampers and the application thereof to high-rise buildings. In: Proceedings of the 5th international symposium on wind effects on buildings and urban environment
- Nishimura H, Kojima A (1999) Seismic isolation control for a buildinglike structure. *IEEE Control Syst Mag* 19(6):38–44
- Nishimura H et al (2002) Semi-active vibration isolation control for multi-degree- of-freedom structures. In: ASME 2002 pressure vessels and piping conference, seismic engineering, vol 2, Paper No. PVP2002-1446, pp 189–196
- Petersen NR (1980) Design of large scale tuned mass damper. In: Leipholz HHE (ed) Structural control, Proceedings of the international IUTAM symposium on structural control held at the University of Waterloo, Ontario, 4–7 June 1979. North-Holland
- Rossetto T, Duffour P (2012) Earthquake resistant design. *Encyclopedia of natural hazards*, 12 Oct 2012, pp 1–13
- Saatcioglu M (2012) Structural mitigation. *Encyclopedia of natural hazards*, 17 Sep 2012, pp 1–25
- Soong TT, Spencer BF Jr (2002) Supplemental energy dissipation: state-of-the-art and state-of-the practice. *Eng Struct* 24:243–259
- Spencer BF Jr, Nagarajaiah S (2003) State of the art of structural control. *J Struct Eng* 2003:845–856
- Spencer BF Jr, Sain MK (1997) Controlling buildings: a new frontier in feedback. *IEEE Control Syst Mag* 17(6):19–35
- Sugimoto K, Fukukita A et al (2013) Response reduction effect of using inertial mass dampers in a base isolated structure. In: 13th world conference on seismic isolation, energy dissipation and active vibration control of structures – commemorating JSSI 20th anniversary -, Sendai, 24–27 Sep 2013
- Watakabe M, Inoue N, Nishimura H et al (2008) Response control performance of semi-active isolation system using the GS control for a multi-degree-of-freedom structure with magneto-rheological fluid damper. *J Struct Constr Eng* 73(628):875–882 (in Japanese)
- Yoshida K, Fujio T (2000) Semi-active base isolation for a building structure. *Int J Comput Appl Technol* 13(1/2):52–58
- Zuo L, Tang X (2013) Large-scale vibration energy harvesting. *J Intell Mater Syst Struct* 24(11):1405–1430

Vision-Based Motion Estimation

Nicholas Gans¹ and Kaveh Fathian²

¹University of Texas at Arlington, Arlington, TX, USA

²Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

Vision-based motion estimation is the problem of recovering the rotation and translation (i.e., pose) of a moving camera or object from camera images. Algorithms that solve this problem are a key component of many control and robotic applications. Traditionally, the pose estimation problem is solved using image feature points, which must satisfy the rigid motion constraint. We explain how reformulation of the rigid motion constraint leads to different pose estimation algorithms. We discuss the advantages and drawbacks of these methods, benchmark their accuracy and runtime, and provide reference to open-source implementations.

Keywords

Camera pose estimation · Homography · Essential matrix · Five-point algorithms

Introduction

Cameras, as passive and information-rich sensors, have become an indispensable part of many robotic and control applications. Autonomous robots often rely on fusing the motion estimates obtained from vision and inertial sensors to improve the localization and navigation accuracy. In visual servoing, the position and orientation of an object estimated from visual feedback can be used to guide a robotic arm to grab or manipulate objects. Camera pose estimation is also a key component in many simultaneous localization and mapping pipelines, both for initial localization and for loop closure detection.

The first practical developments in the pose estimation problem occurred in the 1980's, with the landmark introduction of the eight-point (Longuet-Higgins 1981) and four-point (Faugeras and Lustman 1988) algorithms. These, majority of subsequent algorithms, exploit image feature points, which are regions of the image that can be systematically identified and matched across images; examples are corners, edges, blobs, etc. Fig. 1 shows an example where feature points in two images (captured at slightly different viewpoints from a static scene) are detected and matched as indicated by yellow lines. This example also shows the Cartesian coordinates of some of the feature point pairs on the image plane. Pose estimation algorithms solve the problem using the rigid motion constraint, which relates changes in feature point coordinates to camera pose changes.

Other than the feature-based algorithms discussed here, methods such as optical flow can be used to estimate the motion. This technique relies on the assumption that the camera motion is small; hence, large motions between the viewpoints generally cannot be recovered. In recent years, learning-based strategies have been used to directly estimate the relative pose from images. However, currently, these methods do not match the accuracy of the feature-based methods.

Background

Consider images of a static scene captured by a camera at two viewpoints (e.g., Fig. 1). Alternatively, we can consider two images of a moving object and a static camera with minimal changes to the proceedings. Assume that feature points are detected and matched across images. This can be done systematically using algorithms such as Lowe et al. (1999). The pixel locations of features can be mapped into Cartesian coordinates on the image plane using the *camera calibration matrix*, which is a constant, full-rank matrix that can be computed using a calibration routine. More specifically, if $\mathbf{m}_{\text{pix}} \in \mathbb{R}^3$ denotes the homogeneous pixel coordinate of a feature (obtained by appending a 1 to the 2D x - y pixel



Vision-Based Motion Estimation, Fig. 1 An illustrative example showing pairs of matched feature points across two images and their Cartesian coordinates on the image plane. (Image courtesy of Mathworks)

vector), then the Cartesian coordinate is given by $\mathbf{m} = \mathbf{K}^{-1} \mathbf{m}_{\text{pix}}$, where $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ is the camera calibration matrix. The calibration matrix can compensate for geometric distortions caused by the camera lens, making the *pinhole model* a suitable description of how a camera depicts a 3D scene. Under the pinhole camera model, an image of a 3D point is formed where the light ray that passes through the lens' focal point intersects the image plane. This is illustrated in Fig. 2, where a 3D point is projected (to points $\mathbf{m}$, $\mathbf{m}'$) on the image planes of a camera at two viewpoints. Physically, the image plane is behind the focal point, and the image is inverted. Mathematically, it is equivalent but simpler to consider the image plane in front of the focal point.

Let a camera frame be defined with origin at the focal point, and x - y axes parallel to and z -axis perpendicular to the image plane. Further, let $\mathbf{R} \in \text{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$, respectively, represent the relative rotation and translation of the camera frame between viewpoints. For each matched feature point, the *rigid motion constraint* must hold and is given by

$$u \mathbf{R} \mathbf{m} + \mathbf{t} = u' \mathbf{m}', \quad (1)$$

where $\mathbf{m}, \mathbf{m}' \in \mathbb{R}^3$ are the Cartesian (homogeneous) coordinates of the feature points on the image planes. Positive scalars $u, u' \in \mathbb{R}_+$ are *depths* of the 3D point at each camera frame, and as illustrated in Fig. 2, are the distance from the 3D point to the focal point along the z -axis of

each camera frame. In (1), the point coordinates $\mathbf{m}$ and $\mathbf{m}'$ are known from the images, and the unknowns we seek to recover are $u, u', \mathbf{R}$, and $\mathbf{t}$.

Pose estimation algorithms leverage (1) for multiple matched feature points to recover the unknowns. Note that translation and depths of the points can only be recovered up to a *scale factor*, as (1) can be multiplied by any scalar and still hold. In other words, from images, we can only recover the direction of the translation and not its magnitude. This fundamental limitation cannot be overcome unless additional information, such as size of an object in the image, is available.

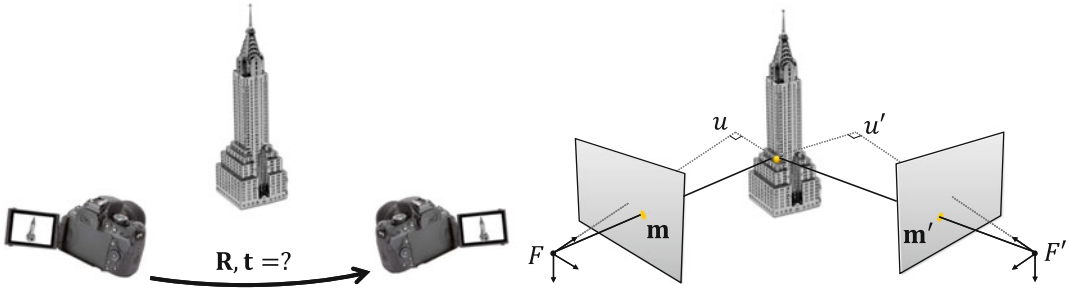
General Approaches to Pose Estimation

Pose estimation techniques can be based on 1) the essential matrix, 2) direct estimation of the rotation and translation, or 3) the homography matrix under the assumption that the 3D points lie on a plane. We explain these approaches and discuss their advantages and drawbacks.

Essential Matrix

Consider the rigid body constraint in (1).

Let $\mathbf{T}_\times \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$ be the skew-symmetric matrix made from the elements of $\mathbf{t} \stackrel{\text{def}}{=} [t_x, t_y, t_z]^T$. Since $\mathbf{T}_\times \mathbf{t} = \mathbf{0}$, multiplying both sides of (1) by $\mathbf{T}_\times$ gives



Vision-Based Motion Estimation, Fig. 2 Projection of a 3D point onto the image plane at two viewpoints based on the pinhole camera model

$$u T_{\times} R m = u' T_{\times} m'. \quad (2)$$

Due to skew-symmetry, for any $m' \in \mathbb{R}^3$, it holds that $m'^{\top} T_{\times} m' = 0$. Consequently, multiplying both sides of (2) by $m'^{\top}$ and dividing by u give the *epipolar constraint*

$$m'^{\top} T_{\times} R m = m'^{\top} E m = 0, \quad (3)$$

where $E \stackrel{\text{def}}{=} T_{\times} R$ is called the *essential matrix* (Longuet-Higgins 1981).

In (3), coordinates m, m' correspond to one pair of matched feature points. By using the epipolar constraint for multiple matched pairs, existing methods solve for E and subsequently decompose it to recover R and t . Depths cannot be recovered directly from E , as u, u' do not appear in (3). However, there are a number of methods to recover depths using the recovered R and t .

There are several issues to consider when using the essential matrix to estimate camera motion. Since depths do not appear in (3), the scale of the translation vector recovered from E does not have any intrinsic relation with the distance of the feature points to the camera. This can cause problem (e.g., chattering) in control applications where t is used in feedback. Another related consequence is the multiplicity of solutions. It can be seen that E and $-E$ both satisfy (3). This can be shown to imply that $(R, t), (R, -t), (R^{\top}, t)$, and $(R^{\top}, -t)$ are all mathematically valid solutions. If the depths for all these solution pairs have been recovered, then

typically, only one solution will have all recovered depths terms positive. Since a negative depth would correspond to the physically impossible situation of a point being behind the camera, the correct solution is easily determined. Lastly, if the motion is a pure rotation (i.e., $t = 0$), then $E = T_{\times} R = 0$, and the epipolar constraint becomes ill-defined.

Algorithms

Let $e_1, e_2, \dots, e_9$ denote the (unknown) elements of E . Expanding (3) gives a linear equation in terms of these variables for each pair of matched feature points. The eight-point algorithm (Longuet-Higgins 1981) leverages eight point pairs to obtain a system of eight linear equations, where variables are consequently recovered from the null space of the system's coefficient matrix. This algorithm is computationally efficient, as it has a linear formulation. On the downside, it can suffer from degeneracies, such as when five or more feature points are coplanar, as this makes the system's coefficient matrix more rank deficient.

It can be shown that the essential matrix can be solved from a minimum of five matched feature points (in general positions in the space). Based on (3) and additional algebraic constraints that E must satisfy, five-point algorithms recover E from a (nonlinear) system of polynomial equations (Stewénius et al. 2006). These methods do not suffer from the degeneracies that are associated with the eight-point algorithm. However, they are generally slower and admit more solu-

tions (up to ten solutions for $\mathbf{E}$). To find the correct solution, a third camera view or an additional feature pair is required.

Direct Estimation

Consider three matched feature points $(\mathbf{m}_i, \mathbf{m}'_i)$, $i = 1, 2, 3$, and the rigid motion constraint (1) written for each pair as $u_i \mathbf{R} \mathbf{m}_i + \mathbf{t} - u'_i \mathbf{m}'_i = \mathbf{0}$. Subtracting the constraints corresponding to $i = 2, 3$ from $i = 1$ eliminates the unknown translation and gives

$$\begin{aligned} u_1 \mathbf{R} \mathbf{m}_1 - u'_1 \mathbf{m}'_1 - u_2 \mathbf{R} \mathbf{m}_2 + u'_2 \mathbf{m}'_2 &= \mathbf{0}, \\ u_1 \mathbf{R} \mathbf{m}_1 - u'_1 \mathbf{m}'_1 - u_3 \mathbf{R} \mathbf{m}_3 + u'_3 \mathbf{m}'_3 &= \mathbf{0}. \end{aligned} \quad (4)$$

By rewriting (4) in the matrix-vector form we obtain

$$\underbrace{\begin{bmatrix} \mathbf{R} \mathbf{m}_1 - \mathbf{m}'_1 & -\mathbf{R} \mathbf{m}_2 & \mathbf{m}'_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{R} \mathbf{m}_1 - \mathbf{m}'_1 & \mathbf{0} & \mathbf{0} & -\mathbf{R} \mathbf{m}_3 & \mathbf{m}'_3 \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} u_1 \\ u'_1 \\ u_2 \\ u'_2 \\ u_3 \\ u'_3 \end{bmatrix} = \mathbf{0}, \quad (5)$$

which (since the depths are nonzero) implies that matrix $\mathbf{M}$ is rank deficient and

$$\det(\mathbf{M}) = 0. \quad (6)$$

Constraint (6), which relates $\mathbf{R}$ and feature coordinates $(\mathbf{m}_i, \mathbf{m}'_i)$, is independent of translation or depths. Based on (6), existing methods directly solve for $\mathbf{R}$ and substitute it in the rigid motion constraint to obtain a system of linear equations in terms of the translation and depth variables. From solving this linear system, translation and depths are recovered up to a common (but unknown) scale factor, which is suitable for control applications. Furthermore, when six or more feature points are used, the recovered rotation solution is unique, which leads to a unique translation solution with positive depths.

Algorithms

By representing the entries of $\mathbf{R}$ as $r_1, r_2, \dots, r_9$, and expanding (6), a polynomial equation in the unknown rotation entries is obtained from any three matched feature pairs. The resulting

system of polynomial equations can be solved to recover the rotation. Alternatively, the quaternion representation of rotation, which uses four unknowns and reduces the number of variables, can be leveraged (Fathian et al. 2018).

Similar to the essential matrix algorithms, methods that recover $\mathbf{R}$ directly need a minimum of five feature points. This minimal number of points generally leads to the multiple solution hypotheses, which requires an additional point to identify the correct solution.

Homography

Consider the rigid motion constraint in (1), and assume that the 3D feature points are *coplanar*. Assume that in camera frame F , the plane containing the feature points has normal vector $\mathbf{n} \in \mathbb{R}^3$ and lies at a distance $d \in \mathbb{R}_+$ from the origin. The identity $d = \mathbf{n}^\top (u \mathbf{m})$ must hold for each point in F . Multiplying $\mathbf{t}$ by $u/d \mathbf{n}^\top \mathbf{m} = 1$ and factoring terms gives

$$\mathbf{m}' = \frac{u}{u'} \left(\mathbf{R} + \frac{\mathbf{t}}{d} \mathbf{n}^\top \right) \mathbf{m} = \alpha \mathbf{H} \mathbf{m}, \quad (7)$$

where $\alpha \stackrel{\text{def}}{=} u'/u$ and $\mathbf{H} \stackrel{\text{def}}{=} \mathbf{R} + 1/d \mathbf{t} \mathbf{n}^\top$ is the *Euclidean homography matrix*.

Given four or more matched points, existing methods solve for $\mathbf{H}$ and decompose it into $\mathbf{R}$, $1/d \mathbf{t}$, $\mathbf{n}$, and α (Faugeras and Lustman 1988). Note that $\mathbf{t}$ is recovered up to the unknown scale d , and only the depth ratio α can be recovered. There are four $(\mathbf{R}, \mathbf{t})$ solutions for a given $\mathbf{H}$, two of which will generally have points behind the camera. This means there are two solutions that are physically valid, and additional information must be used to determine the correct solution.

Homography methods require that all feature points be coplanar and will fail if this condition is not met. Many human-made environments have planar surfaces, so this condition is often met. Long distance or high altitude imagery can also make the 3D features sufficiently close to coplanar. Unlike the essential matrix, the homography matrix is well defined even if $\mathbf{t} = \mathbf{0}$. Further, the scale of $\mathbf{t}$ is recovered up to the (unknown)



Vision-Based Motion Estimation, Fig. 3 Example of images in the datasets used for benchmarking the algorithms

constant d , which is preferable to avoid chattering in control applications.

$$\rho(\mathbf{g}, \mathbf{g}^*) = \frac{1}{\pi} \arccos(\mathbf{g}^\top \mathbf{g}^*) \in [0, 1], \quad (8)$$

Algorithms

The four-point algorithm (Faugeras and Lustman 1988) can be used to solve for $\mathbf{H}$. By representing entries of $\mathbf{H}$ as $h_1, h_2, \dots, h_9$, from (7), each point pair gives three constraints in terms of these entries and α . Eliminating α leaves two constraints, which leads to a system of eight linear equations for four matched points. The homography variables are consequently recovered from the null space of the system's coefficient matrix.

Benchmarks

To provide a benchmark for the accuracy and speed of existing algorithms, we use images from four popular real-world datasets: ICL, KITTI, NAIST, and TUM. An example of images in each dataset is shown in Fig. 3. We benchmark the eight-point algorithm of Longuet-Higgins (Longuet-Higgins 1981) and five-point algorithm of Stewenius et al. (2006), which are based on the essential matrix, and the direct estimation methods of Kneip et al. (2012) and Fathian et al. (2018) based on five points. We provide all eight-point and five-point algorithms with the same set of eight matched feature points. Since these points are generally noncoplanar, we do not provide comparisons with the four-point homography algorithm.

The comparison results are shown in Table 1. To reflect statistics about the errors, the median and 0.25 and 0.75 quantiles of the averaged errors are given in the table. The metric used to assess the errors is given by

where for evaluating the rotation error $\mathbf{g} = \mathbf{q}$ represents the estimated rotation in quaternions, for translation error $\mathbf{g} = \mathbf{t}/\|\mathbf{t}\|$ represents the unit norm estimated translation, and the asterisk represents the ground truth values provided by the dataset. The values of $\rho = 0$ and $\rho = 1$, respectively, correspond to zero and maximum errors. The accuracy of the estimated rotation for five-point algorithms outperforms the eight-point algorithm.

Table 2 lists the average execution time of each algorithm in milliseconds, where the minimum required number of points is used in each algorithm. The results are based on 1000 randomly generated feature points and Mex implementation of algorithms in MATLAB on an Intel's 4th Gen i-7 CPU platform. In comparison, the eight-point algorithm has the smallest runtime, whereas five-point algorithms are slower due to their nonlinear nature.

Software

There are many open-source libraries available for feature point detection/matching, camera calibration, and camera motion estimation. Here, we mention a selected few that currently are widely used by the controls and computer vision community.

- **OpenCV:** This is one of the most comprehensive and widely used computer vision libraries, which has cross-platform C++, Python, Java, and MATLAB interfaces. OpenCV is available at <https://opencv.org/>.

Vision-Based Motion Estimation, Table 1 Statistics of the rotation and translation estimation errors. The translation error of Kneip’s algorithm is omitted as it only returns a rotation estimate

		Rotation error $\times 1000$				Translation error $\times 10$		
		Longuet	Stewenius	Kneip	Fathian	Longuet	Stewenius	Fathian
KITTI	Med	0.4697	0.2933	0.8980	0.2155	0.0669	0.0840	0.0596
	Q1	0.2378	0.1724	0.4321	0.1318	0.0383	0.0443	0.0358
	Q3	0.9129	0.4913	1.9060	0.3452	0.1256	0.1899	0.1063
TUM	Med	1.8289	1.0415	2.0601	1.0099	2.3068	2.1682	1.4550
	Q1	1.0999	0.6559	1.2689	0.6371	1.2453	1.0206	0.7945
	Q3	3.0120	1.6797	3.3846	1.5916	3.4994	3.4727	2.5377
ICL	Med	1.0092	0.6545	1.1879	0.6202	2.2591	2.0429	1.4507
	Q1	0.6093	0.4049	0.7424	0.3937	1.3115	1.0151	0.7085
	Q3	1.6024	0.9601	1.9744	0.9356	3.3328	3.0430	2.3593
NAIST	Med	0.8191	0.7480	0.8487	0.4759	0.2827	0.5727	0.2700
	Q1	0.4885	0.3941	0.5012	0.2607	0.1773	0.2681	0.1555
	Q3	1.3768	1.2932	1.6832	0.8338	0.4212	1.4121	0.4614

- **OpenGV:** This is a collection of computer vision routines for solving geometric vision problems including camera pose estimation. This library provides C++ and MATLAB implementations and is available at <https://github.com/laurentkneip/opengv>.
- **OpenMVG:** This is a C++ library especially targeted to the multi-view geometry problems and designed to provide easy access to the classical solvers. This library is available at <https://github.com/openMVG/openMVG>.
- **MVTB:** This is a MATLAB library that contains implementations of feature detection/matching, camera calibration, and motion estimation algorithms. This library is available at <http://petercorke.com/wordpress/toolboxes/machine-vision-toolbox>.

A MATLAB implementation for generating the benchmark results reported in this note is available at <https://sites.google.com/view/kavehfathian/code/benchmarking-pose-estimation-algorithms>.

Future Directions

Current and future efforts at new feature-based pose estimation algorithms are primarily focused on more robust and computationally efficient

Vision-Based Motion Estimation, Table 2 Average execution time of algorithms in milliseconds

Algorithm	Average execution time (ms)
Longuet (8-point)	0.0428
Stewenius (5-point)	0.1046
Kneip (5-point)	0.9298
Fathian (5-point)	0.8445

minimal-point algorithms. There is also related work in the field of “multi-view geometry,” which uses more than two viewpoints to estimate relative poses of camera views and the 3D structure of the scene. Optimization routines are often employed to refine estimates from methods discussed here, a process known as *bundle adjustment*. Current research in this domain includes improving the computational efficiency. Advances in deep learning have led to the exploration of methods that use camera images directly rather than extracted and matched feature points.

Cross-References

- [Image-Based Estimation for Robotics and Autonomous Systems](#)

- [Image-Based Formation Control of Mobile Robots](#)
- [Image-Based Robot Control](#)
- [Intermittent Image-Based Estimation](#)
- [Networked Control Systems: Estimation and Control over Lossy Networks](#)
- [Robot Visual Control](#)

Recommended Reading

There are a number of excellent books describing the vision-based pose estimation in deeper detail (Hartley and Zisserman 2004; Ma et al. 2003; Corke 2017). The original papers cited throughout this manuscript will also provide greater understanding (Longuet-Higgins 1981; Faugeras and Lustman 1988; Stewénius et al. 2006; Fathian et al. 2018).

Bibliography

- Corke P (2017) *Robotics, vision and control: fundamental algorithms in MATLAB®*, 2nd completely revised, vol 118. Springer
- Fathian K, Ramirez-Paredes JP, Doucette EA, Curtis JW, Gans NR (2018) QuEst: a quaternion-based approach for camera motion estimation from minimal feature points. *IEEE Robot Autom Lett* 3(2):857–864
- Faugeras OD, Lustman F (1988) Motion and structure from motion in a piecewise planar environment. *Int J Pattern Recognit Artif Intell* 2(3):485–508
- Hartley RI, Zisserman A (2004) *Multiple view geometry in computer vision*, 2nd edn. Cambridge University Press
- Kneip L, Siegwart R, Pollefeys M (2012) Finding the exact rotation between two images independently of the translation. Springer
- Longuet-Higgins HC (1981) A computer algorithm for reconstructing a scene from two projections. *Nature* 293(5828):133–135
- Lowe DG et al. (1999) Object recognition from local scale-invariant features. In: *IEEE international conference on computer vision*, vol 99, pp 1150–1157
- Ma Y, Soatto S, Kosecka J, Sastry SS (2003) *An invitation to 3-D vision: from images to geometric models*. Springer Science & Business Media, vol 26
- Stewénius DH, Engels C, Nistér DD (2006) Recent developments on direct relative orientation. *ISPRS J Photogramm Remote Sens* 60(4):284–294

2.5D Vision-Based Estimation

Jian Chen

State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou, China

Abstract

2.5D vision-based techniques, also known as hybrid vision-based techniques, provide flexible ways to estimate the range or velocity of moving objects. The information from both the image space (2D) and the Cartesian space (3D) is simultaneously utilized to construct the system state in this technology, which overcomes the disadvantages of the traditional visual servoing schemes. It has been widely adopted in motion from structure, structure from motion, and structure and motion problems.

Keywords

2.5D vision · Hybrid visual servoing · Vision-based estimation · Motion from structure · Structure from motion · Structure and motion

Introduction

As a class of powerful tools widely utilized in robotics, the 2.5D vision-based techniques arise from visual servoing at the end of the twentieth century (Malis and Chaumette 1999). Visual servoing aims at increasing the flexibility, accuracy, and robustness of a closed-loop robotic system using real-time visual feedback (Hutchinson et al. 1996; Chaumette and Hutchinson 2006; Janabi-Sharifi et al. 2011). Different visual servo algorithms mainly differ in the construction of states. There are two traditional schemes: image-based visual servoing (IBVS) and position-based visual servoing (PBVS). IBVS utilizes the image space

features to construct the closed-loop error system, and PBVS employs the Cartesian space features. However, both schemes have intrinsic drawbacks.

Originally, the 2.5D vision-based techniques are proposed to improve the performance of the traditional IBVS and PBVS schemes. The basic characteristics of the 2.5D vision-based control are simultaneously utilizing the information from both the image space (2D) and the Cartesian space (3D) to construct the states. Therefore, it is also referred to as the hybrid servoing. The 2.5D vision-based control can overcome many shortcomings of IBVS and PBVS because it provides flexible ways to manipulate the translational motion and the rotational motion individually, which greatly facilitates the control development. The classic regulation task of a robotic arm with six degrees of freedoms (DoFs) (Malis and Chaumette 1999, 2000) provides a great tutorial to understand the 2.5D vision-based control.

Due to the effectiveness and advantages of the 2.5D vision-based control, the 2.5D vision-based techniques are also widely adopted in another essential problem: vision-based estimation. Vision-based estimation aims at identifying the unknown key information of an object using visual sensors. There have been many related applications. In the vision-based estimation problem, the measurable variables are utilized as feedback to close the loop. In the estimation problem, the 2.5D vision-based techniques affect the construction of states, which is similar to the control problem. According to distinct design goals, the estimation problems can be roughly divided into motion from structure (MfS) problems, structure from motion (SfM) problems, and structure and motion (SaM) problems.

2.5D Vision-Based Estimation

The dynamic model used in vision-based estimation problems can be formulated as

$$\begin{cases} \dot{s} = f_s(s, v, \phi) \\ y_m = g(s) \end{cases} \quad (1)$$

where $s(t)$ is the system state, $v(t)$ is the camera velocity, ϕ comprises some parameters, and $f_s(\cdot)$ is a function which describes how $s(t)$, $v(t)$, and ϕ determine the evolution of the system state. In (1), $y_m(t)$ is the measurable output, and $g(\cdot)$ describes the relationship between $y_m(t)$ and $s(t)$. In terms of 2.5D vision-based estimation, the system state is generally constructed in the following form:

$$\begin{aligned} s &= [\underbrace{s_t^T}_{\text{Translation}}, \underbrace{s_r^T}_{\text{Rotation}}]^T \\ &= [\underbrace{p^T}_{\text{Image space (2D)}}, \underbrace{\alpha, s_r^T}_{\text{Cartesian space (3D)}}]^T \end{aligned} \quad (2)$$

where $s_t(t) \in \mathbb{R}^3$ and $s_r(t) \in \mathbb{R}^3$ indicate the translational and rotational components, respectively. Specially, $s_t(t)$ consists of $p(t) \in \mathbb{R}^2$ (the image coordinates of the feature point) and $\alpha(t) \in \mathbb{R}$ (a factor related to the depth of the feature point). It is clear that $p(t)$ comes from the image space (2D) and $\alpha(t)$ and $s_r(t)$ come from the Cartesian space (3D).

Motion from Structure

The MfS problem utilizes the known object geometry to estimate the relative motion between the camera and the object, i.e., $s(t)$ is measurable ($y_m(t) = s(t)$) and $v(t)$ remains to be determined. Given that the dynamics $f_s(\cdot)$ is known, the objective can be achieved by estimating $\dot{s}(t)$.

$$\begin{cases} \tilde{s} \triangleq s - \hat{s} \\ \dot{\hat{s}} = h(\tilde{s}). \end{cases} \quad (3)$$

In (3), an appropriately designed observer $h(\cdot)$ can guarantee that $\hat{s}(t) \rightarrow s(t)$ and $\dot{\hat{s}}(t) \rightarrow \dot{s}(t)$. Then, the velocity $v(t)$ can be calculated utilizing the known $f_s(\cdot)$.

The two critical requirements mentioned before ($f_s(\cdot)$ is known and s is measurable) can be easily satisfied by using the 2.5D vision-based techniques with the aid of the homography. For example, Chitrakarana et al. (2005) utilize this

strategy to asymptotically identify the six DoF velocity of a moving object using a single fixed camera. A reference state s_d is introduced and an error signal is constructed as $e(t) = s(t) - s_d$. Then, a computable interaction matrix is derived which relates the variation of the error $\dot{e}$ to the velocity $\mathbf{v}$. Next, a nonlinear continuous estimator is designed to identify $\dot{e}$ asymptotically which is further utilized to determine the velocity provided that a single geometric length between two feature points and the rotation information between the camera and the reference image are known.

Structure from Motion

Contrary to the MfS problem, the SfM problem utilizes the known relative motion between the camera and the object to reconstruct the object geometry. In other words, $\mathbf{v}(t)$ is known, while the unknown structure information may exist in $s(t)$ (when $s(t)$ is partially unmeasurable) or ϕ (when $s(t)$ is measurable). Let the coordinates of a feature point expressed in the camera frame be denoted as $P = (X, Y, Z)^T$ where Z indicates the depth.

When the structure information exists in $s(t)$, the available information comes from both $\mathbf{v}(t)$ and $y_m(t)$. This happens when the factor $\alpha(t)$ takes the absolute depth ($\alpha = Z$ or $\alpha = \ln(Z)$). The observer is designed in the following way:

$$\dot{\hat{s}} = h(y_m, \mathbf{v}). \quad (4)$$

With some necessary conditions satisfied, an appropriate observer can drive the estimation error $\tilde{s}(t)$ to zero, which means $\hat{s}(t) \rightarrow s(t)$. Sometimes, it is not practical to directly estimate the state. Under such circumstances, indirect methods can be adopted in which some auxiliary variables are designed at first and then be used to construct the state estimator. Suppose that the auxiliary variable is denoted as $\zeta(t)$, this strategy can be formulated as

$$\begin{cases} \dot{\zeta} = \psi(\zeta, y_m, \mathbf{v}) \\ \hat{s} = h(\zeta, y_m). \end{cases} \quad (5)$$

Dani et al. (2011) adopt this strategy to recast the structure estimation of a moving object into an unknown input observer design problem. Consequently, a causal algorithm is provided for estimating the structure of a moving object using a moving camera with relaxed assumptions on the object motion.

When the structure information exists in ϕ , $s(t)$ is available ($y_m(t) = s(t)$), and the relationship between $s(t)$ and ϕ should be figured out to estimate ϕ . This situation occurs when $\alpha(t)$ takes the ratio of depths ($\alpha = \frac{Z_d}{Z}$ or $\alpha = \ln\left(\frac{Z_d}{Z}\right)$) because this ratio can be directly calculated by means of homography techniques. Under this circumstance, both $s(t)$ and $\mathbf{v}(t)$ can be utilized to construct the observer for ϕ .

$$\dot{\hat{\phi}} = h(s, \mathbf{v}). \quad (6)$$

This method is adopted in Chen et al. (2011) where the state is decomposed as the multiplication of two matrices and a structure-related vector. This relationship is further utilized to estimate the state-related vector. As a matter of fact, Chen et al. (2011) deals with the SaM problem which will be introduced in detail later.

Structure and Motion

The SaM problem focuses on identifying both the Cartesian coordinates of the feature points and the relative motion information (Dani et al. 2012). In SaM problems, both $s(t)$ and $\mathbf{v}(t)$ are (partially) unmeasurable and to be determined, which leads to two kinds of methods.

The intuitive strategy is using more prior knowledge about either the object or the camera. One factor should be estimated first and then facilitates the estimation of the other one, such as the following form:

$$\begin{cases} \dot{\hat{\mathbf{v}}} = h_1(y_m, \phi) \\ \dot{\hat{s}} = h_2(\hat{\mathbf{v}}, y_m). \end{cases} \quad (7)$$

This strategy is utilized by Chen et al. (2011) based on Chitrakarana et al. (2005). Similar to Chitrakarana et al. (2005), the rotation between

the reference frame and the inertial frame is required to be known as well as the coordinates of a feature point on the object in this work. This strategy is also adopted by Dani et al. (2012). It is required in Dani et al. (2012) that at least one linear velocity of the camera should be known. The angular velocity is estimated using a filter. The estimated angular velocity and a measured linear velocity are combined to estimate the scaled 3D coordinates of the feature points.

The alternative choice to deal with SaM problems is to introduce multiple objects or cameras which conduct different motions, i.e., one is static, while the other one is moving. Then, multiple states and corresponding dynamics become available

$$\begin{cases} \dot{s}_1 = f_{s1}(s_1, \mathbf{v}_1, \phi) \\ \dot{s}_2 = f_{s2}(s_2, \mathbf{v}_2, \phi). \end{cases} \quad (8)$$

In the equations above, if $\mathbf{v}_1(t)$ and ϕ are unknown, all other available variables can be adopted in observers to determine $\mathbf{v}_1(t)$ and ϕ . Although the two dynamics are different, there is common information between them which facilitates the observer development.

$$\begin{cases} \dot{\hat{\mathbf{v}}}_1 = h_1(s_1, s_2) \\ \dot{\hat{\phi}} = h_2(s_1, s_2, \mathbf{v}_2). \end{cases} \quad (9)$$

In (9), $\bar{\mathbf{v}}_1(t)$ is the normalized value of $\mathbf{v}_1(t)$ up to a scale resulting from the unknown absolute range. As the range is contained in $\phi(t)$, $\hat{\phi}$ will further benefit the estimation of $\mathbf{v}_1(t)$.

$$\dot{\hat{\mathbf{v}}}_1 = h_3(\hat{\mathbf{v}}_1, \hat{\phi}). \quad (10)$$

This strategy is adopted in the work of Chwa et al. (2015) where a scene comprising both static and moving objects is introduced to eliminate the dependence on a priori knowledge of the object. Similarly, Chen et al. (2018) extend their previous work Chen et al. (2011) by introducing a static-moving camera configuration to identify the velocity and range of the feature points on a moving object simultaneously. Unlike Chen et al. (2011), it is no longer required that the

coordinates of a feature point on the object should be known in Chen et al. (2018).

Future Directions for Research

The 2.5D vision-based techniques have attracted much interest since they were proposed. However, there are still many aspects that can be improved such as the range of applications and rigorous theoretical analysis.

- The existing applications of 2.5D vision-based techniques have been confined to robotic arms with 6 DoFs. Much effort has been devoted to vision-based control and estimation on wheeled mobile robots (WMRs). However, the existing works are mainly image-based (Mariottini et al. 2007) or position-based (Zhang et al. 2018). Applying the 2.5D vision-based techniques on WMRs is still faced with many challenges due to the nonholonomic constraints.
- Many works mention that the 2.5D-based scheme increases the likelihood that the object will stay in the camera field of view (FoV) (Malis and Chaumette 1999; Chen et al. 2005), but few of them provide sufficiently persuasive and rigorous theoretical analysis. Chen et al. (2007) adopt an image space navigation function together with the 2.5D-based scheme to generate a Cartesian space trajectory which ensures all feature points remain visible. However, the tracking errors may still result in the feature points leaving the FoV. Parikh et al. (2017) provide a different perspective on this issue for reference. They investigate the state estimation without visual feedback when the feature points are out of the FoV by means of switched systems. However, this strategy leads to the stabilization problem of the overall system.
- In many works, the local minimums have not been rigorously analyzed. A finite number of simulations or experiments cannot guarantee the global convergence. Zhang et al. (2019) make a good demonstration of investigating the state space and analyzing the existence of

multiple equilibriums. This helps in figuring out whether the global convergence holds or what the influence of those undesired equilibriums is.

Cross-References

- [Image-Based Estimation for Robotics and Autonomous Systems](#)
- [Image-Based Robot Control](#)
- [Robot Motion Control](#)
- [Robot Visual Control](#)

Bibliography

- Chaumette F, Hutchinson S (2006) Visual servo control part I: Basic approaches. *IEEE Robot Autom Mag* 13(4):82–90
- Chen J, Dawson DM, Dixon WE, Behal A (2005) Adaptive homography-based visual servo tracking for a fixed camera configuration with a camera-in-hand extension. *IEEE Trans Control Syst Technol* 13(5):814–825
- Chen J, Dawson DM, Dixon WE, Chitrakaran VK (2007) Navigation function-based visual servo control. *Automatica* 43(7):1165–1177
- Chen J, Chitrakaran VK, Dawson DM (2011) Range identification of features on an object using a single camera. *Automatica* 47(1):201–206
- Chen J, Zhang K, Jia B, Gao Y (2018) Identification of a moving object's velocity and range with a static-moving camera system. *IEEE Trans Autom Control* 63(7):2168–2175
- Chitrakaran VK, Dawson DM, Dixon WE, Chen J (2005) Identification of a moving object's velocity with a fixed camera. *Automatica* 41(3):553–562
- Chwa D, Dani AP, Dixon WE (2015) Range and motion estimation of a monocular camera using static and moving objects. *IEEE Trans Control Syst Technol* 24(4):1174–1183
- Dani AP, Kan Z, Fischer NR, Dixon WE (2011) Structure estimation of a moving object using a moving camera: An unknown input observer approach. In: 50th IEEE conference on decision and control and European control conference, Orlando, pp 5005–5010
- Dani AP, Fischer NR, Dixon WE (2012) Single camera structure and motion. *IEEE Trans Autom Control* 57(1):241–246
- Hutchinson S, Hager GD, Corke PI (1996) A tutorial on visual servo control. *IEEE Trans Robot Autom* 12(5):651–670
- Janabi-Sharifi F, Deng L, Wilson WJ (2011) Comparison of basic visual servoing methods. *IEEE/ASME Trans Mechatron* 16(5):967–983
- Malis E, Chaumette F (1999) 2-1/2D visual servoing. *IEEE Trans Robot Autom* 15(2):238–250
- Malis E, Chaumette F (2000) 2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *Int J Comput Vis* 37(1):79–97
- Mariottini GL, Oriolo G, Prattichizzo D (2007) Image-based visual servoing for nonholonomic mobile robots using epipolar geometry. *IEEE Trans Robot* 23(1):87–100
- Parikh A, Cheng TH, Chen HY, Dixon WE (2017) A switched systems framework for guaranteed convergence of image-based observers with intermittent measurements. *IEEE Trans Robot* 33(2):266–280
- Zhang K, Chen J, Li Y, Gao Y (2018) Unified visual servoing tracking and regulation of wheeled mobile robots with an uncalibrated camera. *IEEE/ASME Trans Mechatron* 23(4):1728–1739
- Zhang K, Chaumette F, Chen J (2019) Trifocal tensor-based 6-DOF visual servoing. *Int J Robot Res* 38(10–11):1208–1228

Walking Robots

Ambarish Goswami¹ and Katsu Yamane²

¹Intuitive Surgical, Sunnyvale, CA, USA

²Honda Research Institute, Inc., San Jose, CA, USA

Abstract

This entry presents an overview of mobile “walking” robots that use their legs to move from one place to another. Walking robots represent a fascinating class of machines which holds the potential for breakthrough applications and inspires multidisciplinary research with rich scientific content. The key feature that separates walking robots from all other classes of mobile robots is their ability to explore unprepared surfaces using discrete footholds. In this respect, these robots are truly the machine counterparts of biological land animals.

Keywords

Walking robots · Gait · Balance · Humanoid robots

Introduction

The adventure of modern robotics is generally considered to have started from the middle of

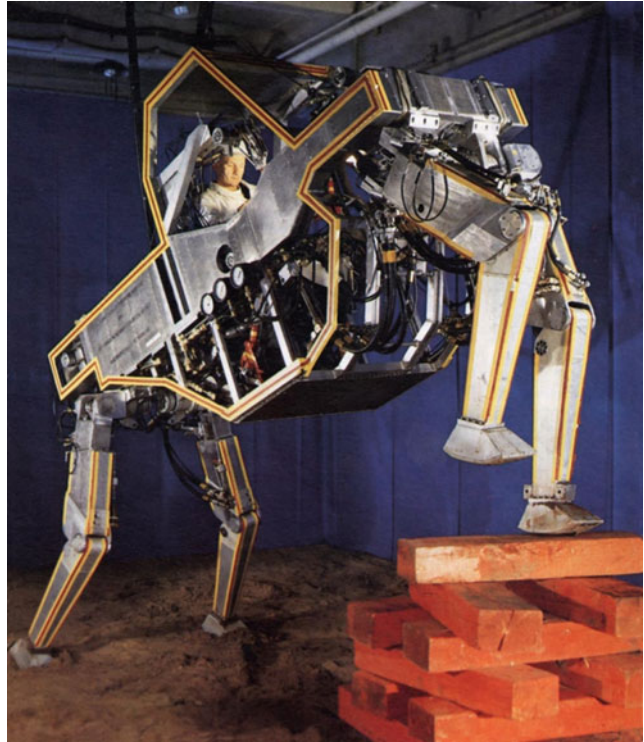
the twentieth century ([International Federation of Robotics \(IFR\)](#)). During the first few decades of this new journey, robots were not mobile. Somewhat similar to trees, these so-called “arm” manipulator robots were securely rooted to a stationary base. The free end of these robots typically consisted of an end-effector “hand” with which a number of mostly manufacturing-related tasks, such as welding, spray painting, and pick-and-place operations, were performed. Life was simple, if a bit boring. However, from the end of 1960s, this started to change.

Fiction writers had earlier imagined a variety of mobile robots such as in “I, Robot” Asimov (1950), Otho Hamilton et al. (1940), and Maria Malone et al. (2004). Scientists and engineers also ventured to build a number of quite sophisticated machines such as the General Electric experimental “walking truck” quadruped robot by Mosher shown in Fig. 1 and the Sparko and Elektro by Westinghouse (<http://en.wikipedia.org/wiki/Elektro>). However, they were not considered truly autonomous in the sense we describe modern robots. Some of the major personalities who are primarily responsible for forever transforming the state of stationary existence of robots and giving them intelligent mobility are Profs. I. Katoh, M. Vukobratovic, and R. McGhee, followed by Prof. M. Raibert.

Because walking robots used legs for locomotion, they immediately became the mechatronic cousins to the entire range of biological legged creatures, starting from tiny insects to large ani-

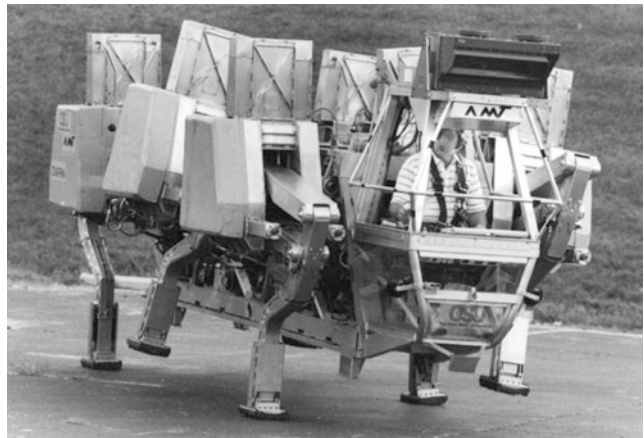
Walking Robots,

Fig. 1 GE “walking truck” developed by Mosher



Walking Robots,

Fig. 2 Adaptive Suspension Vehicle (ASV), OSU



imals. Indeed, today we have robotic versions of spiders and cockroaches, geckoes and lizards, dogs and cheetah, and even humanoids. We have seen very large robots such as the ASV (Waldron and McGhee 1986) shown in Fig. 2 and the Dante (Bares and Wettergreen 1999). We have also seen single-legged robots, which even the mother nature has not considered creating so far.

Early History

The early researchers whom we mentioned above started paving the way for walking robots. These robots walked with their legs, explored their own environments, and sometimes even ventured outside. Once these walking robots started appearing on the scene, life was never the same.

Prof. Katoh pioneered walking robot research at Waseda University (Japan) through a series of remarkable biped humanoid robots, of which WL-5 is credited with genuine bipedal walking and WL-6 with displaying the first dynamic gait. At the same time, Prof. Vukobratovic was conducting research activities in exoskeleton and other areas at the Mihajlo Pupin Institute (former Yugoslavia). He was instrumental in formalizing the concept of dynamic balance and introduced zero moment point (ZMP), which is used to this day (Vukobratović and Juričić 1969; Sardain and Bessonnet 2004). In the USA, Prof. McGhee conducted path-breaking research on computer-controlled machines at the Ohio State University. He created the Ohio hexapod and later, with colleague Prof. Ken Waldron, developed the truly spectacular adaptive suspension vehicle (ASV) hexapod.

Prof. Raibert started building robots in the USA, first at Carnegie Mellon University and then at Massachusetts Institute of Technology (Raibert 1989). With his colleagues he created a series of robots, which unlike their stationary predecessors were characteristically full of energy. Situation permitting, they would occasionally deviate from conventional walking and running and would burst into aerial somersaults and other acrobatic motions. Prof. Raibert continues to actively shape the field of walking robots to the present day; his company Boston Dynamics has introduced a number of high-performance robots, such as Little Dog, Big Dog, R-Hex, Petman, and Atlas.

The hardware, sensing, and control aspects of walking robots were steadily gaining in sophistication during the 1990s. However, except for the new appreciation of walking dynamics in the study of passive bipedal gait (McGeer 1990), there was no unexpected leap in the world of walking robots. This changed in 1996 when Honda publicly announced the humanoid robot P2, the result of their robotics project, till then unknown to the outside world. This was to be superseded by the P3 robot, and then the ASIMO humanoid robot project in 2000, which became another important event in the humanoid robot history.

Characteristics of Walking Robots

Compared to other forms of land locomotion, legged walking possesses the distinct capability of locomotion using discrete footholds (Raibert 1989). Unlike wheeled mobile robots or cars, walking robots do not need a continuous prepared surface such as paved road, trail, or track in order to travel. By virtue of this single feature, a vast extent of land surface, which is not accessible to wheeled robots, opens up to walking robots. Indeed, at least in principle, walking robots are able to reach almost any location, on earth and on other planets, wherever human and other legged creatures can go.

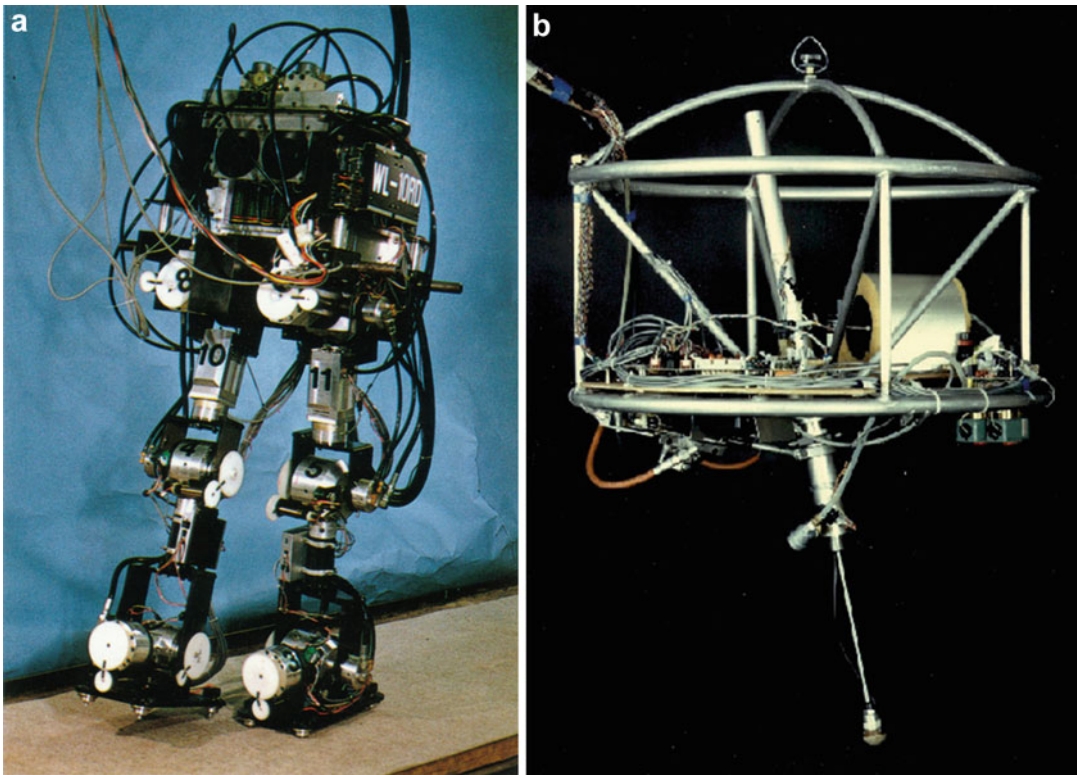
Legged locomotion is natural to terrains where the only means of locomotion must be through the use of unstructured footholds, which can be irregularly spaced both horizontally and vertically. Due to the unique design of the leg, legged creatures can largely isolate the “payload” or the upper body from the geometric details of the terrain profile during locomotion. Both for biological creatures and for walking robots, this brings benefit in the form of significant energy savings. For walking robots this also reduces mechanical stress, vibration, and wear on the system hardware, which makes them suitable for locomotion in rough natural terrain (Fig. 3).

In contrast, wheeled robots are typically faster, mechanically less complex, and energetically more efficient. However, these benefits must be supported by very expensive infrastructure overhead. In many places such expenditure is not practical or not even desirable.

For this reason, legged robots are believed to be useful in applications that require mobility in unstructured, uneven environments such as “last-mile delivery” and disaster response. Examples of bipedal and quadrupedal robots developed for such applications include Digit (Fig. 4a), ANYmal (Fig. 4b), and Spot (Fig. 4c).

Classification of Walking Robots

Walking robots have been built in different sizes and morphologies. These robots have ranged in



Walking Robots, Fig. 3 Early walking robots, (a) Waseda WL-10. (Image courtesy of Atsuo Takanishi) (b) One-Legged Robot. (Image courtesy of Boston Dynamics)

sizes from small hexapods (Lewinger et al. 2005), medium-sized robots (Fig. 9), and relatively large robots such as the BigDog (Raibert et al. 2008) from Boston Dynamics and Toyota iWalk (Fig. 8) and also a few giant robots such as Dante (Bares and Wettergreen 1999) from CMU and the ASV (Waldron and McGhee 1986) from OSU (Fig. 7), as mentioned before. With further miniaturization it is conceivable that we will see even smaller walking robots in the future with unanticipated and surprising application domains. One can also imagine gigantic walking robots in potential applications in large construction sites such as in bridge, building, or ships, but we have not started seeing them just yet.

In terms of the number of legs, we have already seen monopods, Figs. 3b and 5a; bipeds, Fig. 9a, b, c; tripod, Fig. 5b; quadruped, Fig. 6a; b, hexapods, Figs. 5c, d, and 2; octopod, Fig. 5e;

and “centipede” robots with many legs, Fig. 5f. Modular robots could have variable number of legs (Fig. 5g) (Kim et al. 2017).

Other than monopods, robots with odd-numbered legs are curiously absent in this list. Creatures with odd-numbered legs are also not found in the nature. (Although kangaroos, crocodiles, and lizards have been shown to employ their substantial tails as the “fifth leg” for balance maintenance.) It is not clear if an engineering rationale is present behind this trend or the biological inspiration is simply missing for the creators of legged robots.

In addition to size and morphology, walking robots can be classified in terms of the number and types of leg joints, type of gait (e.g., walking or running), or the domain of movement. The next section is devoted to the humanoid robots, which is perhaps the most popular class of walking robots (Fig. 8).



Walking Robots, Fig. 4 Bipedal and quadrupedal robots developed for disaster response and last-mile delivery applications: (a) Digit. (Image courtesy: Agility

Robotics), (b) ANYmal. (Image courtesy: ANYbotics), (c) Spot. (Image courtesy: Boston Dynamics)

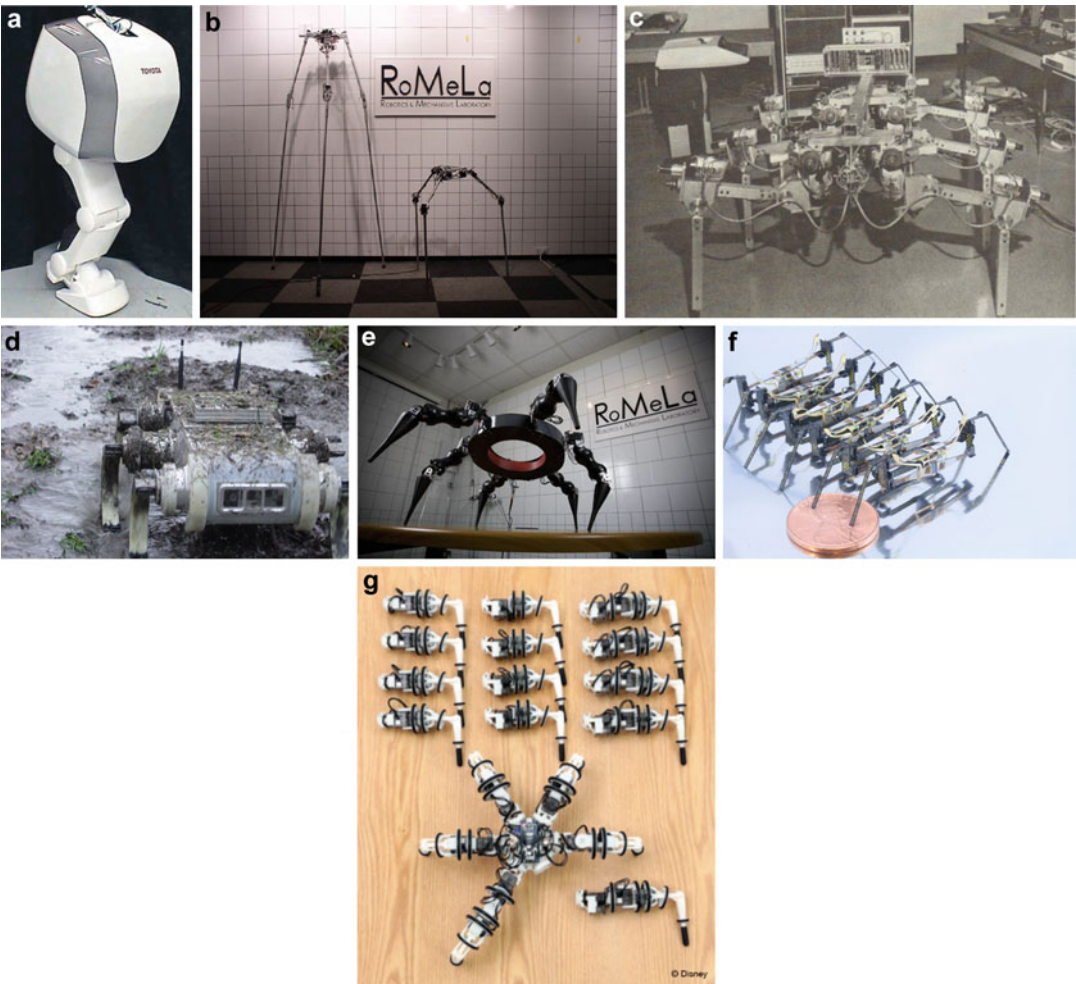
Humanoid Robots

Humanoid robots belong to a unique class of two-legged walking robots that has a special place in the popular psyche (Figs. 9 and 10). These robots are the subject of special affection and fascination due to their similarity with human beings. In fact, humanoid robots might be the original inspiration behind the entire field of robotics and perhaps also its ultimate goal. Being perpetually inspired by movies and novels, a long-standing dream of the human has been to create a mechatronic replica of themselves, the human, which will be fully general purpose and endowed with all human functionalities except perhaps the full independence of thought and action.

Humanoid robots exist in different sizes, including smaller robots such as NAO (Gouaillier et al. 2009), HOAP, and QRIO (Ishida et al. 2004) (Fig. 11) and life-sized robots such as HRP, HUBO, and ASIMO. Despite their differences, these robots bear a close resemblance to the kinematic design and proportions of a human

being and share a common human-mimicking morphology. Indeed, the perceived similarity between humanoid robots and the human is so close that we routinely describe aspects of such robots using anthropomorphic terms. Terms like head, arm, hand, leg, thigh, shank, ankle, spine, gait, stumble, fall, facial expression, and even emotion are hardly ever used to describe any other man-made device.

At current technical level, humanoid robots cannot compete in their actual utility with robots such as Roomba, the vacuum cleaner, the bomb-sniffing robot, or the huge population of fully active and cost-effective welding and spray painting robots. Yet, our fascination with humanoids remains as strong as ever, and novel applications of such robots are continuously being explored. Humanoid robots are currently considered in roles of educators (Yamasaki and Nakagawa 2006; Falconer 2013), dance partners (Kosuge 2010), waiter, baby-sitter, companions for autistic children or for seniors (Robins et al. 2012), and security or emergency response team.



Walking Robots, Fig. 5 Walking robots with different number of legs: (a) monoped, Toyota Hopping Robot; (b) tripod, STriDER, RoMeLa. (Image courtesy: Dennis Hong); (c) large hexapod, McGhee, OSU; (d) R-Hex. (R-Hex Robot image courtesy of Boston Dynamics);

(e) octopod, Spider, RoMeLa. (Image courtesy: Dennis Hong); (f) many legs, Centipede, Harvard; (g) variable number of legs, Snapbot, Disney. (Image courtesy: Disney)



Walking Robots, Fig. 6 Two quadruped robots, (a) Sony Aibo. (Image courtesy of Sony) and (b) BigDog Robot. (Image courtesy of Boston Dynamics)



Walking Robots, Fig. 7 Large walking robots, (a) Dante II, CMU, (b) Ambler, CMU, and (c) John Deere Walking Tractor

Curiously, the functionality of walking is not relevant or central to many of these roles.

Dynamic Equations of Walking Robots

The dynamic equations of a walking robot can be expressed in the following form:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{\tau}_g(\mathbf{q}) = \boldsymbol{\Gamma} + \boldsymbol{\Gamma}_c + \boldsymbol{\Gamma}_{\text{ext}}, \quad (1)$$

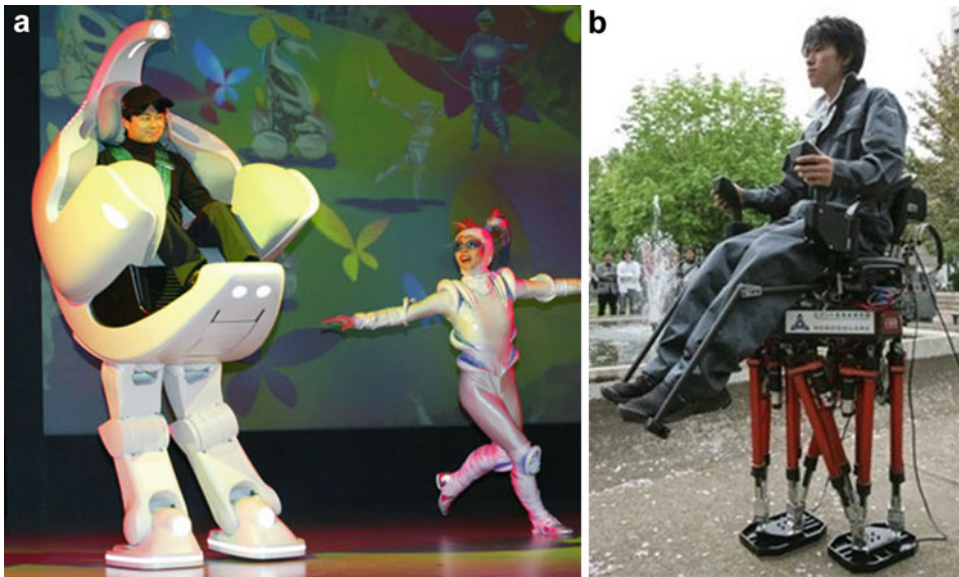
where $\mathbf{q}$ is the vector of the robot's generalized coordinates, which contains the world frame transformation matrix of its base link and all its joint angles. The generalized velocity vector is expressed as $\dot{\mathbf{q}} = [\mathbf{v}_B \ \dot{\boldsymbol{\theta}}]^T$ where $\mathbf{v}_B$ is the base velocity and $\dot{\boldsymbol{\theta}}$ is the vector of joint velocities.

Additionally, $\mathbf{H}$ is the joint-space inertia matrix; $\mathbf{C}$ is the matrix of Coriolis, centrifugal, and gyroscopic terms; and $\boldsymbol{\tau}_g$ is the vector of gravity terms. Finally, $\boldsymbol{\Gamma} = [\mathbf{0} \ \boldsymbol{\tau}]^T$ is the joint torque vector, $\boldsymbol{\Gamma}_c = \mathbf{J}_c^T \mathbf{f}_c$ is the joint torque resulting from the contact forces $\mathbf{f}_c$ such as from the ground, and $\boldsymbol{\Gamma}_{\text{ext}} = \mathbf{J}_e^T \mathbf{f}_e$ is the joint torque due to external interaction forces $\mathbf{f}_e$.

The contact conditions which the robot must satisfy can be written in the form of Eq. 2. The physical constraints due to ground friction, center of pressure (CoP) condition (explained subsequently), torque limits, etc., can be expressed as in Eq. 3:

$$\mathbf{J}_c(\ddot{\mathbf{q}}) = \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}), \quad (2)$$

$$\mathbf{A}[\ddot{\mathbf{q}} \ \boldsymbol{\tau} \ \mathbf{f}_c]^T \leq \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}), \quad (3)$$



Walking Robots, Fig. 8 Novel application of walking robots: Human carrying “chair” robots: (a) iWalk of Toyota and (b) WL-16RV Multi-purpose Biped Locomotor from Waseda University. (Image courtesy Atsuo Takanishi)

The friction condition ensures that the robot feet do not slide on the ground and the CoP condition corresponds to maintaining the resultant of the ground reaction force (GRF) within the perimeter of the support polygon (Sardain and Bessonnet 2004), so that toppling is prevented.

Some of the generalized coordinates of the robot, specifically those which describe the base link of the robot to the world frame, are not powered, as apparent from the joint torque vector representation $\Gamma = [\mathbf{0} \ \boldsymbol{\tau}]^T$, in Eq. 1. In other words, the robot is called *underactuated*. In fact, *all* walking robots are underactuated, and it is one of the central characteristics that sets these robots apart from other robots. Underactuation plays a very important role in the dynamics, motion planning, and control of walking robots (Chevallereau et al. 2005).

Balance

Even after several decades of research, balance maintenance has remained one of the most important issues of walking robots and especially of humanoid robots. Although the basic dynamics of

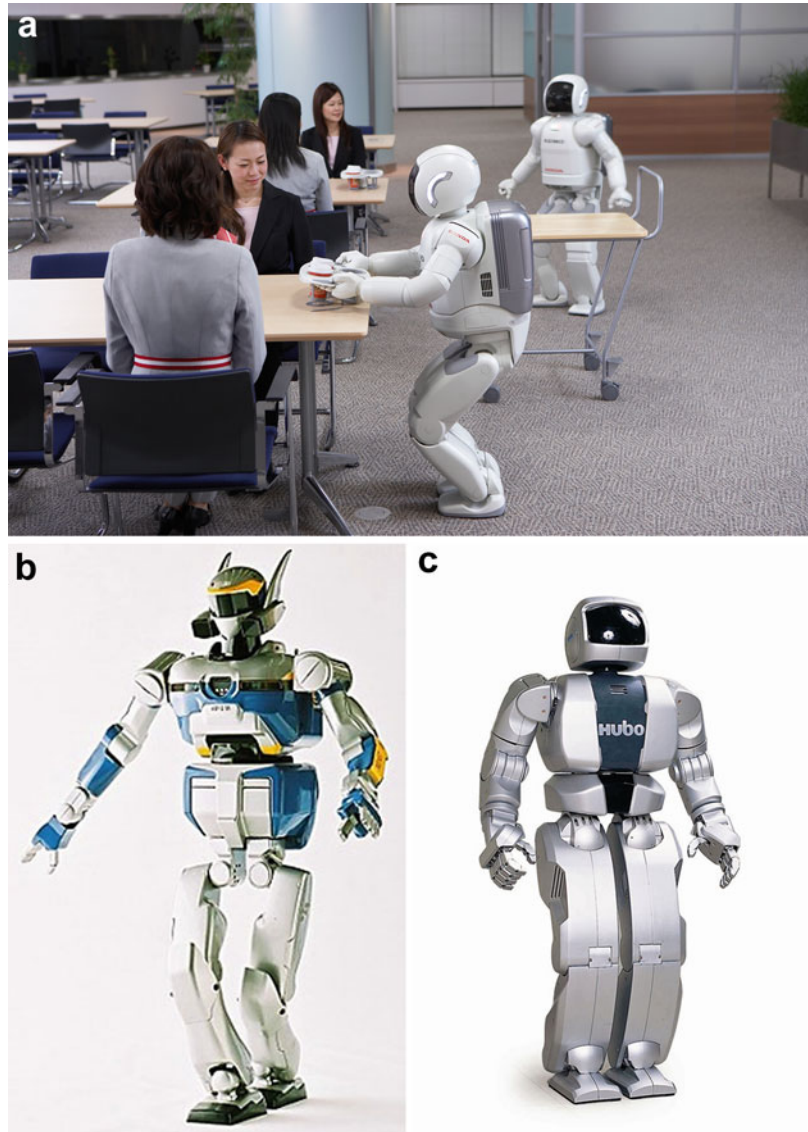
balance are currently understood (Vukobratović and Juričić 1969; Sardain and Bessonnet 2004), robust and general controllers that can deal with discrete and non-level foot support as well as large, unexpected, and unknown external disturbances such as from a moving support, a slip, and a trip are emerging only recently. In comparison with the elegance and versatility of human balance, present-day humanoid robots appear quite deficient.

Balance generally refers to the ability of a walking robot to maintain a sustained gait with a reasonably upright posture without falling (Kajita and Espiau 2008). Robot gait can be static or dynamic. A robot with a static gait would continue to stay upright even if its joints were suddenly frozen. Static gait and movement under static balance are safe, but are slow and lack elegance. A dynamic gait is fluid and natural-looking as it harnesses and exploits the inertial characteristics of the physical robot. However, the robot must be in motion for it to sustain an upright stature. Suddenly locking the joints may cause a fall.

The location and the nature of the resultant GRF on the support polygon of the robot have

Walking Robots, Fig. 9

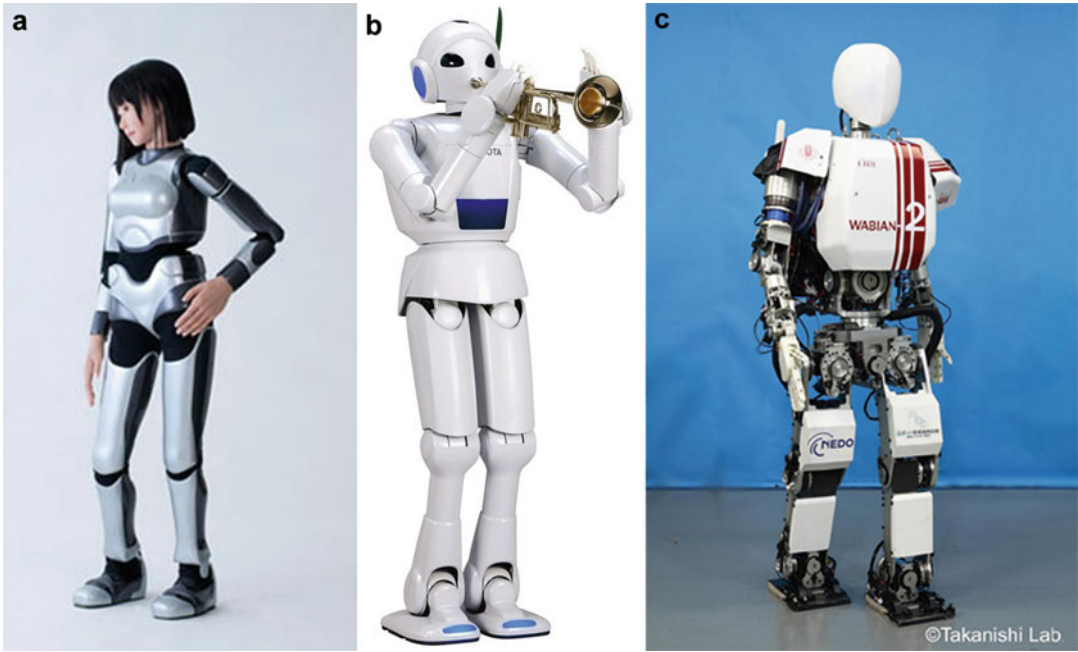
Three well-known human size humanoid robots. (a) ASIMO, Honda. (b) HRP-2, AIST. (Image courtesy of AIST). (c) HUBO, Korea



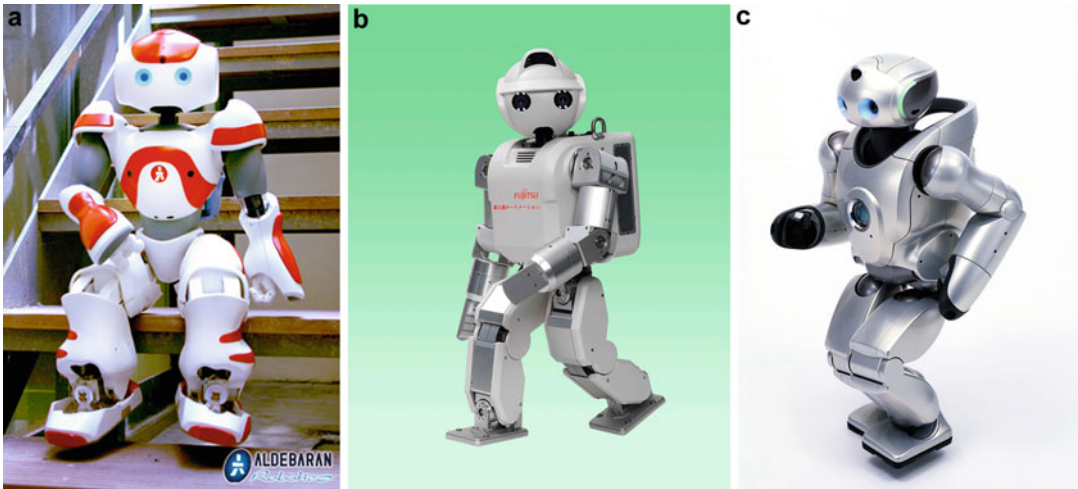
been traditionally used to interpret the dynamic state of the robot's movement. The point where the resultant GRF acts on the robot is called its zero moment point (ZMP), and it is equivalent to the CoP for planar support. ZMP is computed from dynamic equations, whereas CoP is measured using, e.g., a force platform. If during gait planning the computed ZMP is outside the support polygon, the gait is not feasible and must be replanned. Incidentally, CoP is a much older engineering concept, which was originated in the area of fluid mechanics and

aerodynamics. Figure 12 explains the concept of CoP as applied to foot-ground interaction of legged robots.

As shown in Fig. 12, two types of interaction forces act on the foot at the foot/ground interface. They are the normal forces f_{ni} , always directed upwards (Fig. 12, left) and the frictional tangential forces f_{ti} (Fig. 12, middle). CoP, denoted by P , is the point where the resultant $\mathbf{R}_n = \sum \mathbf{f}_{ni}$ acts. With respect to a coordinate origin O , $\mathbf{OP} = \frac{\sum \mathbf{r}_i f_{ni}}{\sum f_{ni}}$, where $\mathbf{r}_i$ is the vector to



Walking Robots, Fig. 10 Three popular humanoid robots, (a) AIST HRP-4. (Image courtesy of AIST), (b) Toyota Partner Robot, and (c) Waseda University Wabian. (Image courtesy Atsuo Takanishi)



Walking Robots, Fig. 11 Three small humanoid robots: Aldebaran NAO. (Image courtesy Aldebaran), Fujitsu HOAP2, and Sony QRIO. (Image courtesy Sony)

the point of action of force f_i and f_{ni} is the magnitude of f_{ni} .

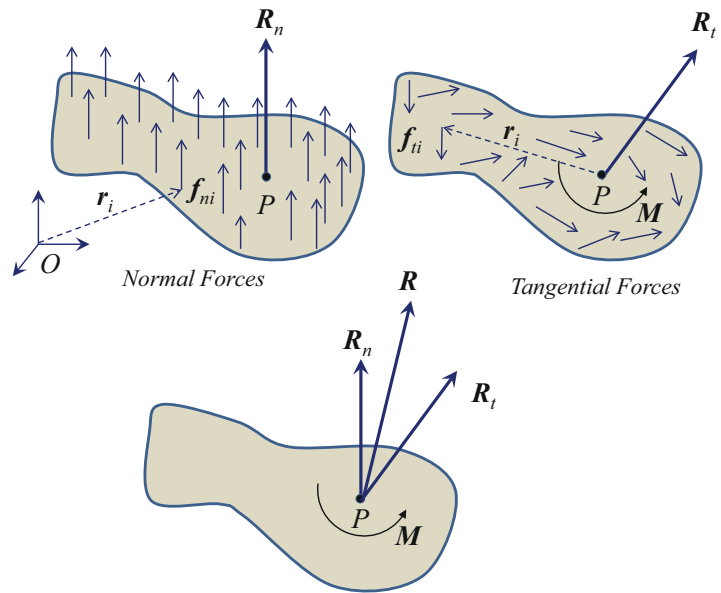
Because of the unilaterality of the foot/ground constraint $f_{ni} \geq 0$, P must lie within the support polygon. The resultant of the tangential forces may be represented at P by a force $\mathbf{R}_t = \sum \mathbf{f}_{ti}$

and a moment $\mathbf{M} = \sum \mathbf{r}_i \times \mathbf{f}_{ti}$ where $\mathbf{r}_i$ is the vector from P to the point of application of $\sum \mathbf{f}_{ti}$.

For a number of years, the word stability was mistakenly used in the walking robot literature to imply robustness against disturbance, loosely meaning that the robot does not fall. Some

Walking Robots, Fig. 12

Definition of center of pressure (CoP), shown for one foot of a humanoid robot. The idea can be extended to any walking robot, and in a general setting, a single footprint is replaced by the support polygon which is the convex hull of all ground contact of the robot



researchers equated the presence of ZMP within the support polygon to mean guaranteed stability. This created unnecessary confusion, because there is also a well-established, precise, but different definition of stability in the control literature. More recently, however, the robotics community has generally stopped associating stability with balance and accepted the fact that the ZMP location indicates the feasibility of gait. Maintaining the ZMP and CoP around the center of the support polygon does improve the robustness by having larger margin to cope with unexpected disturbances.

Arguably, the robots developed by Boston Dynamics such as BigDog (Fig. 6b) and Atlas continue to demonstrate the state of the art in terms of balancing capability and robustness against disturbances and terrain variation. Recently, researchers have started applying deep reinforcement learning techniques to legged robots with the goal of more robust balance control (Lillicrap et al. 2015).

Safety

Safety is a serious concern that is paramount to any application where robots are likely to

coexist in interactive human environments. The power of mobility of walking robots adds to this concern.

Out of a number of possible situations where safety is an issue, one that involves a balance loss and fall is particularly worrisome for walking robots. All walking robots, and in fact all mobile robots, are subjected to this unique “failure” mode. A fall may be caused due to unexpected or excessive external forces, unusual or unknown slipperiness, slope, or profile of the ground, causing the robot to slip, trip, or topple. Fall can also result when the balance controller is partially or fully incapacitated due to an internal failure of the robot involving its sensor or actuator.

Fall can be costly in terms of the damage to the robot and also, depending on the shape and size of the robot, can result in external damage and injury to human.

For humanoid robots fall is a particularly serious issue (Fujiwara et al. 2002). Humanoid robots, similar to humans, have a larger ratio of CoM height to support area size, which makes them more susceptible to fall, in case of a failure. At the same time, due to their higher CoM, a fall of such robots contains generally higher kinetic energy which is able to cause higher damage and injury.

Summary

Walking robots represent an important class of autonomous machines which can find application in the general area of service robotics. The power of mobility makes these robots uniquely capable of serving in niche need areas such as plant maintenance and security, disaster response, personal companion, and so on. Humanoid walking robots have attracted strong popular fascination, and this has fueled their rapid development. At present it appears that defence-related applications are the most likely to experience practical use of walking robots.

Walking robots possess interesting and complex kinematics and dynamics. Control of such machines, especially with regard to balancing, motion planning, and reactive behavior are rich research areas that are challenging and demand special skill sets.

Cross-References

- [Disaster Response Robot](#)
- [Redundant Robots](#)
- [Robot Motion Control](#)
- [Robot Teleoperation](#)
- [Underactuated Robots](#)

Recommended Reading

Out of the references listed below, (Vukobratović and Juričić 1969) is the earliest paper dealing with bipedal robot balance, and it introduces the concept of ZMP. A very good recent overview of legged robots can be found in Kajita and Espiau (2008). Also of interest is the foundational paper on passive bipedal gait is McGeer (1990).

Bibliography

- Asimov I (1950) *I robot*. Gnome Press, New York
- Bares JE, Wettergreen DS (1999) Dante II: technical description, results, and lessons learned. *Int J Robot Res* 18(7):621–649
- Chevallereau C, Westervelt ER, Grizzle JW (2005) Asymptotically stable running for a five-link, four-actuator, planar bipedal robot. *Int J Robot Res* 24(6):431–464
- Falconer J (2013) Nao robot goes to school to help kids with autism. *IEEE Spect*
- Fujiwara K, Kanehiro F, Kajita S, Kaneko K, Yokoi K, Hirukawa H (2002) UKEMI: falling motion control to minimize damage to biped humanoid robot. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 2521–2526
- Gouaillier D, Hugel V, Blazevic P, Kilner C, Monceaux J, Lafourcade P, Marnier B, Serre J, Maisonnier B (2009) Mechatronic design of NAO humanoid. In: *IEEE international conference on robotics and automation (ICRA)*, pp 2124–2129
- Hamilton E (1940) *Captain future: the wizard of science*. Thrilling Publications, New York
- International Federation of Robotics (IFR) Press Release (2012). <http://www.ifr.org/news/ifr-press-release/50-years-industrial-robots-410/>
- Ishida T, Kuroki Y, Takahashi T (2004) Analysis of motions of a small biped entertainment robot. *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 142–147
- Kajita S, Espiau B (2008) Legged robots. In: B Siciliano, O Khatib (eds) *Springer handbook of robotics*. Springer, Berlin, pp 361–389
- Lillicrap TP, et al (2015) Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*
- Kim J, Alspach A, Yamane K (2017) Snapbot: a reconfigurable legged robot. In: *Proceedings of 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 5861–5867
- Kosuge K (2010) Dance partner robot: an engineering approach to human-robot interaction. In: *5th ACM/IEEE international conference on human-robot interaction (HRI)*, Osaka
- Lewinger WA, Branicky MS, Quinn RD (2005) Insect-inspired, actively compliant hexapod capable of object manipulation. In: *Proceedings of CLAWAR 2005 8th international conference on climbing and walking robots*
- Malone R (2004) *Ultimate robot*. DK Publishing, New York
- McGeer T (1990) Passive dynamic walking. *Int J Robot Res* 9(2):62–82
- Raibert M (1989) Legged robots. In: Brady M (ed) *Robotics science. System development foundation benchmark series*. The MIT Press, Cambridge
- Raibert M, Blankespoor K, Nelson G, Playter R and the BigDog Team (2008) BigDog, the rough-terrain quadruped robot. In: *Proceedings of the 17th IFAC world congress, Seoul*, pp 10822–10825
- Robins B, Dautenhahn K, Dickerson P (2012) Embodiment and cognitive learning – can a humanoid robot help children with autism to learn about tactile social behaviour? *Social robotics lecture notes in com-*

- puter science, vol 7621. Springer, Berlin/Heidelberg, pp 66–75
- Sardain P, Bessonnet G (2004) Forces acting on a biped robot. Center of pressure-zero moment point. *IEEE Trans Syst Man Cybern* 34:630–637
- Vukobratović M, Juričić D (1969) Contribution to the synthesis of biped gait. *IEEE Trans Bio-Medical Eng* 16(1):1–6
- Waldron KJ, McGhee RB (1986) The adaptive suspension vehicle. *IEEE Control Syst Mag* 6(6):7–12
- Yamasaki F, Nakagawa Y (2006) Education using small humanoid robot. In: *Proceedings of 3rd international symposium on autonomous minirobots for research and edutainment (AMiRE 2005)*, pp 248–253

Wheeled Robots

Giuseppe Oriolo

Sapienza Università di Roma, Roma, Italy

Abstract

The use of mobile robots in applications is steadily increasing, both in the industrial and the service domains. Most mobile robots achieve locomotion using wheels. As a consequence, they are subject to differential constraints that are nonholonomic, i.e., non-integrable. This article reviews the kinematic models of wheeled robots arising from these constraints and discusses their fundamental properties and limitations from a control viewpoint. An overview of the main approaches for trajectory planning and feedback motion control is provided.

Keywords

Wheeled robots · Nonholonomic constraints · Differential flatness · Nonlinear controllability · Smooth stabilizability

Introduction

Although all robots are by definition capable of movement, the expression *mobile robots* is mainly used to indicate robots that can displace their own base by means of some locomotion

mechanism. Most often, this consists of a set of wheels. The main advantage of mobile robots over fixed-base manipulators is their virtually unlimited workspace. As a consequence, such robots are increasingly being used in both the industrial and the service domains, and in general in all applications which require increased capabilities of autonomous motion.

From a mechanical viewpoint, a *wheeled robot* essentially consists of a rigid body (*base*, or *chassis*) equipped with a system of wheels. This basic arrangement may be complicated, for example, by attaching to the base one or more trailers, or by mounting a manipulator on the base (*mobile manipulator*).

Any wheeled vehicle is subject to kinematic constraints that in general reduce its local mobility, while leaving intact the possibility of reaching arbitrary configurations by appropriate maneuvers. For example, any driver knows by experience that, while it is impossible to move instantaneously a car in the direction orthogonal to its heading, it is still possible to park it anywhere, at least in the absence of obstacles. This peculiar feature makes wheeled mobile robots very challenging from the control viewpoint, and in fact some recent developments in nonlinear control were motivated and inspired by the study of these systems.

Here, we will consider only mobile robots that are equipped with conventional wheels, either orientable or fixed (as the front or rear wheels of a car, respectively). Omnidirectional mobile robots realized using, e.g., Mecanum wheels are not covered in this article. Indeed, the local mobility of these vehicles is unrestricted, and therefore no special control treatment is necessary.

The most popular wheel arrangement for mobile robots is the *differential drive*, in which two fixed wheels whose axes of rotation coincide are controlled by separate actuators (see Fig. 1). One or more passive (*caster*) wheels are usually added for statical balance. This wheeled robot is the most agile, as it can rotate on the spot by applying equal and opposite angular speeds to the wheels. A kinematically equivalent arrangement is the *synchro drive*, in which three orientable wheels are synchronously driven by two motors



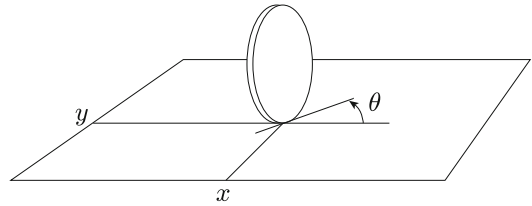
Wheeled Robots, Fig. 1 The Roomba (here, model 880) series of vacuum cleaning robots produced by iRobot features differential-drive kinematics

through mechanical coupling; the first motor provides traction, whereas the second controls the common orientation of the wheels.

Other possible kinematic structures include the *tricycle* (one steering and two fixed wheels) and the *car-like* (two steering and two fixed wheels). Vehicles of this type are however less common in robotics, due partly to their reduced maneuverability (they have a nonzero turning radius) and partly to their increased mechanical complexity. For example, both these vehicles require a specific device (*differential*) for distributing traction torque to the driving wheels.

Modeling

The starting point for modeling wheeled mobile robots is the single wheel. This may be represented as an upright disk rolling on the ground. Its configuration is described by three generalized coordinates: the Cartesian coordinates (x, y) of



Wheeled Robots, Fig. 2 Generalized coordinates for a single wheel

the contact point with the ground, measured in a fixed reference frame, and the orientation θ of the disk plane with respect to the x axis (see Fig. 2). The configuration vector is therefore $q = (x \ y \ \theta)^T$. The *pure rolling* constraint, expressed as

$$(\sin \theta - \cos \theta) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = 0, \quad (1)$$

entails that in the absence of slipping, the velocity of the contact point has zero component in the direction orthogonal to the wheel plane. The angular speed of the wheel around the vertical axis is instead unconstrained. The kinematic constraint (1) is *nonholonomic*, i.e., it cannot be integrated to a geometric constraint; this may be easily shown using Frobenius Theorem, a well-known differential geometry result on integrability of differential forms. An important consequence of its being nonholonomic is that constraint (1) does not imply a loss of accessibility in the configuration space of the wheel.

In a single-body vehicle equipped with multiple wheels, the n -dimensional configuration vector q consists of the Cartesian coordinates of a representative point on the robot, the orientation of all independently orientable wheels, plus the orientation of the body if there are fixed wheels. By writing one pure rolling constraint like (1) for each independent wheel, either orientable or fixed, and expressing it in the chosen generalized coordinates, one obtains a set of k constraints in the form

$$A^T(q) \dot{q} = 0. \quad (2)$$

Kinematic constraints of this form (i.e., linear in the generalized velocities) are called *Pfaffian*. In

wheeled mobile robots, Pfaffian constraints are in general completely nonholonomic.

The k Pfaffian constraints (2) reduce the number of degrees of freedom (i.e., independent instantaneous motions) of the robot to $m = n - k$. In particular, at each configuration q the generalized velocities $\dot{q}$ must belong to the m -dimensional null space of matrix $A^T(q)$:

$$\dot{q} = \sum_{j=1}^m g_j(q) u_j = G(q)u, \quad (3)$$

where vector fields $g_1(q), \dots, g_m(q)$ are a basis of $\mathcal{N}(A^T(q))$ and $u = (u_1 \dots u_m)^T$ is a coefficient vector. Kinematically admissible trajectories are the solutions of (3), which is called *kinematic model* of the wheeled mobile robot. This model can be seen as a nonlinear dynamical system, with q as state and u as input. In particular, system (3) is driftless and has more state variables than control inputs.

For example, consider the *unicycle*, a somewhat ideal mobile robot equipped with a single, orientable wheel. The configuration vector for this robot is $q = (x \ y \ \theta)^T$, the same as the single wheel, and the vehicle is subject to the rolling constraint (1). One possible kinematic model of the form (3) for the unicycle is then

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega, \quad (4)$$

where $v = \pm\sqrt{\dot{x}^2 + \dot{y}^2}$ and $\omega = \dot{\theta}$ represent, respectively, the driving and steering velocity of the wheel. Both the differential-drive and the synchro-drive robots are kinematically equivalent to the unicycle, i.e., their kinematic model can be put in the form (3) by properly defining q and u .

Similarly to what is done for robot manipulators, dynamic models of wheeled mobile robots may be derived following the Euler-Lagrange method. The main difference is the presence of the nonholonomic Pfaffian constraints, which give rise to reaction forces expressed via Lagrange multipliers (Neimark and Fufaev 1972).

Structural Properties

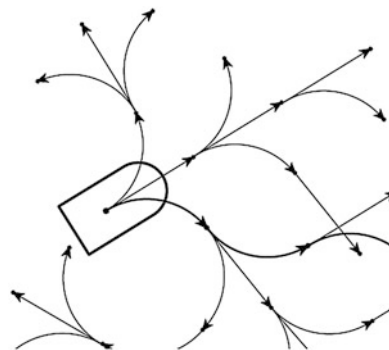
The nonholonomic nature of wheeled mobile robots has precise consequences in terms of structural properties of the kinematic model (3).

The first, and most important, is that in spite of the reduced number of degrees of freedom, wheeled robots are *controllable* in their configuration space; i.e., given two arbitrary configurations, there always exists a kinematically admissible trajectory (with the associated velocity inputs) that transfers the robot from one to the other (Fig. 3). Since the kinematic model (3) is driftless, this fact can be verified using a well-known result (Chow Theorem), according to which the system is controllable if and only if the accessibility rank condition holds:

$$\dim \bar{\Delta} = n, \quad (5)$$

where $\bar{\Delta}$ denotes the involutive closure of distribution $\Delta = \{g_1, \dots, g_m\}$ under the Lie bracket operation. In turn, this is guaranteed to be true in view of the nonholonomy of constraints (2). For example, since the Lie bracket of the two input vector fields in (4) is always linearly independent from them, the kinematic model of the unicycle is controllable.

However, the controllability of wheeled mobile robots is intrinsically nonlinear. In fact, the linear approximation of (3) at any configuration turns out to be uncontrollable



Wheeled Robots, Fig. 3 In spite of its restricted local mobility, a nonholonomic wheeled robot can reach any point in its configuration space

due to the reduced number of inputs. This indicates that no linear feedback can stabilize the system at a given configuration. The situation is actually worse: for nonholonomic robots, there exist no continuous time-invariant feedback law that achieves stabilization at a point. This can be established on the basis of a celebrated result on smooth stabilizability of control systems (Brockett 1983). This negative result does not apply to time-varying stabilizing controllers, which may thus be continuous in q .

Another (related) complication of wheeled mobile robots is that in general they do not admit *universal* controllers, i.e., feedback control laws that can asymptotically stabilize *arbitrary* state trajectories, either persistent or not (Lizárraga 2004). Therefore, for these systems one needs in principle different controllers for solving trajectory tracking and point regulation problems.

All the above limitations of nonholonomic systems are established with reference to the kinematic model, but of course they are passed on to dynamic models. Altogether, they contribute to making wheeled mobile robots much more difficult to control than, for example, robotic manipulators, which are linearly controllable, smoothly stabilizable, and admit universal controllers.

Trajectory Planning

Trajectory planning for wheeled robots is a non-trivial problem, because not all trajectories are feasible – once again, a consequence of non-holonomy. This leads to the necessity of *maneuvering*, i.e., performing certain specific movements, in order to execute transfer motions.

Most kinematic models of wheeled mobile robots exhibit a property known as *differential flatness* (Fliess et al. 1995): namely, there exists a set of outputs z , called *flat* outputs, such that the state q and the control inputs u can be expressed algebraically as a function of z and its time derivatives up to a certain order σ :

$$q = \varphi(z, \dot{z}, \ddot{z}, \dots, z^{(\sigma)}) \quad (6)$$

$$u = \gamma(z, \dot{z}, \ddot{z}, \dots, z^{(\sigma)}). \quad (7)$$

As a consequence, the state trajectory $q(t)$ and control history $u(t)$ associated to a given output trajectory $z(t)$ are uniquely determined. For example, the unicycle admits $z = (x \ y)^T$ as flat outputs. In fact, once a Cartesian trajectory is assigned for the contact point with the ground, the wheel orientation $\theta(t)$ is constrained to be tangent to the trajectory; the associated control input v and ω are then uniquely and algebraically computable from $q(t)$.

Differential flatness is particularly useful for planning. In fact, assume that we want to transfer a wheeled mobile robot from an initial configuration q_i to a final configuration q_f . One then computes the corresponding values z_i and z_f of the flat outputs, plus the appropriate boundary conditions, and uses any interpolation scheme (e.g., polynomial interpolation) to plan the trajectory of z . The evolution of the generalized coordinates q , together with the associated control inputs u , can then be computed algebraically from (6)–(7). The resulting configuration space trajectory will automatically satisfy the nonholonomic constraints (2).

Another approach to nonholonomic trajectory planning relies on the possibility of putting the equations of most wheeled robots into a canonical format known as a 2-input *chained form*

$$\begin{aligned} \dot{z}_1 &= w_1 \\ \dot{z}_2 &= w_2 \\ \dot{z}_3 &= z_2 w_1 \\ &\vdots \\ \dot{z}_n &= z_{n-1} w_1 \end{aligned} \quad (8)$$

by means of a feedback transformation, i.e., a change of coordinates $z = \alpha(q)$ coupled with an input transformation $w = \beta(q)u$. In particular, this is always possible for kinematic models like (3) when $n \leq 4$ and $m = 2$ (e.g., unicycle or car-like robots). Once the system is cast in the form (8), one may use sinusoidal open-loop controls at integrally related frequencies to drive all variables sequentially to their final values (Murray and Sastry 1993). This approach

is particularly interesting from a theoretical viewpoint because such control maneuvers achieve motion in the direction of the Lie brackets of the input vector fields.

Note that differential flatness and chained-form transformability are equivalent properties for 2-input nonholonomic mobile robots.

Feedback Control

The motion control problem for wheeled mobile robots is generally formulated with reference to the kinematic model (3). For example, in the case of the unicycle (4) this means that the control inputs are directly v and ω , the driving and steering velocities. There are essentially two reasons for taking this simplifying assumption.

First, the kinematic model (3) fully captures the essential nonlinearity of single-body wheeled robots, which stems from their nonholonomic nature. This is another fundamental difference with respect to the case of robotic manipulators, in which the main source of nonlinearity is inertial coupling among multiple bodies. Second, in mobile robots it is typically impossible to command directly the wheel torques, because there are low-level wheel control loops integrated in the hardware or software architecture. These loops will accept as input a reference value for the wheel angular speed, which is then reproduced as accurately as possible by standard regulation actions (e.g., PID controllers). In this situation, the actual inputs available for high-level control are precisely these reference velocities.

Two basic control problems can be considered:

- *Trajectory tracking*: the robot must asymptotically track a desired Cartesian trajectory $(x_d(t), y_d(t))$.
- *Point stabilization*: the robot must asymptotically reach a desired configuration q_d .

From a practical point of view, the most relevant of these problems is arguably the first. This

is because mobile robots must be able to operate in unstructured workspaces that invariably contain obstacles. Clearly, forcing the robot to move along (or close to) a trajectory planned in advance reduces considerably the risk of collisions. The point stabilization problem, however, is more difficult and therefore particularly interesting from a scientific perspective. In a certain sense, the relative difficulty of the two problems is reminiscent of human car driving: learning to drive a car along a road is relatively easy, whereas parking poses a greater challenge.

Trajectory Tracking

Several methods are available to drive a wheeled mobile robot in feedback along a desired trajectory. A straightforward possibility is to compute first the linear approximation of the system along the desired trajectory (which, unlike the approximation at a configuration, results to be controllable) and then stabilize it using linear feedback. Only local convergence, however, can be guaranteed with this approach. For the kinematic model of the unicycle, global asymptotic stability may be achieved by suitably morphing the linear control law into a nonlinear one (Canudas de Wit et al. 1993).

In robotics, a popular approach for trajectory tracking is input-output linearization via static feedback. In the case of a unicycle, consider as output the Cartesian coordinates of a point B located ahead of the wheel, at a distance b from the contact point with the ground. The linear mapping between the time derivatives of these coordinates and the velocity control inputs turns out to be invertible provided that b is nonzero; under this assumption, it is therefore possible to perform an input transformation via feedback that converts the unicycle to the parallel of two simple integrators, which can be globally stabilized with a simple proportional controller (plus feedforward). This simple approach works reasonably well. However, if one tries to improve tracking accuracy by reducing b (so as to bring B closer to the ground contact point) the control effort quickly increases.

Trajectory tracking with $b = 0$ (i.e., for the actual contact point on the ground)

can be achieved using dynamic feedback linearization (Oriolo et al. 2002). In particular, this method provides a one-dimensional dynamic compensator that transforms the unicycle into a parallel of two double integrators, which is then globally stabilized with a proportional-derivative controller (plus feedforward). In contrast to static feedback linearization, no residual zero dynamics is present in the transformed system. However, the dynamic compensator has a singularity when the unicycle driving velocity is zero. This is expected, because otherwise the tracking controller would represent a universal controller. Note that dynamic feedback linearizability using the x, y outputs is related to them being flat – the two properties are equivalent.

Point Stabilization

The impossibility of stabilizing a nonholonomic mobile robot using continuous pure-state feedback has generated two main directions of research to solve the problem:

- *discontinuous* feedback, i.e., time-invariant control laws $u = \gamma(q)$, where γ is discontinuous precisely at the configuration that one seeks to stabilize;
- *time-varying* feedback, in the form $u = \gamma(q, t)$ where γ may or may not be continuous at the desired configuration.

For the unicycle, a well-known stabilizing controller belonging to the first category was designed by Aicardi et al. (1995) by formulating the problem in polar coordinates centered at the goal and then using a Lyapunov-like analysis to establish asymptotic convergence. The controller, once rewritten in original coordinates, turns out to be discontinuous at the goal (not surprisingly). Although this rules out proper stability in the sense of Lyapunov, this controller is effective in that it produces rather natural approach trajectories to the goal.

Continuous time-varying stabilizers in the sense of Lyapunov exist (Samson 1993) but have mainly theoretical interest due to their provably slow (polynomial) rate of convergence;

this is a direct consequence of the fact that the linear approximation of the system is not controllable. A more effective approach is to give up (Lipschitz-) continuity at the desired configuration. As shown by M'Closkey and Murray (1997) and Morin and Samson (2000), this allows to design control laws that guarantee a modified form of exponential convergence to the goal.

Most of the aforementioned control designs – both for trajectory tracking and point stabilization – were first developed with reference to the unicycle robot but can be carried out on chained forms, thereby providing an effective extension to other kinematic models, e.g., the car-like robot.

Summary and Future Directions

Wheeled mobile robots are increasingly present in applications. Over the last two decades, significant results have been reached in terms of modeling, planning, and control of these systems, and the field is now considered to be well-established, at least from an application point of view. Nevertheless, a number of research directions are still open, including the following:

- *Planning and control for non-flat systems:* Relatively harmless wheeled robots (such as a unicycle towing more than one off-hooked trailer) are not flat.
- *Robustness:* The design of controllers that can effectively withstand disturbances and model perturbations is still an active area of research.
- *Localization:* Feedback control requires accurate measurements of the configuration variables, which in mobile robots cannot be reliably reconstructed from on-board sensors (odometric data). Integration of exteroceptive sensing is essential to this end.
- *Vision-based control:* As an alternative to localization-based methods, the feedback loop may be closed directly in the image plane, with significant advantages in terms of simplicity and robustness.

- *Multi-robot systems*: The problem is to control the motion of multiple mobile robots in order to perform a cooperative motion task, e.g., formation control.

Cross-References

- [Differential Geometric Methods in Nonlinear Control](#)
- [Feedback Linearization of Nonlinear Systems](#)
- [Feedback Stabilization of Nonlinear Systems](#)
- [Lie Algebraic Methods in Nonlinear Control](#)
- [Vehicle Dynamics Control](#)

Recommended Reading

For background material on nonlinear controllability, including the necessary concepts of differential geometry, see Sastry (2005). General introductions to mobile robots can be found in Siegwart and Nourbakhsh (2004), Choset et al. (2005), Morin and Samson (2008), Siciliano et al. (2009), and Tzafestas (2013). A classification of wheeled mobile robots based on the number, placement and type of wheels was proposed by Bastin et al. (1996). A detailed extension of some of the planning and control techniques reviewed in this article to the case of car-like kinematics is given in De Luca et al. (1998). A detailed treatment of control problems for mobile robots can be found in Samson et al. (2016). A framework for the stabilization of non-flat nonholonomic robots was presented by Oriolo and Vendittelli (2005). Recent work aimed at designing practical universal controllers was carried out by Morin and Samson (2009).

Bibliography

- Aicardi M, Casalino G, Bicchi A, Balestrino A (1995) Closed loop steering of unicycle-like vehicles via Lyapunov techniques. *IEEE Robot Autom Mag* 2(1):27–35
- Bastin G, Campion G, D’Andréa-Novel B (1996) Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Trans Robot Autom* 12:47–62
- Brockett RW (1983) Asymptotic stability and feedback stabilization. In Brockett RW, Millman RS, Sussmann HJ (eds) *Differential geometric control theory*. Birkhauser, Boston
- Canudas de Wit C, Khenouf H, Samson C, Sjørdalen OJ (1993) Nonlinear control design for mobile robots. In: Zheng YF (ed) *Recent trends in mobile robots*. World Scientific Publisher, Singapore, pp 121–156
- Choset H, Lynch KM, Hutchinson S, Kantor G, Burgard W, Kavraki LE, Thrun S (2005) *Principles of robot motion: theory, algorithms, and implementations*. MIT Press, Cambridge
- De Luca A, Oriolo G, Samson C (1998) Feedback control of a nonholonomic car-like robot. In: Laumond J-P (ed) *Robot motion planning and control*. Springer, London, pp 171–253
- Fliess M, Lévine J, Martin P, Rouchon P (1995) Flatness and defect of nonlinear systems: introductory theory and examples. *Int J Control* 61:1327–1361
- Lizárraga DA (2004) Obstructions to the existence of universal stabilizers for smooth control systems. *Math Control Signals Syst* 16:255–277
- M’Closkey RT, Murray RM (1997) Exponential stabilization of driftless nonlinear control systems using homogeneous feedback. *IEEE Trans Autom Control* 42:614–628
- Morin P, Samson C (2000) Control of non-linear chained systems: from the Routh-Hurwitz stability criterion to time-varying exponential stabilizers. *IEEE Trans Autom Control* 45:141–146
- Morin P, Samson C (2008) Motion control of wheeled mobile robots. In: Khatib O, Siciliano B (eds) *Handbook of robotics*. Springer, Berlin/Heidelberg, pp 799–826
- Morin P, Samson C (2009) Control of nonholonomic mobile robots based on the transverse function approach. *IEEE Trans Robot* 25:1058–1073
- Murray RM, Sastry SS (1993) Nonholonomic motion planning: steering using sinusoids. *IEEE Trans Autom Control* 38:700–716
- Neimark JJ, Fufaev FA (1972) *Dynamics of non-holonomic systems*. American Mathematical Society, Providence
- Oriolo G, Vendittelli M (2005) A framework for the stabilization of general nonholonomic systems with an application to the plate-ball mechanism. *IEEE Trans Robot* 21:162–175
- Oriolo G, De Luca A, Vendittelli M (2002) WMR control via dynamic feedback linearization: design, implementation and experimental validation. *IEEE Trans Control Syst Technol* 10:835–852
- Samson C (1993) Time-varying feedback stabilization of car-like wheeled mobile robots. *Int J Robot Res* 12(1):55–64
- Samson C, Morin P, Lenain R (2016) Modeling and control of wheeled mobile robots. In Siciliano B, Khatib O (eds.) *Springer handbook of robotics*, pp 1235–1266. Springer, Berlin
- Sastry S (2005) *Nonlinear systems: analysis, stability and control*. Springer, New York

- Siciliano B, Sciavicco L, Villani L, Oriolo G (2009) Robotics: modelling, planning and control. Springer, London
- Siegwart R, Nourbakhsh IR (2004) Introduction to autonomous mobile robots. MIT Press, Cambridge.
- Tzafestas S (2013) Introduction to mobile robot control. Elsevier, Amsterdam

Wide-Area Control of Power Systems

Aranya Chakraborty
Electrical and Computer Engineering
Department, North Carolina State University,
Raleigh, NC, USA

Abstract

This chapter discusses the problem of *wide-area control* (WAC) for large-scale power transmission systems. The discussion covers two main applications, namely, wide-area oscillation damping control and wide-area voltage control. Scalable control methods are reported for both applications, highlighting important challenges in design, implementation, and computational complexity. This is followed by two newly evolving control paradigms, namely, sparsity-promoting WAC and online learning-based WAC. The first design reduces the cost and complexity of data communication by imposing structural constraints on the controller. The second design transitions conventional model-based WAC to a data-driven approach by employing system identification and real-time control actions. The chapter ends with a list of open problems and future research directions in this research area including the need for efficient cyber-physical architectures, cyber-security, machine learning, and game theory for WAC.

Keywords

Power system dynamics · Oscillations · Damping control · Global control · System-wide communication · Model reduction · System identification

Introduction

The US Northeast blackout of 2003, followed by the timely emergence of sophisticated GPS-synchronized digital instrumentation technologies such as Wide-Area Measurement Systems (WAMS) using Phasor Measurement Units (PMUs), was a landmark event in the history of the North American power system. It led utility owners to understand how the interconnected nature of a grid topology couples the controller performance within their local operating regions with that of the others. It also encouraged them to realize the importance of system-wide or global control for the grid using wide-area measurement feedback (Phadke and Thorp 2008) on top of the existing local control architectures. These global controllers are commonly referred to as *wide-area controls*. Over the past decade, several researchers have investigated wide-area controls using various innovative design approaches including H_∞ control (Chaudhuri et al. 2010a, 2012; Zhang and Vittal 2013), LMIs and conic programming (Jabr et al. 2010), wide-area protection (Zima et al. 2005), model reduction and control inversion (Chakraborty 2012; Xue and Chakraborty 2019), adaptive control (Chow and Ghiocel 2012), LQR-based optimal control (Dorfler et al. 2014; Raoufat et al. 2016), etc., complemented with insightful case studies of controller implementation for various real power systems such as the US west coast grid (Taylor et al. 2005), Hydro Quebec (Kamwa et al. 2001), and the Nordic system. A tutorial on the ongoing practices for wide-area control has recently been presented in Chakraborty and Khargonekar (2013).

Two primary challenges for wide-area control lie in computation and communication. Given the extreme-scale size and complexity of any realistic power system with thousands of generators and loads, and hundreds of thousands of transmission lines, designing a wide-area controller that encompasses all aspects of such a large dynamic complex network easily becomes challenging. Furthermore, once the design is done, the next step is to implement this controller in the real grid, which requires a reliable wide-area

communication network that must be capable of carrying massive volumes of PMU data from one part of the grid to another with minimal delays and disruptions. Issues on data privacy and cyber-security also need attention. In recent years, increase in renewable generation has made the situation even more taxing as on one hand the loss of inertia and damping necessitates WAC to a greater extent while on the other hand the uncertainty of the renewable models demand more robustness of WAC than usual. In the following sections, we provide a brief overview of how several of these challenges can be resolved through advanced control designs.

Wide-Area Oscillation Damping Control

Wide-area control is especially suited to the problem of power oscillation damping. Conventional damping controllers such as PSS often exhibit poor controllability on low frequency oscillations, also known as inter-area oscillations, due to the localized effect of their control actions. Therefore, a system-wide wide-area control action is needed. Several papers have been written on wide-area oscillation damping, but scalability of these designs and complexity of their implementation stand as two major roadblocks. Conventional optimal controllers such as the linear-quadratic regulator (LQR) have been shown to be plausible candidates, but they require $\mathcal{O}(n^3)$ computational complexity (n for a power system can be in the order of thousands) and usually demand all-to-all communication between every generator for implementing the feedback.

One way to resolve this problem is to make use of dimensionality reduction, and design the WAC using an approach called *control inversion*, which was recently proposed in Xue and Chakraborty (2019). The method involves three steps. The first step is to project the full-scale power system model with n generators to a lower-dimensional state-space by aggregating the generators into r groups ($r \leq n$). This projection is defined by

a clustering set $\mathcal{I}$. The second step is to design an LQR controller in the lower-dimensional state-space using the aggregated model. The final step is to project this controller back to the original dimension using an inverse projection. The overall complexity of this design scales only with r instead of n . Moreover, due to the structure of the projections, the controller naturally results in a simple two-layer hierarchical implementation strategy. The main problem, therefore, reduces to finding the set $\mathcal{I}$ such that the closed-loop performance of the proposed WAC matches that of an optimal LQR controller, which, for example, can be considered as a reference controller. The design in Xue and Chakraborty (2019) proposed two relaxations using which this model-matching problem reduces to designing $\mathcal{I}$ from a quadratic optimization problem that can be constructed and solved in a numerically inexpensive way. For example, consider the performance variables for the damping control as the phase angle differences between any chosen pairs of generators and the frequency deviations of all generators. Using this definition, consider two closed-loop transfer function matrices (TFMs) from a disturbance input d to the performance output y :

- With optimal controller: $g(s) = C(sI - A + BK)^{-1}B_d$.
- With the desired suboptimal controller $\hat{g}(s) = C(sI - A + B\hat{K})^{-1}B_d$.

(A , B , C) is the small-signal state-space model of the power system (e.g., a flux-decay model), B_d is the disturbance input matrix, and K and $\hat{K}$ are optimal and sub-optimal state-feedback controllers. The disturbance d can be a design metric for evaluating the dynamic response of the electro-mechanical swing states. One may consider the worst-case scenario where this disturbance sufficiently excites the responses of all states and, therefore, assume (A , B_d) to be controllable. The WAC damping control problem of interest can then be stated as follows:

Given TFMs $g(s)$ and $\hat{g}(s)$, find a sub-optimal LQR controller $u = -K\hat{x}$ that solves the WAC model-matching problem

$$\min_{\hat{K}} \|g(s) - \hat{g}(s)\|_{H_2, \bar{\omega}} \quad (1)$$

where the norm $\|\cdot\|_{H_2, \bar{\omega}}$ is defined as $\|h(s)\|_{H_2, \bar{\omega}} = \sqrt{\frac{1}{2\pi} \int_{-\bar{\omega}}^{\bar{\omega}} \text{trace}(h^*(jw)h(jw))dw}$ for any stable TFM $h(s)$, and $[0, \bar{\omega}]$ indicates the frequency range of inter-area oscillations in the power system model. The controller $\hat{K}$ should satisfy the following three constraints:

1. *Consensus* – since the total amount of power in the network remains conserved, the power system model exhibits a consensus property which manifests as a zero eigenvalue in A . The same property must also be true in closed-loop, i.e., $A - B\hat{K}$ must have a zero eigenvalue (often referred to as the DC mode).
2. *Computation* – The design complexity of $\hat{K}$ should be no more than $\mathcal{O}(n^3)$.
3. *Implementation* – The structure of $\hat{K}$ is desired to produce a much simpler communication topology between the generators.

The minimization problem (1) under these three constraints can be solved using the control inversion method. For details of the design please see Xue and Chakraborty (2019).

Wide-Area Voltage Control

Wide-area voltage stability is another significant concern for power systems operating with remote generation and limited reactive power support along congested power transfer paths. Traditional methods of investigating voltage stability span two extremes. At one end of the spectrum, detailed models consisting of thousands of buses are used to calculate voltage stability margins for many $N-1$ contingency cases. At the other end of the spectrum, much simpler models of radial lines connected to a load center, also referred to as the voltage instability predictor (VIP) model, has been proposed (Julian et al. 2000). A three-level voltage control architecture involving slow, mid-range, and fast voltage dynamics for low-, medium-, and high-voltage grids has been provided in Ilic and Zaborsky

(2000) and Chow and Sanchez-Gasca (2020). The realm of possible approaches for real-time voltage stability analysis, especially in the set of fast and mid-range dynamics, changes substantially with the availability of PMUs. For example, the VIP model can be made much more accurate by using PMU-based model reduction techniques. From a controls perspective, the primary application of Synchrophasors for mitigating voltage oscillations in the mid-range voltage dynamics across wide areas lies in appropriate control of voltage regulating devices such as Static VAR Compensators (SVC).

For example, consider a bus j in a power system network, and assume the instantaneous voltage phasor at this bus at any time t to be $\tilde{V}_j(t) = V_j(t) \angle \theta_j(t)$. Consider that a voltage controller in the form of a Static VAR Compensator (SVC) or any other shunt FACTS device is installed at this bus to regulate V_j to a constant setpoint V_j^* . The dynamic model of this controller can be written as

$$\tau \dot{B}_j(t) = -B_j(t) + u_j(t) \quad (2)$$

where $B_j = \frac{1}{x_j}$ with x_j being the capacitive reactance which shorts Bus j to the ground voltage and $u_j(t)$ is a designable input. Traditionally, a proportional controller of the form $u_j = k_j(V_j^* - V_j)$ is applied with $k_j > 0$ being the controller gain. Consider any other Bus i in the network with voltage phasor $\tilde{V}_i(t) = V_i(t) \angle \theta_i(t)$. Let the equilibrium values for $V_i(t)$ and $\theta_i(t)$ be V_{i0} and θ_{i0} , respectively. Evaluate the sensitivity of the voltage deviation $\Delta V_i(t) = V_i(t) - V_{i0}(t)$, and angle deviation $\Delta \theta_i(t) = \theta_i(t) - \theta_{i0}(t)$ with respect to the control gain k_j for any given i . One pertinent research question then is – how do these sensitivities depend on the network topology and parameters as captured, for instance, by the network admittance matrix? Analytical expressions for such sensitivity functions were derived in Chakraborty et al. (2011) for a two-area radial system with intermediate voltage control. An important observation made from that result is that for a two area system as the control gain of the SVC is increased for better voltage regulation, the voltage deviations at the

other points on the tie-line increase from their pre-disturbance value. In other words, the voltage fluctuations at these spatial locations become more dominant due to the control action taken by the SVC. This naturally raises the question of how such local voltage control at one specific bus may impact other neighboring buses not just for a two-area system but for larger systems with multiple controllers and more arbitrary topologies. A relevant control problem can then be formulated as follows:

Consider m SVCs installed at selected buses in the power system. Each SVC is defined by a model of the form (2) with corresponding control input u_i for the i th SVC. Denote $u = [u]_{i=1}^m$. Let the measured output from PMUs be denoted as $y(t)$. Define a performance metric $\mathcal{J}$ that reflects the deviation of the voltage at any bus j from its setpoint for $j = 1, 2, \dots, n$. Design an output-feedback dynamic controller $u(t) = F(y(t))$ that minimizes $\mathcal{J}$.

However, there are likely to be fundamental constraints that impose lower bounds on the best achievable closed-loop performance for these wide-area control problems. Quantification of such lower bounds can be an interesting direction that could yield fundamental new insights into network-level voltage control problems.

New Control Paradigms

Sparse and Distributed Wide-Area Control

As mentioned earlier, WAC for power oscillation damping has been formulated in terms of the LQR optimal control problem on several occasions (Xue and Chakraborty 2019; Dorfler et al. 2014). The control signal is added as supplemental control input to the AVR of the synchronous generator resulting in an additional damping torque actuated through the PSS, especially on the low-frequency oscillations. The structure of the controller is $u = -Kx$, $K \in \mathcal{S}$, where x is the state vector of the generators and $\mathcal{S}$ represents admissible controllers encapsulating the distributed nature of WAC. Once K is designed, the wide-area control is implemented by setting up a communication network between

the generators. For simplicity, the communication is assumed to be ideal, that is, it does not have any delays or packet losses. Works on WAC under communication delays or packet losses can be found in papers such as Wu et al. (2002) and Chaudhuri et al. (2004). The generator state vector x is assumed to be available (for state estimation using PMU measurements and decentralized Kalman filters, please see Singh and Pal 2014). Owing to the wide availability of PMU data, utilities these days have reasonably good models for their operating regions. The independent system operators also have increasingly accurate power system models. For a robust implementation of WAC, these models should be updated every few hours using fresh PMU data as the operating point changes, followed by an update in K .

If K is a dense matrix then every generator would need to communicate with every other generator for implementing the WAC, leading to a very expensive and complex communication system. This can be avoided by choosing $\mathcal{S}$ to promote sparsity in K_G without sacrificing closed-loop performance much. The usual philosophy for constructing $\mathcal{S}$ is as follows. For a natural number $L \leq n$, where n is the number of generators, consider a set of groups $\{\mathcal{G}_l\}_{l \in \{1, \dots, L\}}$ such that $\bigcup_{l \in \{1, \dots, L\}} \mathcal{G}_l = \{1, \dots, n\}$. Note that the groups are not necessarily disjoint, namely, there may exist pair (l, l') such that $\mathcal{G}_l \cap \mathcal{G}_{l'} \neq \emptyset$. Let K_{ij} denote the (i, j) -block matrix of K . Define $\mathcal{S}$ as

$$\mathcal{S} := \left\{ K : K_{ij} = 0, \text{ there does not exist } l \right. \\ \left. \in \{1, \dots, L\}, \text{ s.t. } i \in \mathcal{G}_l \wedge j \in \mathcal{G}_l \right\}. \quad (3)$$

The problem then is to find K minimizing the LQR cost under the constraint $\mathcal{S}$. Due to the sparse structure of the communication network, the controllers can be implemented in a distributed way instead of using all-to-all communication.

The concept can also be extended to nonlinear WAC using Koopman operator theory (Hernandez-Ortega and Messina 2018) in which case the controller is given as a combination of

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \end{bmatrix} = \underbrace{\begin{bmatrix} \times & \times & \cdots & 0 \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & \ddots & 0 \\ \times & 0 & \cdots & 0 \end{bmatrix}}_{\Omega_{1ord}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} + \underbrace{\begin{bmatrix} \times & \times & \cdots & 0 \\ 0 & 0 & \cdots & \times \\ \vdots & \vdots & \ddots & 0 \\ \times & 0 & \cdots & 0 \end{bmatrix}}_{\Omega_{2ord}} \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ \vdots \\ x_N x_N \end{bmatrix}$$

Wide-Area Control of Power Systems, Fig. 1 Sparse WAC for nonlinear models using Koopman mode analysis

two separate state-feedback control signals, one for the linear model and another for the nonlinear model as shown in Fig. 1. The sparse structure of the two controllers need not be the same. The total number of communication links is decided by the union of the set of the non-zero entries of the two controllers.

Online Learning-Based Control

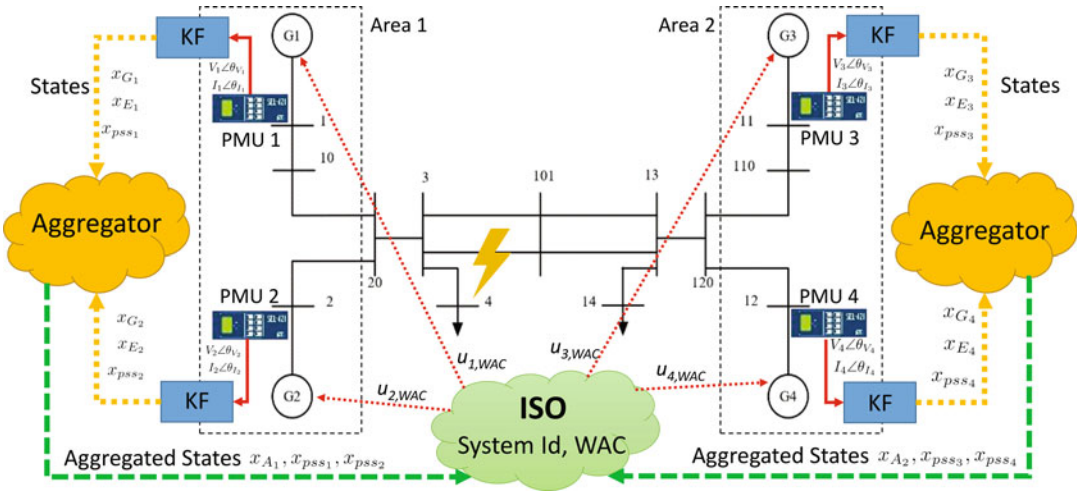
With the growing levels of uncertainty in grid operations such as constantly changing loads, wind and solar penetration, and changes in grid topology, system operators currently are inclining more toward model-free and online control approaches. But the greatest challenge, again, is the continental size of any real-world grid. The problem can be resolved by identifying a severely reduced-dimensional grid model by making use of time-scale separation property of the grid dynamics. Based on this identified reduced-order model, a reduced-dimensional linear quadratic regulator (LQR) can be designed, and finally implemented through an inverse projection-based broadcast control strategy as explained in the foregoing section. The challenge, however, arises from the fact that not all states of a generator exhibit time-scale separation. From coherency theory we know that the electro-mechanical (swing) states and excitation system states exhibit coherency and can be expressed as aggregate states by means of simple inertia-weighted averaging. The states of the power system stabilizer (PSS), however, do not exhibit this property. A hybrid state vector consisting of a mixture of averaged and non-averaged states, therefore,

need to be used for identifying the reduced-dimensional model, followed by a LQR design based on this model. The reduced-order part of the model saves significant amount of online time for both learning and control, while retaining the full-order PSS states enhances closed-loop stability. Newly evolving numerical methods such as N4SID with randomized singular value decomposition (*r*SVD) can be used for speeding up the identification. The cyber-physical architecture for implementing this learning-based WAC is shown in Fig. 2. The aggregator or control center in each area computes the averaged state of that area, while the independent system operator (ISO) identifies the reduced-order model followed by the WAC design.

Open Questions and Future Research

With proliferation of distributed renewable generation, many interesting opportunities for control and optimization are arising in wide-area control research. The following list of open questions is presented for future work.

1. The controllers outlined in this chapter are meant to address small-signal stability and voltage stability. In an actual grid, however, many other parallel control mechanisms will also exist – for example, load frequency control (LFC) – which maintains the grid frequency at the synchronous value despite fluctuations in loads. Typically, LFC is designed independent of WAC because of



Wide-Area Control of Power Systems, Fig. 2 Learning based wide-area control

its slower time-scale. However, with gradual disintegration of the grid into smaller microgrids this time-scale separation may become less dominant leading to a stronger coupling between the controllers. The positive and negative effects of this coupling need more research.

2. Another open question is on how control systems for transmission-level power systems such as those outlined in this chapter may become more dependent on distribution-level dynamics, and vice versa. Generally, transmission-scale controllers assume that distribution system dynamics are too fast and hence negligible. Distribution controllers, on the other hand, often assume the transmission grid to be an infinite source of current that can act as a slack bus whenever needed. Neither of these assumptions will hold over long-term. Thus, transmission and distribution (T&D) controllers may need to be designed together, raising a new set of research challenges for WAC.
3. Direct data-driven online design of the WAC damping controller K may become more useful than the system identification-based control described above. In that case, one must learn K directly after a contingency using online PMU measurements via machine learn-

ing algorithms such as reinforcement learning and Q-learning. This will be critical if the contingency changes the nominal A and B significantly.

4. A long list of cyber-physical challenges exist for WAC. Any WAC design must accommodate communication constraints such as delays, queuing, routing, data loss, synchronization loss, quantization, and cyber-security. A thorough economic analysis for fairly allocating the cost of implementing a wide-area communication network shared between multiple utility companies is also needed. Game theory can be a useful tool for formulating and solving such cost sharing problems (Lian et al. 2017).

Further Reading

For a more detailed discussion on how renewables such as wind and solar energy and their associated power electronic controllers can be co-design in conjunction with WAC, we refer the reader to the tutorial in Sadamoto et al. (2019). Fundamental dynamic models of synchronous generators that are used for designing WAC are listed in that tutorial. For more discussions on the cyber-physical aspects and cyber-security of

WAC, please refer to Ashok et al. (2017). Some recent results on the use of machine learning for WAC and PMU data analytics, in general, have been reported in Mukherjee et al. (2019) and Xie et al. (2014).

Cross-References

- [Cascading Network Failure in Power Grid Blackouts](#)
- [Power System Voltage Stability](#)

Bibliography

- Ashok A, Govindarasu M, Wang J (2017) Cyber-physical attack-resilient wide-area monitoring, protection, and control for the power grid. *Proc IEEE* 105(7):1389–1407
- Chakraborty A (2012) Wide-area damping control of power systems using dynamic clustering and TCSC-based redesigns. *IEEE Trans Smart Grid* 2(3):1503–1514
- Chakraborty A, Khargonekar P (2013) Introduction to wide-area monitoring and control. In: American control conference, Washington, DC
- Chakraborty A, Chow JH, Salazar A (2011) A measurement-based framework for dynamic equivalencing of power systems using wide-area phasor measurements. *IEEE Trans Smart Grid* 1(2):68–81
- Chaudhuri B, Majumder R, Pal B (2004) Wide-area measurement-based stabilizing control of power system considering signal transmission delay. *IEEE Trans Power Syst* 19(4):1971–1979
- Chaudhuri NR, Chaudhuri B, Ray S, Majumder R (2010a) Wide-area phasor power oscillation damping controller: a new approach to handling time-varying signal latency. *IET Gener Transm Distrib* 4(5):620–630
- Chaudhuri NR, Domahidi A, Chaudhuri B, Majumder R, Korba P, Ray S, Uhlen K (2010b) Power oscillation damping control using wide-area signals: a case study on nordic equivalent system. In: IEEE PES T&D conference and exposition
- Chaudhuri NR, Chakraborty D, Chaudhuri B (2012) Damping control in power systems under constrained communication bandwidth: a predictor corrector strategy. *IEEE Trans Control Syst Technol* 20(1):223–231
- Chow JH, Ghiocel SG (2012) An adaptive wide-area power system damping controller using synchrophasor data. In: *Control and Optimization methods for electric smart grids*, Editors: A. Chakraborty and M. D. Ilic, Springer, NY, pp 327–342
- Chow JH, Sanchez-Gasca J (2020) *Power system modeling, computation, and control*. New Jersey, Wiley/IEEE
- Dorfler F, Jovanovic M, Chertkov M, Bullo F (2014) Sparsity-promoting optimal wide-area control of power networks. *IEEE Trans Power Syst* 29(5):2281–2291
- Hernandez-Ortega MA, Messina AR (2018) Nonlinear power system analysis using Koopman mode decomposition and perturbation theory. *IEEE Trans Power Syst* 33(5):5124–5134
- Ilic M, Zaborsky J (2000) *Dynamics and control of large electric power systems*. New York, Wiley
- Jabr RA, Pal BC, Martins N (2010) A sequential conic programming approach for the coordinated and robust design of power system stabilizers. *IEEE Trans Power Syst* 25(3):1627–1637
- Julian DE, Schulz RP, Vu KT, Quaintance WH, Bhatt NB, Novosel D (2000) Quantifying proximity to voltage collapse using the voltage instability predictor (VIP). In: *Proceedings of IEEE PES summer power meeting*, vol 2, pp 931–936
- Kamwa I, Grondin R, Hebert Y (2001) Wide-area measurement based stabilizing control of large power systems – a decentralized/hierarchical approach. *IEEE Trans Power Syst* 16(1):136–153
- Lian F, Chakraborty A, Duel-Hallen A (2017) Game-theoretic multi-agent control and network cost allocation under communication constraints. *IEEE J Sel Areas Commun* 35(2):330–340
- Martinez EM, Vanfretti L, Sevilla FR (2013) Automatic triggering of the interconnection between Mexico and Central America using discrete control schemes. In: *IEEE PES ISGT Europe*
- Mukherjee S, Babaei S, Chakraborty A, Fardanesh B (2019) Measurement-driven optimal control of utility-scale power systems: a New York state grid perspective. *Int J Electr Power Energy Syst*, vol. 115, pp. 1–9.
- Phadke AG, Thorp JS (2008) *Synchronized phasor measurements and their applications*. Springer, New York
- Raoufat ME, Tomsovic K, Djouadi SM (2016) Virtual actuators for wide-area damping control of power systems. *IEEE Trans Power Syst* 31(6):4703–4711
- Sadamoto T, Chakraborty A, Ishizaki T, Imura J (2019) Dynamic modeling, stability, and control of power systems with distributed energy resources. *IEEE Control Syst Mag* 39(2):34–65
- Singh AK, Pal B (2014) Decentralized dynamic state estimation in power systems using unscented transformation. *IEEE Trans Power Syst* 29(2):794–804
- Taylor CW, Erickson DC, Martin KE, Wilson RW, Venkatasubramanian V (2005) Wide-area stability and voltage control system: R & D and online demonstration. *Proc IEEE* 93(5):892–906
- Wu H, Ni H, Heydt GT (2002) The impact of time delay on robust control design in power systems. In: *IEEE power engineering society winter meeting*, pp 1511–1516
- Xie L, Chen Y, Kumar PR (2014) Dimensionality reduction of synchrophasor data for early event detection: linearized analysis. *IEEE Trans Power Syst* 29(6):2784–2794
- Xue N, Chakraborty A (2019) Control inversion: a clustering-based method for distributed wide-area

- control of power systems. IEEE Trans Control Netw Syst. 6(3):937–949
- Zhang S, Vittal V (2013) Design of wide-area power system damping controllers resilient to communication failures. IEEE Trans Power Syst 28(4):4292–4300
- Zima M, Larsson M, Korba P, Rehtanz C, Andersson G (2005) Design aspects for wide-area monitoring and control systems. Proc IEEE 93(5):980–996